

C O M P U T E ! ' S

MORE

MACHINE LANGUAGE

GAMES

FOR THE

COMMODORE

64

Seven fast-paced machine language arcade games for the Commodore 64 and 128 (in 64 mode). Battle in space, shoot targets, try karate, and much more. Complete source code for all games listed, plus an easy-to-use assembler for writing your own games.

A **COMPUTE! Books** Publication

\$16.95



COMPUTE!'s
MORE
Machine
Language
Games
for the
COMMODORE
64

COMPUTE! Publications, Inc. 

Part of ABC Consumer Magazines, Inc.
One of the ABC Publishing Companies

Greensboro, North Carolina

The following article was originally published in *COMPUTE!* magazine, copyright 1986, COMPUTE! Publications, Inc.: "Prisonball" (October).

The following articles were originally published in *COMPUTE!'s Gazette*, copyright 1986, COMPUTE! Publications, Inc.: "Space Arena" and "The Fast Assembler" (January); "Saloon Shootout" (July); "Eagles and Gators" (September); "Ringside Karate" (October); "Bump-N-Run" (November); "Q-Bird" (December).

Copyright 1987, COMPUTE! Publications, Inc. All rights reserved.

Reproduction or translation of any part of this work beyond that permitted by Sections 107 and 108 of the United States Copyright Act without the permission of the copyright owner is unlawful.

Printed in the United States of America

ISBN 0-87455-094-7

10 9 8 7 6 5 4 3 2 1

The authors and publisher have made every effort in the preparation of this book to insure the accuracy of the programs and information. However, the information and programs in this book are sold without warranty, either express or implied. Neither the authors nor COMPUTE! Publications, Inc. will be liable for any damages caused or alleged to be caused directly, indirectly, incidentally, or consequentially by the programs or information in this book.

The opinions expressed in this book are solely those of the authors and are not necessarily those of COMPUTE! Publications, Inc.

COMPUTE! Publications, Inc., Post Office Box 5406, Greensboro, NC 27403, (919) 275-9809, is part of ABC Consumer Magazines, Inc., one of the ABC Publishing Companies, and is not associated with any manufacturer of personal computers. Commodore 64 and Commodore 128 are trademarks of Commodore Electronics Limited.

Contents

Chapter 1. The Games	1
More Amazing Games	
<i>Gregg Keizer, Editor</i>	3
Space Arena	
<i>Bryan Files</i>	6
Saloon Shootout	
<i>David Hensley, Jr. and Kevin Mykytyn</i>	8
Prisonball	
<i>John Scarborough</i>	11
Q-Bird	
<i>Mike Sedore</i>	13
Bump-N-Run	
<i>Tim J. Midkiff</i>	15
Ringside Karate	
<i>Anthony Bertram</i>	17
Eagles and Gators	
<i>Philip I. Nelson</i>	19
Chapter 2. Type It In	25
MLX: Machine Language Entry Program	
<i>Ottis R. Cowper</i>	27
Program 1. Space Arena	37
Program 2. Saloon Shootout	48
Program 3. Prisonball	61
Program 4. Q-Bird	67
Program 5. Bump-N-Run	79
Program 6. Ringside Karate	84
Program 7. Eagles and Gators	99
Chapter 3. The Source Code	113
The Fast Assembler	
<i>Yves Han</i>	115
The Source Code	133
Space Arena	
<i>Source code by Bryan Files</i>	137
Saloon Shootout	
<i>Source code by David Hensley, Jr. and Kevin Mykytyn</i>	148
Prisonball	
<i>Source code by Kevin Mykytyn</i>	162

Q-Bird
 Source code by Mike Sedore 169

Bump-N-Run
 Source code by Tim J. Midkiff 182

Ringside Karate
 Source code by Anthony Bertram 188

Eagles and Gators
 Source code by Philip I. Nelson 197

Appendices 215

A. How to Type In Programs 217

B. The Automatic Proofreader
 Philip I. Nelson 219

Disk Coupon 222

Chapter 1
The Games



More Amazing Games

Gregg Keizer, Editor

For arcade-game speed on the Commodore 64, nothing beats machine language.

Machine language (ML) is the native "tongue" of any computer, and the 64 is no exception. The 1's and 0's (ons and offs) of machine language, though, are a bit more difficult to work with than the English-style syntax of a computer language like BASIC. To write an ML game, you have to know what you're doing.

The authors of these seven machine language games for the Commodore 64 (and 128 running in 64 mode) know their stuff. First (and most importantly), the games are fun to play. But what makes them stand out, what makes them so impressive, is their length. Since they all appeared originally in *COMPUTE!* magazine or *COMPUTE!'s Gazette*, they had to be short enough to publish—not so long that people took one look and refused to type them in. The longest game, "Eagles and Gators," is just over 5K long. That's considerably shorter than most commercial games.

Consider the length of these seven games and they're simply amazing.

Play Them All

You'll find a variety of games in *COMPUTE!'s More Machine Language Games for the Commodore 64*. Much like those found in the first such book (*COMPUTE!'s Machine Language Games for the Commodore 64*), the seven arcade games here are as challenging (and often as frantic) as we can make them. There are enough things going *blip* and *zip* to satisfy any gamester.

Space Arena puts two players in a futuristic arena where spaceships maneuver and fire while dodging sluggish asteroids. Shove your opponent into a hunk of slow-moving rock and iron to win. The victor decides the fate of the galaxy.

Saloon Shootout lets you blast mugs, plink at running

Chapter 1

mice, gun down bad guys, and plug playing cards. You've got a gun in your hand, plenty of bullets, and only so much time in this entertaining arcade game for shootists of all ages.

Prisonball is a two-player hybrid—it combines the best elements from the classic computer arcade games, *Breakout* and *Pong*. Knock out as many bricks as possible—more than your opponent does, of course—before time runs out. In this game, fortunes turn as fast as your paddles.

Q-Bird makes you responsible for a defenseless baby bluebird who's the entree for a variety of hungry predators. Hop and jump on the grid to advance to the next level, but be sure there's not a mouth waiting for you when you land.

Bump-N-Run may make friends into enemies as you and an opponent catch, steal, and shoot a ball in this tumble-of-fun game of bumps and bashes.

Ringside Karate may not provide the black belt, but it has everything else you need to kick, jab, punch, and block your way to the title of World Karate Expert. A quick-reflex two-player game that tests your joystick control.

Eagles and Gators features a split screen, 16 sprites (amazing!), and clever animation. Two players turn a flock of eagles and a ton of turtles into clamping, shoving opponents in this unique game.

Easy Entry

If you had to use an assembler (see Chapter 3) to enter these games, you'd probably give up before you started. Fortunately, you can use our "MLX" machine language entry program instead. MLX is a program in its own right, one that checks your typing and lets you know—almost immediately—that you've made a mistake. Typing in machine language programs with MLX guarantees that you'll have a working copy of the game the first time.

All seven games in *COMPUTE!'s More Machine Language Games for the Commodore 64* are listed in MLX format. Refer to Chapter 2 for the MLX program, its instructions, and each game's MLX listing and individual directions.

From the Source

Source code is the original code of a program. In the case of machine language games like these seven, the source code was written with an assembler (in many cases, the *PAL* assembler was originally used). If you know how to read it (even if you

don't know how to write it), source code can be illuminating. It shows you exactly how the program was put together, what routines were used, and which tricky techniques were called into play.

Source code is hard to come by. Many programmers won't show it to their best friends.

That makes this book even more unusual, because we've published the complete assembled source code for each game.

If you do dive into machine language programming, then you'll want to be sure to turn to Chapter 3. You'll undoubtedly find programming methods that will make your own creations run faster, look better, and play easier.

Grab the Joystick

No matter what you use this book for—to simply play the seven games or pick apart the source code—*COMPUTE!'s More Machine Language Games for the Commodore 64* will satisfy even the most experienced player or programmer. Some serious fun is waiting for you either way.

All the programs in this book are available in disk form. Each game's object code (what you run, and what you'd normally create by typing in the MLX listing in Chapter 2) and complete source code (in single-statement-line PAL format) is included on the disk. You can order this \$12.95 disk only through COMPUTE! Publications. Call toll-free 1-800-367-6767 (in NY 1-212-887-8525).

Space Arena

Bryan Files

This engaging two-player game, which could have been titled "Indirect Aggression" for its unique design, features superior graphics and playability. Two joysticks required.

In a distant galaxy, a fierce war between two great empires has reached a stalemate. The cost to both civilizations has been high with no gain to either side. The stalemate has forced the two sides to agree to settle the dispute in a controlled environment—a space arena.

Enclosed in this arena are four asteroids and a ship representing each side. Both ships are equipped with photon blasters, which have the ability to push objects without damaging them. A ship is destroyed only by a collision with an asteroid.

Let the Game Begin

To load "Space Arena," type

LOAD "filename",8,1 (for disk)

LOAD "filename",1,1 (for tape)

where *filename* is the name you used to save the program.

Press RETURN; then type

SYS 49152

and press RETURN again.

You'll need two joysticks (and two players) to play Space Arena. Plug a joystick into each of the 64's two ports.

Rotate and Fire

Movement in Space Arena is much like that in the classic arcade game *Asteroids*. Rotate your ship by moving the joystick left or right. Push the joystick forward to activate the ship's thrusters, and press the fire button to release a photon blast. You have no braking thrusters—to slow down, you must turn the ship around and thrust in the direction in which you're moving.

Game options include speed (1–3, where 1 is the slowest and 3 the fastest) and time limit (1–5 minutes).

Points and Strategy

Victory conditions are simple—score more points than your opponent in the time allowed. You get one point each time you hit the other ship with a photon blast. If the enemy's ship is destroyed by running into an asteroid, you receive 20 points.

There are two basic strategies to playing Space Arena. You can try to push your opponent into an asteroid, or push the asteroids into your opponent. The effect of photon blasts on asteroids is much less than on ships because the asteroids are heavier. A single photon blast can send a ship flying across the screen, especially if the ship has to turn around before being able to slow down.

Typing

Refer to Chapter 2, page 37, for typing instructions for Space Arena and its MLX listing.

Source Code

The source code for Space Arena is in Chapter 3, beginning on page 137.

Saloon Shootout

David Hensley, Jr. and Kevin Mykytyn

You'll have your hands full in this action-packed, arcade-style game. It's one of the most challenging and graphically charming games we've published. A joystick is required.

If you're a sure-shot, dead-eye, trigger-happy gunslinger, you'll be well suited to this wild and woolly action game. In "Saloon Shootout," the pace is frantic and it never slows up. The object of the game is to accumulate the highest number of points before time runs out or before you run out of bullets.

Start Shooting

To play Saloon Shootout, load the game by typing

LOAD "filename",8,1 (for disk)

LOAD "filename",1,1 (for disk)

Press RETURN. Now enter

SYS 10240

and press RETURN again.

Plug the joystick into port 2 and you're ready to play.

Of Mice and Mugs

There's a lot to contend with in Saloon Shootout. A gun, which appears on the bar, is your weapon. It's controlled by your joystick. By moving the joystick up and down, you control the distance your bullet travels. You'll notice the gun barrel shift up and down as you move the joystick. The arrow on the right of the screen will help you gauge the distance your shot will fire.

A main objective of the game is to shoot as many mugs as you can before they fall off the bar. For every ten mugs you shoot, a running mouse appears. Each time you shoot this critter, you receive a bonus score, and the mouse reverses direction and speeds up. Every time you hit it, the point total escalates, and so does the mouse's speed. (This is where you can get a lot of points for those potential high scores.)

This may sound pretty easy so far, but your job at the saloon is much more complicated. You must also keep your eye on the three windows on the back wall of the saloon. They're sometimes opened by a masked desperado who will steal ten of your bullets unless you shoot him before he disappears. But don't be too hasty: Sometimes good guys appear (the good guys smile), and shooting them results in a stiff penalty—100 points subtracted from your score.

If your supply of bullets is running low, you can replenish them by shooting the cards that appear on top of the player piano. However, you must shoot the cards in order: 10, J, Q, K, and A. Be careful here—one card hit out of order will re-shuffle the deck and you'll have to start over. Not so simple, is it?

There are a few features that help to calm your nerves, however. If you reach a score of 500, you receive a bonus time of 50 seconds. To help you keep track of the time remaining, the screen flashes when there are 25 seconds left. As noted above, the arrow on the right side of the screen helps make sure that those valuable bullets you're shooting aren't wasted. And finally, the player piano cranks out ragtime tunes to help relieve the tension. To change the tune, shoot one of the white keys on the player piano. To turn the music off, shoot a white key twice.

Statistics and Point Totals

Time	2 minutes
Bullets	35
Scoring	
Mugs	10
Missed mugs	-5
Cards	10
Back of card	-10
Good guy	-100
Bad guy	25
Mouse	20,40,60,80,100

Shootist Strategy

When the game first runs, you have a few seconds before the mugs start to appear. It's wise to use this time to shoot a few of the cards (in order, of course) in case your bullets run low

Chapter 1

during the game. Try to make the mouse appear as many times as possible to receive higher scores. When it appears, try to shoot it as many times as possible. Unless your bullet count is high, it's a good idea to set your sights on the windows as they start to open. And remember to keep an eye on your bullet supply and the time remaining.

Typing

You'll find typing instructions for Saloon Shootout and its MLX listing in Chapter 2, page 48.

Source Code

Saloon Shootout's source code can be found in Chapter 3, page 148.

Prisonball

John Scarborough

Commodore 64 Version by Kevin Mykytyn

Nearly everyone has played Pong or Breakout, two computer-game classics. "Prisonball" creates an intense, two-player action game by drawing features from both of those games. Requires either a pair of paddles or two joysticks.

"Prisonball" is a two-player action game that combines the best elements from two classic computer games, *Pong* and *Breakout*. The object of the game is simple—knock out as many bricks as you can in the allotted time.

Start Your Paddles

Either joysticks or paddles can be used to play Prisonball. (To play with paddles, plug a pair of paddles into port 2.) Even though the program is written in machine language, load and run it as you would an ordinary BASIC program, by typing

LOAD "filename",8 (for disk)

LOAD "filename",1 (for tape)

and pressing RETURN. Type

RUN

and press RETURN again.

Break to the Center

The game begins by displaying five colored walls running vertically down the center of the screen. Each player controls two paddles located at the left and right sides of the screen. Three balls appear at a random location and start bouncing around the screen. When a ball is on your side of the screen, move one of your paddles into its path to deflect the ball toward the walls. You can only hit a ball when it's moving toward your paddles (away from the interior walls). Balls traveling from the opposite direction go right through your paddles. If you happen to miss a ball, it wraps around the screen and appears on

Chapter 1

the other side, giving your opponent a chance to score.

At the beginning of the game, all three balls are a neutral color. Each time you hit a ball, it changes to the color of your paddle. You score whenever a ball of your color hits one of the five interior walls. The score depends on which wall you hit. The center wall is the hardest to reach, so it yields the most points. The two intermediate walls are worth less than the center wall. The outermost walls are easiest to hit and score the fewest points.

Time and Points

The top of the screen displays each player's score and a count-down timer. Each game lasts three minutes. When the timer runs to zero, the game ends and the player with the most points wins. The screen border flashes briefly as a warning when only 20 seconds remain on the timer.

Bricks from the center wall are worth 30 points, those from the two adjacent walls are worth 20, and the outermost bricks each score 10 points.

Every time a ball hits one of the walls, a brick is knocked out of the wall at the point of impact. By aiming your shots carefully, you can bore a path through a wall and move a ball into the interior space between two walls. When this happens, the ball bounces wildly back and forth between the walls, scoring many points in a short time.

An additional bit of strategy has to do with the redrawing of walls. Whenever a wall has been destroyed, it is immediately redrawn. Some of the highest scores result when you trap one or more balls behind a wall when it is redrawn. Since the wall is new, the trapped balls may hit it many times before they break back out to the exterior.

Typing

Look to Chapter 2, page 61, for typing instructions and the MLX listing for Prisonball.

Source Code

Prisonball's source code can be found in Chapter 3, beginning on page 162.

Q-Bird

Mike Sedore

This delightful and colorful arcade-style game challenges your character—a defenseless (but nimble) baby bluebird—to survive among a crew of nasty, hungry enemies. A joystick is required.

Other birds say you're paranoid, but you're not—everyone really *is* out to get you. You often ask yourself how long a defenseless baby bluebird can hope to survive when a host of voracious predators are looking for a meal. If only you could leave this place to find a new home. But alas—you're too young to fly. The best you can manage is a flapping long jump. But you'd better be careful not to jump too far: You could fall a long way down.

Ready to Jump

Although "Q-Bird" is written in machine language, it can be loaded and run like a BASIC program. To play, type

LOAD "filename",8 (for disk)

LOAD "filename",1 (for tape)

where *filename* is the name you gave the program. Press RETURN; then type

RUN

pressing RETURN once more. Plug the joystick into port 2.

Playing on the Grid

Q-Bird is played on a 6×7 grid. There are 15 levels of play from which to choose. As you play, you advance level by level by completing grids. A grid is complete when the color of each square matches the goal color shown at the bottom of the screen. To change the color of a square, simply hop on it. You have to jump on each square once in level 1, twice in level 2, and so on, up to level 15. A bonus life is awarded each time you complete a level. If you reach level 15, you

Chapter 1

should feel satisfied. It doesn't get any harder than this. But don't relax; it doesn't get any easier either.

You start with five lives. Choose your starting level by pushing forward on the joystick. If you pass the level you want to play, pull back on the stick to reverse the level counter. Press the fire button to begin play.

To hop, simply move the joystick in the direction you want to follow. For a super jump, push the fire button as you hop—you'll leap over a square and land on the next. If you time it right, you can leap right over your enemies. But be careful not to jump over the side of the grid. That costs one life.

Press SHIFT-LOCK to pause the game. To restart a game, press RUN/STOP-RESTORE.

Leapin' Lizards

Three purple lizards live on the grid. They randomly leap off of their tails from square to square, landing on any bluebirds careless enough to get in their way. But they're the least of your worries. There's a king cobra that relentlessly chases you. He's got a hole in his stomach just the size of a baby bluebird. If he doesn't get you, then perhaps the low-flying and hungry hawk will. And while you're looking out for all of these villains, try to dodge the runaway balls that roll down the grid. Any one of them could turn you into a bluebird pancake in a moment.

Fortunately, you do have an ally. Occasionally, a flashing egg appears on a random square. If you hop on it, you momentarily stun all the grid inhabitants. You can now go anywhere you please without harm. Unfortunately, this doesn't last long. When the safe time is nearly through, the bluebird begins flashing. All action returns to normal after the third flash, so be ready for the frenzy to continue.

Typing

Chapter 2, page 67, has the typing instructions and MLX listing for Q-Bird.

Source Code

The source code for Q-Bird begins on page 169, Chapter 3.

Bump-N-Run

Tim J. Midkiff

Challenge a friend in this fast-paced, action/strategy game for two players. Rapid play and smooth movement are its hallmarks. Two joysticks are required.

This two-player action game can be played at two conceptual levels: When you first play, it seems to be purely an action game—all speed and reflexes—but after you and your opponent master the basics of the game, you'll discover the importance of strategy.

The object of "Bump-N-Run" is to get the ball and shoot it into the other player's goal. Of course, your opponent is trying to do the same.

Begin to Bump

To load the game, type

LOAD "BUMP-N-RUN",8,1 (for disk)

LOAD "BUMP-N-RUN",1,1 (for tape)

and press RETURN (substitute the correct filename for *BUMP-N-RUN* if you saved it to tape or disk with a different name).

Now type

SYS 49152

and press RETURN to begin play. Be sure you have two joysticks plugged in.

Press RESTORE to exit the game, and enter **SYS 49152** again to replay.

Bumping and Running

As with many computer games, the best way to learn Bump-N-Run is by playing it.

The game begins with the ball in the middle of the court and each player guarding his or her own goal. The joystick in port 1 controls the purple player. The yellow player is controlled by the joystick in port 2. Catch the ball by touching it

Chapter 1

with your player. Your opponent can then steal the ball by bumping into you. The speed of your player and the direction in which the player travels are controlled by the joystick. When you first push the joystick, you move slowly; keep pushing and you'll accelerate. Press the fire button to shoot the ball. When the ball is released, you lose your speed, but the ball continues on with your previous speed and direction. Make sure your shot is accurate, because once you shoot the ball you can't catch it again until it bounces off a wall. Remember that you can steal the ball at any time simply by bumping your opponent. The first player to score 15 goals wins, and the game is reset when both players press the fire button.

Here are a few more details concerning the physics of the game. When players collide, their velocities and directions are exchanged. Collisions with the walls cause players to bounce back. A player may go beyond the walls only if he or she is pushed by another player.

Typing

Refer to Chapter 2, page 79, for Bump-N-Run's typing instructions and MLX listing.

Source Code

Bump-N-Run's source code starts on page 182, Chapter 3.

Ringside Karate

Anthony Bertram

Go head to head with another player for the coveted black belt in this outstanding commercial-quality, arcade-style game. Two joysticks are required.

As in real karate, skill, speed, and timing are the necessary ingredients to play "Ringside Karate." You've got an impressive arsenal of kicks, punches, and defensive blocks at your disposal—but then, so does your opponent.

Kick Away

The game is long, but much of the sprite and screen DATA are repeated numbers, which makes the typing a little easier.

To load Ringside Karate, type

LOAD "filename",8,1 (for disk)

LOAD"filename",1,1 (for tape)

and press RETURN. (*Filename* is the name you used to save the program.)

To start the game, type

SYS 15490

and press RETURN again.

The joystick plugged into port 1 controls the fighter dressed in white, while the joystick in port 2 controls the fighter in black.

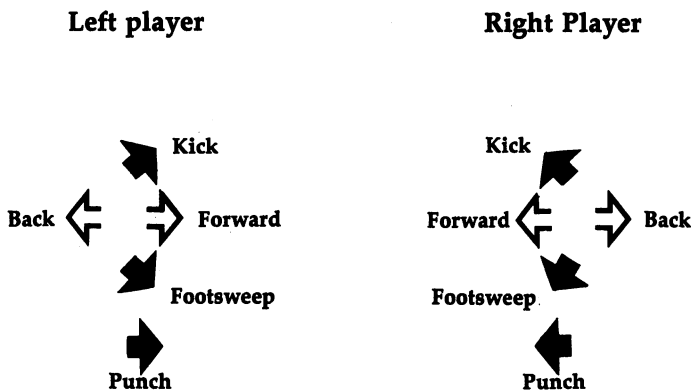
Hee-Aee!

Each player controls one fighter with a joystick. Moving the joystick and/or pressing the button in various combinations delivers different kicks, punches, and defensive blocks.

It may take some practice to come up with the right offensive move. Keep at it.

To block, just press the fire button while the joystick is pushed in any other direction than the ones outlined in the il-

Chapter 1



Solid arrows indicate that the joystick button should be pressed.

lustration. To win, you must score ten knockdowns, each worth 100 points.

Timing is crucial because the fire button must be released for a moment before you can make another attack. If you rapidly press the button, the fighter won't move and probably will be knocked down. Blocking provides a limited defense and is best used just before a counterattack or when backing up. You'll find the controls much easier to master if you've played an arcade karate game.

When the game is over, a winner is declared, and another game may be started by pressing any key.

Typing

Typing instructions and the MLX listing for Ringside Karate can be found in Chapter 2, page 84.

Source Code

Ringside Karate's source code starts on page 188, Chapter 3.

Eagles and Gators

Philip I. Nelson

Featuring 16 sprites on a split screen, great sound effects, and realistic animation, this all machine language, two-player action game pits creature against creature as each tries to shove the other into oblivion. Requires at least one joystick.

"Eagles and Gators," an intense two-player action game, displays 16 moving sprites at all times—double the normal limit—and includes highly realistic animation, too.

Ready to Swim and Fly

To play, plug two joysticks into the joystick ports. (See the instructions below for playing with only one joystick.) Although the program is written in machine language, you load and run it just as you would a BASIC program, with

LOAD "filename",8 (for disk)

LOAD "filename",1 (for tape)

followed by **RUN**.

Eagles and Gators begins with an animated title screen which asks whether you want to read the game instructions. Press Y to view the instruction screen or N to skip it. Next, you're shown the game screen and are prompted to choose one of the nine skill levels. If you've never played the game before, you'll probably want to press 1 to choose the easiest level.

Flyers Against Swimmers

When play begins, the screen is split into upper and lower color zones—sky-blue for the eight high-flying eagles and murky black for the eight gators swimming below. Each player controls a group of eight sprites. The game is written to be

Chapter 1

played with two joysticks, but you can substitute keypresses to control the gators, normally controlled by joystick 1. Press the CTRL key to move left or the 2 key to move right. The space bar serves as a fire button.

Your sprites are in constant, realistic motion, flying or swimming up and down in their zones, rebounding when they hit the border between the two worlds. The flashing colored sprite in your group is the one controlled by your joystick. To switch control to a different sprite, push the joystick left or right.

The goal of the game is to expand your zone and shove your opponents off the screen. This is done by convincing your creatures to push against the sky/water border. Press the fire button just as your flashing eagle or gator bumps the border, and it "clamps" on and starts to shove. Whenever you clamp successfully, your opponent's screen flashes red, and all your creatures thrash momentarily with excitement.

Your zone expands whenever you have more sprites pushing than your opponent does. The game ends when one side or the other takes over the entire screen. That sounds simple, but it takes precise timing to clamp just one sprite. If you press the fire button at the wrong time, your sprite moves away from the border, regardless of its initial direction. If you press the fire button when your flashing sprite's already stuck to the border, it bounces back to the farthest end of your zone.

Your opponent can also unclamp one of your creatures by clamping one of his or her own creatures directly across the border from yours. The ability to unclamp an opponent's creatures at any time adds an extra element of strategy and makes it possible to turn the tables on your opponent even when defeat seems almost certain. The game is designed to reward concentration and careful play, and to discourage wild tactics such as holding the fire button down constantly. At best, you'll prevent your creatures from approaching the border. At worst, you'll inadvertently unclamp those from your side who are already pushing.

When you win a game, the screen clears to your color, and a joyful, double-sized sprite from the winning side performs a victory dance across the screen. At this point you can pick a new skill level for the next round or quit by pressing the Q key. After you've exited, the game can be restarted by typing **SYS 16384** and pressing RETURN. To stop in midplay, press **RUN/STOP-RESTORE**.

Displaying 16 Sprites

Space doesn't permit a detailed explanation of every routine in this program, but you may be curious about how to display more than eight sprites at a time, something often mentioned, but rarely seen. To understand how it's done, you'll need to know a little about raster displays.

Your TV or monitor makes a picture by scanning its screen with a moving electron beam called a *raster*. This is done line by line, from top to bottom, 60 times a second. The visible portion of a TV picture is made up of 200 raster lines, numbered from 50 at the top through 249 at the bottom. The same numbering scheme is used for locating sprites vertically on the screen, so raster line 100, for example, matches up with sprite vertical location 100. This correspondence between raster lines and sprite positions greatly simplifies the task of creating more than the usual number of sprites.

The Commodore 64 keeps track of the raster's vertical location in a special memory location (register) at 53266 (\$D012). BASIC is too slow to make much use of it, but machine language (ML) is fast enough to let us check this register repeatedly and monitor the position of the raster beam as it scans down the screen. This information is used to create a split screen as well as to double the usual number of sprites.

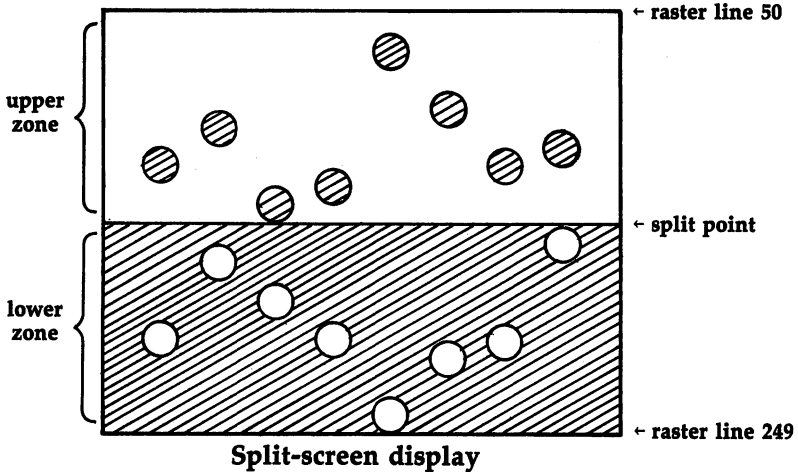
Split Screens

To make the two-color split screen, we first establish an arbitrary "split" point somewhere near the midpoint of the screen (say, at line 100), then repeatedly compare the raster's location to that value. When the raster value is less, we know it's above the split, so we color the screen cyan to create the sky zone. When the raster value is greater, that means it has dipped below the split point, so we color the screen black for the water zone. Remember, moving downward on the screen increases the raster (or sprite) vertical location value. By changing the border color along with the screen background color, we can make the split extend the full width of the screen for heightened realism.

The same sort of monitoring makes it possible to display 16 sprites. When the raster's above the split, we display all of the 64's 8 sprites at vertical locations in the upper color zone. When it drops below the split, we display the same 8 sprites at locations in the lower zone. The two sets of sprites are never actually displayed at the same time. The program just

Chapter 1

changes their locations so fast that your eye sees 16 solid figures, without a trace of flicker.



Shadow Registers

If you stopped at this point, you'd have two identical groups of eight sprites—upper and lower—sitting motionless on a split screen. Nice, but boring.

To exploit the effect in a game, we want the two sprite groups to look different and move independently. Unfortunately, the Commodore 64 still has only eight sets of sprite control registers for handling the sprites' shapes, colors, and so on. Remember, we're just flipping the same group of eight sprites back and forth at very high speed between different vertical locations. Thus, top sprite 1 doesn't just look the same as bottom sprite 1—it *is* the same sprite.

The minute you change the shape or color of any sprite in the upper group, its lower twin immediately follows suit, and vice versa.

What we really need is 16 sets of sprite control registers. Commodore didn't design 16 sprites for the 64, but you can achieve much the same effect through software. Here's an outline of how it's done in this program.

First, we set aside some free memory to use as "shadow" registers in place of the actual sprite control registers. Since none of the eagle or gator sprites move horizontally, we don't need to duplicate the 8 horizontal position registers. However,

every sprite must be able to move freely up or down and change its shape and color at any time. Thus, for each group of 8 sprites we need 8 registers for vertical position, 8 for color, and 8 more to hold shape data pointers. Three sets of 8 make 24. Double that for the second set of 8 sprites, and you now have 48 shadow registers.

To make each sprite appear in the correct spot, we must transfer the contents of each shadow register into the corresponding control register at precisely the right instant. When the raster beam is in the upper portion of the screen, we'll take data from the eagles' set of shadow registers and store it in the actual sprite control registers. When the raster creeps down below the split, we'll fetch gator data instead and plug that into the control registers.

Of course, it's not necessary to do this on a two-color split screen. This particular program changes screen colors at the split point to accentuate the division between sides. But in another application, you might prefer to leave out the color-changing code. The split location is just an invisible reference point that tells you when it's time to flip the sprites from one zone into the other.

Coasting on the Interrupt

Expressed in this simple, schematic form, the job of displaying 16 sprites sounds quite simple. When the raster beam is above the midpoint, make the sprites appear in the upper area. When it's below the split, move them down into the lower area and give them new shapes and colors. The process is uncomplicated in theory. But don't bother trying to do this in BASIC, which is many times too slow to read the raster's position accurately. This program uses an interrupt-driven routine to manage the 16-sprite effect. Although interrupt handling goes beyond the scope of this article, the technique essentially wedges our custom sprite display routines into the computer's operating system along with its normal hardware-servicing routines, which do background tasks such as scanning the computer's keyboard. Conveniently, the interrupt-driven routine is executed 60 times per second, exactly the same frequency at which the screen is redrawn. As a result, the sprite display and color split are quite stable.

The advantage of driving an ML routine on the hardware interrupt is that it becomes completely automatic. Sixty times every second, the computer executes your custom routine

Chapter 1

along with its own housekeeping routines. An interrupt-driven routine seems to coast along all by itself, without slowing down other program events. In effect, we've enhanced the 64 by adding the ability to display 16 sprites, with the option of a two-color split screen. Because the custom routines operate in the background, the shadow sprite control registers seem to work exactly the way the actual control registers do. We can change the color, shape, or position of any of our 16 sprites by plugging different values into their shadow control registers. To move our border up or down, we just change the value for our split point, and everything else follows accordingly.

The price you pay for these new features is that the sprites can't leave their respective zones. If you try to move an eagle sprite down into the gator zone, the custom display routine automatically gives it a gator shape and color, and the sky is minus one eagle. The same thing happens in reverse if you move a gator into the upper zone. If you move any sprite so that it overlaps the border, you get a weird hybrid creature with an eagle-shaped upper body and a gator tail. It's easy to forget because the shapes look so real, but the 64 still has only 8 sprites built into hardware. Using similar techniques, it's possible to display as many as 64 sprites on the screen at once. All you need to do is divide the screen vertically into additional zones, add extra shadow registers, and modify the display routines to handle them. However, the same general rules apply. Each group of sprites is confined to its own vertical zone, and you may not have more than 8 sprites occupying the same screen raster line.

Typing

Refer to Chapter 2, page 99, for the typing instructions and MLX listing for Eagles and Gators.

Source Code

Eagles and Gators' source code can be found in Chapter 3, page 197.

Chapter 2
Type It In



MLX

Machine Language Entry Program

Ottis R. Cowper

“MLX” is a labor-saving utility that allows almost fail-safe entry of machine language programs.

Type in and save at least two copies of “MLX” (you’ll want to use it to enter future Commodore 64 machine language programs from *COMPUTE!* magazine, *COMPUTE!’s Gazette*, and *COMPUTE!* books). If you’re using a Commodore 128, you can still use MLX—you must enter it and any ML programs in 64 mode, however.

When you’re ready to enter an ML program, load and run MLX. It asks you for a starting address and an ending address. These addresses appear in the short article accompanying the MLX-format listing of each game. You’ll find the listings in Chapter 2.

If you’re unfamiliar with machine language, the addresses (and all other values you enter in MLX) may appear strange. Instead of the usual decimal numbers you’re accustomed to, these numbers are in *hexadecimal*—a base 16 numbering system commonly used by ML programmers. Hexadecimal—*hex* for short—includes the numerals 0–9 and the letters A–F. But don’t worry—even if you know nothing about ML or hex, you should have no trouble using MLX.

After you enter the starting and ending addresses, you’ll be offered the option of clearing the workspace. Choose this option if you’re starting to enter a new listing. If you’re continuing a listing that’s partially typed from a previous session, don’t choose this option.

A functions menu appears. The first option in the menu is ENTER DATA. If you’re just starting to type in a program, pick this. Press the E key, and type the first number in the first line of the program listing. If you’ve already typed in part of a program, type the line number where you left off typing

Chapter 2

at the end of the previous session (be sure to load the partially completed program before you resume entry). In any case, make sure the address you enter corresponds to the address of a line in the listing you are entering. Otherwise, you'll be unable to enter the data correctly. If you press E by mistake, you can return to the command menu by pressing RETURN alone when asked for the address. (You can get back to the menu from most options by pressing RETURN with no other input.)

Entering a Listing

Once you're in Enter mode, MLX prints the address for each program line for you. You then type in all nine numbers on that line, beginning with the first two-digit number after the colon (:). Each line represents eight data bytes and a checksum. Although an MLX-format listing appears similar to the "hex dump" listings from a machine language monitor program, the extra checksum number on the end allows MLX to check your typing.

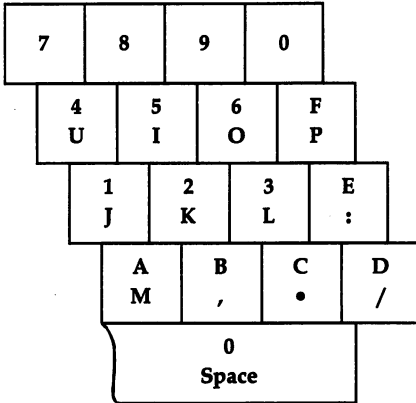
When you enter a line, MLX recalculates the checksum from the eight bytes and the address, and compares this value with the number from the ninth column. If the values match, you'll hear a bell tone, the data will be added to the workspace area, and the prompt for the next line of data will appear. But if MLX detects a typing error, you'll hear a low buzz and see an error message. The line will then be redisplayed for editing.

Invalid Characters Banned

Only a few keys are active while you're entering data, so you may have to unlearn some habits. You *do not* type spaces between the columns; MLX automatically inserts these for you. You *do not* press RETURN after typing the last number in a line; MLX automatically enters and checks the line after you've typed the last digit.

Only the numerals 0-9 and the letters A-F can be typed in. If you press any other key (with some exceptions noted below), you'll hear a warning buzz. To simplify typing, a keypad is available. The keypad—part of the keyboard—is active only while you're entering data. Addresses must be entered with the normal letter and number keys. The figure below shows the keypad configuration.

64 MLX Keypad



MLX checks for transposed characters. If you're supposed to type in A0 and enter 0A instead, MLX will catch your mistake. There is one error that can slip past MLX: Because of the checksum formula used, MLX won't notice if you accidentally type FF in place of 00, and vice versa. And there's a very slim chance that you could garble a line and still end up with a combination of characters that adds up to the proper checksum. However, these mistakes should not occur if you take reasonable care while entering data.

Editing Features

To correct typing mistakes before finishing a line, use the INST/DEL key to delete the character to the left of the cursor. (The cursor-left key also deletes.) If you really mess up a line, press CLR/HOME to restart the line. The RETURN key is also active, but only before any data is typed on a line. Pressing RETURN at this point returns you to the command menu.

After you've typed a character of data, MLX disables RETURN until the cursor returns to the start of a line. Remember, you can press CLR/HOME to get to a line number prompt quickly.

More editing features are available when you're correcting lines in which MLX has detected an error. To make corrections in a line that MLX has redisplayed for editing, compare the line on the screen with the one printed in the listing; then move the cursor to the mistake and type the correct key. The cursor-left and -right keys provide the normal cursor controls. (The INST/DEL key now works as an alternative cursor-left

Chapter 2

key.) You cannot move left beyond the first character in the line. If you try to move beyond the rightmost character, you'll reenter the line. During editing, RETURN is active; pressing it tells MLX to recheck the line. You can press the CLR/HOME key to clear the entire line if you want to start from scratch or if you want to get to a line number prompt to use RETURN to get back to the menu.

Display Data

The second menu choice, DISPLAY DATA, examines memory and shows the contents in the same format as that of the program listing (including the checksum). When you press D, MLX asks you for a starting address. Be sure that the starting address you give corresponds to a line number in the listing. Otherwise, the checksum display will be meaningless. MLX displays program lines until it reaches the end of the program, at which point the menu is redisplayed. You can pause the display by pressing the space bar. (MLX finishes printing the current line before halting.) Press the space bar again to restart the display. To break out of the display and get back to the menu before the ending address is reached, press RETURN.

Other Menu Options

Two more menu selections let you save programs and load them back into the computer. These are SAVE FILE and LOAD FILE; their operation is quite straightforward. When you press S or L, MLX asks you for the filename. You'll then be asked to press either D or T to select disk or tape.

You'll notice the disk drive starting and stopping several times during a load or save. Don't panic; this is normal behavior. MLX opens and reads from or writes to the file instead of using the usual LOAD and SAVE commands. Disk users should also note that the drive prefix 0: is automatically added to the filename (line 750), so it should *not* be included when you enter the name. This also precludes the use of @ for Save-with-Replace, so remember to give each version you save a different name.

Remember that MLX saves the entire workspace area from the starting address to the ending address, so the save or load may take longer than you might expect if you've entered only a small amount of data from a long listing. When saving a partially completed listing, be sure to note the address where

you stopped typing so that you'll know where to resume entry when you reload.

MLX reports the standard disk or tape error messages if any problems are detected during the save or load. (Tape users should bear in mind that Commodore computers are never able to detect errors during a save to tape.) MLX also has three special load error messages: INCORRECT STARTING ADDRESS, which means the file you're trying to load does not have the starting address you specified when you ran MLX; LOAD ENDED AT *address*, which means the file you're trying to load ends before the ending address you specified when you started MLX; and TRUNCATED AT ENDING ADDRESS, which means the file you're trying to load extends beyond the ending address you specified when you started MLX. If you see one of these messages and feel certain that you've loaded the right file, exit and rerun MLX, being careful to enter the correct starting and ending addresses.

The QUIT menu option has the obvious effect—it stops MLX and enters BASIC. The RUN/STOP key is disabled, so the Q option lets you exit the program without turning off the computer. (Of course, RUN/STOP-RESTORE also gets you out.) You'll be asked for verification; press Y to exit to BASIC or any other key to return to the menu. After quitting, you can type RUN again and reenter MLX without losing your data, as long as you don't use the CLEAR WORKSPACE option.

The Finished Product

When you've finished typing all the data for an ML program and you've saved your work, you're ready to see the results. The instructions for loading and using the finished product vary from program to program. Some ML programs are designed to be loaded and run like BASIC programs, so all you need to type is LOAD "*filename*",8 for disk or LOAD "*filename*" for tape, and then RUN. Such programs will usually have a starting address of 0801. Other programs must be reloaded to specific addresses with a command such as LOAD "*filename*",8,1 for disk or LOAD "*filename*",1,1 for tape, then started with a SYS to a particular memory address. On the Commodore 64, the most common starting address for such programs is 49152, which corresponds to MLX address C000. In either case, refer to the general description for each game—you'll find these in Chapter 1.

Chapter 2

An Ounce of Prevention

By the time you finish typing in the data for a long ML program, you may have several hours invested in the project. Don't take chances—read Appendix A, "How to Type In Programs," use our "Automatic Proofreader" to type in MLX, and then test your copy *thoroughly* before first using it to enter any significant amount of data. Make sure all the menu options work as they should. Enter fragments of the program starting at several different addresses; then use the DISPLAY option to verify that the data has been entered correctly. And be sure to test the SAVE and LOAD options several times to insure that you can recall your work from disk or tape. Don't let a simple typing error in the new MLX cost you several nights of hard work.

MLX

Read Appendix A before entering the program. For error-free entry, use "The Automatic Proofreader" (Appendix B) to type in this program.

```
SS 10 REM VERSION 1.1: LINES 830,950 MODIFIED, LINES
    ES 485-487 ADDED
EK 100 POKE 56,50:CLR:DIM IN$,I,J,A,B,A$,B$,A(7),N
    $
DM 110 C4=48:C6=16:C7=7:Z2=2:Z4=254:Z5=255:Z6=256:
    Z7=127
CJ 120 FA=PEEK(45)+Z6*PEEK(46):BS=PEEK(55)+Z6*PEEK
    (56):H$="0123456789ABCDEF"
SB 130 R$=CHR$(13):L$="{LEFT}":S$=" ":D$=CHR$(20):
    Z$=CHR$(0):T$="{13 RIGHT}"
CQ 140 SD=54272:FOR I=SD TO SD+23:POKE I,0:NEXT:PO
    KE SD+24,15:POKE 788,52
FC 150 PRINT"{CLR}"CHR$(142)CHR$(8):POKE 53280,15:
    POKE 53281,15
EJ 160 PRINT T$ "{RED}{RVS}{2 SPACES}{8 @}
    {2 SPACES}"SPC(28)"{2 SPACES}{OFF}{BLU} MLX
    II {RED}{RVS}{2 SPACES}"SPC(28)"
    {12 SPACES}{BLU}"
FR 170 PRINT"{3 DOWN}{3 SPACES}COMPUTE!'S MACHINE
    {SPACE}LANGUAGE EDITOR{3 DOWN}"
JB 180 PRINT"{BLK}STARTING ADDRESS[4]";:GOSUB300:S
    A=AD:GOSUB1040:IF F THEN180
GF 190 PRINT"{BLK}{2 SPACES}ENDING ADDRESS[4]";:GO
    SUB300:EA=AD:GOSUB1030:IF F THEN190
```

Type It In

```
KR 200 INPUT"{3 DOWN}{BLK}CLEAR WORKSPACE [Y/N]{4}
";A$:IF LEFT$(A$,1)<>"Y"THEN220
PG 210 PRINT"{2 DOWN}{BLU}WORKING...";:FORI=BS TO
{SPACE}BS+EA-SA+7:POKE I,0:NEXT:PRINT"DONE"
DR 220 PRINTTAB(10)"{2 DOWN}{BLK}{RVS} MLX COMMAND
MENU {DOWN}{4}":PRINT T$"{RVS}E{OFF}NTER D
ATA"
BD 230 PRINT T$"{RVS}D{OFF}ISPLAY DATA":PRINT T$"
{RVS}L{OFF}OAD FILE"
JS 240 PRINT T$"{RVS}S{OFF}AVE FILE":PRINT T$"
{RVS}Q{OFF}UIT{2 DOWN}{BLK}"
JH 250 GET A$:IF A$=N$ THEN250
HK 260 A=0:FOR I=1 TO 5:IF A$=MID$("EDLSQ",I,1)THE
N A=I:I=5
FD 270 NEXT:ON A GOTO420,610,690,700,280:GOSUB1060
:GOTO250
EJ 280 PRINT"{RVS} QUIT ":INPUT"{DOWN}{4}ARE YOU S
URE [Y/N]";A$:IF LEFT$(A$,1)<>"Y"THEN220
EM 290 POKE SD+24,0:END
JX 300 IN$=N$:AD=0:INPUTIN$:IFLEN(IN$)<>4THENRETUR
N
KF 310 B$=IN$:GOSUB320:AD=A:B$=MID$(IN$,3):GOSUB32
0:AD=AD*256+A:RETURN
PP 320 A=0:FOR J=1 TO 2:A$=MID$(B$,J,1):B=ASC(A$)-
C4+(A$>"@")*C7:A=A*C6+B
JA 330 IF B<0 OR B>15 THEN AD=0:A=-1:J=2
GX 340 NEXT:RETURN
CH 350 B=INT(A/C6):PRINT MID$(H$,B+1,1);:B=A-B*C6:
PRINT MID$(H$,B+1,1);:RETURN
RR 360 A=INT(AD/Z6):GOSUB350:A=AD-A*Z6:GOSUB350:PR
INT":";
BE 370 CK=INT(AD/Z6):CK=AD-Z4*CK+Z5*(CK>Z7):GOTO39
0
PX 380 CK=CK*Z2+Z5*(CK>Z7)+A
JC 390 CK=CK+Z5*(CK>Z5):RETURN
QS 400 PRINT"{DOWN}STARTING AT{4}";:GOSUB300:IF IN
$<>N$ THEN GOSUB1030:IF F THEN400
EX 410 RETURN
HD 420 PRINT"{RVS} ENTER DATA ":GOSUB400:IF IN$=N$
THEN220
JK 430 OPEN3,3:PRINT
SK 440 POKE198,0:GOSUB360:IF F THEN PRINT IN$:PRIN
T"{UP}{5 RIGHT}";
GC 450 FOR I=0 TO 24 STEP 3:B$=S$:FOR J=1 TO 2:IF
{SPACE}F THEN B$=MID$(IN$,I+J,1)
HA 460 PRINT"{RVS}"B$L$;:IF I<24THEN PRINT"{OFF}";
HD 470 GET A$:IF A$=N$ THEN470
```

Chapter 2

```
FK 480 IF(A$>"/"ANDA$<":")OR(A$>"@"ANDA$<"G")THEN5
40
GS 485 A=- (A$="M")-2*(A$=",")-3*(A$=".")-4*(A$="/"
)-5*(A$="J")-6*(A$="K")
FX 486 A=A-7*(A$="L")-8*(A$=":")-9*(A$="U")-10*(A$
="I")-11*(A$="O")-12*(A$="P")
CM 487 A=A-13*(A$=S$):IF A THEN A$=MID$("ABCD123E4
56F0",A,1):GOTO 540
MP 490 IF A$=R$ AND((I=0)AND(J=1)OR F)THEN PRINT B
$;:J=2:NEXT:I=24:GOTO550
KC 500 IF A$="{HOME}" THEN PRINT B$:J=2:NEXT:I=24:
NEXT:F=0:GOTO440
MX 510 IF(A$="{RIGHT}")ANDF THENPRINT B$L$;:GOTO54
0
GK 520 IF A$<>L$ AND A$<>D$ OR((I=0)AND(J=1))THEN
{SPACE}GOSUB1060:GOTO470
HG 530 A$=L$+S$+L$:PRINT B$L$;:J=2-J:IF J THEN PRI
NT L$;:I=I-3
QS 540 PRINT A$;:NEXT J:PRINT S$;
PM 550 NEXT I:PRINT:PRINT"{UP}{5 RIGHT}";:INPUT#3,
IN$:IF IN$=N$ THEN CLOSE3:GOTO220
QC 560 FOR I=1 TO 25 STEP3:B$=MID$(IN$,I):GOSUB320
:IF I<25 THEN GOSUB380:A(I/3)=A
PK 570 NEXT:IF A<>CK THEN GOSUB1060:PRINT"{BLK}
{RVS} ERROR: REENTER LINE [4]":F=1:GOTO440
HJ 580 GOSUB1080:B=BS+AD-SA:FOR I=0 TO 7:POKE B+I,
A(I):NEXT
QQ 590 AD=AD+8:IF AD>EA THEN CLOSE3:PRINT"{DOWN}
{BLU}** END OF ENTRY **{BLK}{2 DOWN}":GOTO7
00
GQ 600 F=0:GOTO440
QA 610 PRINT"{CLR}{DOWN}{RVS} DISPLAY DATA ":GOSUB
400:IF IN$=N$ THEN220
RJ 620 PRINT"{DOWN}{BLU}PRESS: {RVS}SPACE{OFF} TO
{SPACE}PAUSE, {RVS}RETURN{OFF} TO BREAK[4]
{DOWN}"
KS 630 GOSUB360:B=BS+AD-SA:FORI=BTO B+7:A=PEEK(I):
GOSUB350:GOSUB380:PRINT S$;
CC 640 NEXT:PRINT"{RVS}";:A=CK:GOSUB350:PRINT
KH 650 F=1:AD=AD+8:IF AD>EA THENPRINT"{DOWN}{BLU}*
* END OF DATA **":GOTO220
KC 660 GET A$:IF A$=R$ THEN GOSUB1080:GOTO220
EQ 670 IF A$=S$ THEN F=F+1:GOSUB1080
AD 680 ONFGOTO630,660,630
CM 690 PRINT"{DOWN}{RVS} LOAD DATA ":OP=1:GOTO710
PC 700 PRINT"{DOWN}{RVS} SAVE FILE ":OP=0
RX 710 IN$=N$:INPUT"{DOWN}FILENAME[4]";:IN$:IF IN$=
N$ THEN220
PR 720 F=0:PRINT"{DOWN}{BLK}{RVS}T{OFF}APE OR
{RVS}D{OFF}ISK: [4]";
```


Type It In

```

FP 730 GET A$:IF A$="T"THEN PRINT"T{DOWN}":GOTO880
HQ 740 IF A$<>"D"THEN730
HH 750 PRINT"D{DOWN}":OPEN15,8,15,"I0:":B=EA-SA:IN
    $="0:"+IN$:IF OP THEN810
SQ 760 OPEN 1,8,8,IN$+",P,W":GOSUB860:IF A THEN220
FJ 770 AH=INT(SA/256):AL=SA-(AH*256):PRINT#1,CHR$(
    AL);CHR$(AH);
PE 780 FOR I=0 TO B:PRINT#1,CHR$(PEEK(BS+I));:IF S
    T THEN800
FC 790 NEXT:CLOSE1:CLOSE15:GOTO940
GS 800 GOSUB1060:PRINT"{DOWN}{BLK}ERROR DURING SAV
    E:[4]":GOSUB860:GOTO220
MA 810 OPEN 1,8,8,IN$+",P,R":GOSUB860:IF A THEN220
GE 820 GET#1,A$,B$:AD=ASC(A$+Z$)+256*ASC(B$+Z$):IF
    AD<>SA THEN F=1:GOTO850
RX 830 FOR I=0 TO B:GET#1,A$:POKE BS+I,ASC(A$+Z$):
    IF(I<>B)AND ST THEN F=2:AD=I:I=B
FA 840 NEXT:IF ST<>64 THEN F=3
FQ 850 CLOSE1:CLOSE15:ON ABS(F>0)+1 GOTO960,970
SA 860 INPUT#15,A,A$:IF A THEN CLOSE1:CLOSE15:GOSU
    B1060:PRINT"{RVS}ERROR: "A$
GQ 870 RETURN
EJ 880 POKE183,PEEK(FA+2):POKE187,PEEK(FA+3):POKE1
    88,PEEK(FA+4):IFOP=0THEN920
HJ 890 SYS 63466:IF(PEEK(783)AND1)THEN GOSUB1060:P
    RINT"[DOWN]{RVS} FILE NOT FOUND ":GOTO690
CS 900 AD=PEEK(829)+256*PEEK(830):IF AD<>SA THEN F
    =1:GOTO970
SC 910 A=PEEK(831)+256*PEEK(832)-1:F=F-2*(A<EA)-3*
    (A>EA):AD=A-AD:GOTO930
KM 920 A=SA:B=EA+1:GOSUB1010:POKE780,3:SYS 63338
JF 930 A=BS:B=BS+(EA-SA)+1:GOSUB1010:ON OP GOTO950
    :SYS 63591
AE 940 GOSUB1080:PRINT"{BLU}** SAVE COMPLETED **":
    GOTO220
XP 950 POKE147,0:SYS 63562:IF ST>0 THEN970
FR 960 GOSUB1080:PRINT"{BLU}** LOAD COMPLETED **":
    GOTO220
DP 970 GOSUB1060:PRINT"{BLK}{RVS}ERROR DURING LOAD
    :{DOWN}[4]":ON F GOSUB980,990,1000:GOTO220
PP 980 PRINT"INCORRECT STARTING ADDRESS (":GOSUB3
    60:PRINT")":RETURN
GR 990 PRINT"LOAD ENDED AT ":AD=SA+AD:GOSUB360:PR
    INT D$:RETURN
FD 1000 PRINT"TRUNCATED AT ENDING ADDRESS":RETURN
RX 1010 AH=INT(A/256):AL=A-(AH*256):POKE193,AL:POK
    E194,AH
FF 1020 AH=INT(B/256):AL=B-(AH*256):POKE174,AL:POK
    E175,AH:RETURN
FX 1030 IF AD<SA OR AD>EA THEN1050

```

Chapter 2

```
HA 1040 IF(AD>511 AND AD<40960)OR(AD>49151 AND AD<
53248)THEN GOSUB1080:F=0:RETURN
HC 1050 GOSUB1060:PRINT"{RVS} INVALID ADDRESS
{DOWN}{BLK}":F=1:RETURN
AR 1060 POKE SD+5,31:POKE SD+6,208:POKE SD,240:POK
E SD+1,4:POKE SD+4,33
DX 1070 FOR S=1 TO 100:NEXT:GOTO1090
PF 1080 POKE SD+5,8:POKE SD+6,240:POKE SD,0:POKE S
D+1,90:POKE SD+4,17
AC 1090 FOR S=1 TO 100:NEXT:POKE SD+4,0:POKE SD,0:
POKE SD+1,0:RETURN
```

Program 1

Space Arena

Once you've typed in and saved a copy of "MLX" to disk or tape, load and run it. You'll be asked to provide two addresses before you can begin entering "Space Arena." Respond to the prompts with

Starting Address: C000

Ending Address: CF8F

Space Arena

For mistake-proof program entry, use "MLX: Machine Language Entry Program" to type in this program.

```
C000:A9 37 85 FC A9 D7 85 FE 5B
C008:A0 00 84 FB 84 FD 78 A5 DC
C010:01 29 FB 85 01 B1 FD 91 90
C018:FB 88 D0 F9 C6 FE C6 FC 30
C020:A5 FC C9 2F D0 EF A0 2F 96
C028:B9 36 CF 99 00 34 88 10 99
C030:F7 A5 01 09 04 85 01 58 58
C038:A9 1C 8D 18 D0 A0 00 84 56
C040:FB A9 DD 85 FD A9 20 85 9A
C048:FC A9 CB 85 FE B1 FD F0 30
C050:1D 91 FB E6 FB D0 02 E6 C0
C058:FC E6 FD D0 02 E6 FE A5 2E
C060:FE C9 CF D0 E8 A5 FD C9 7E
C068:36 D0 E2 4C 8C C0 E6 FD 8D
C070:D0 02 E6 FE B1 FD A8 A9 28
C078:00 88 91 FB D0 FB B1 FD E5
C080:18 65 FB 85 FB 90 02 E6 4C
C088:FC 4C 59 C0 A0 18 A9 00 8B
C090:99 00 D4 88 10 FA A0 18 C7
C098:B9 38 C7 99 00 D4 88 10 0C
C0A0:F7 A0 00 B9 DD C7 99 00 23
C0A8:04 B9 DD C8 99 00 05 B9 73
C0B0:DD C9 99 00 06 B9 DD CA 64
C0B8:99 00 07 A9 0F 99 00 D8 3A
C0C0:99 00 D9 99 00 DA 99 00 82
C0C8:DB 88 D0 D7 20 FF C6 A0 21
C0D0:26 B9 66 CF 99 28 D8 88 45
C0D8:D0 F7 A9 00 8D 20 D0 8D 12
C0E0:21 D0 AD 36 C7 09 30 8D 90
C0E8:5E 07 AD 37 C7 09 30 8D D4
C0F0:0E 07 20 E4 FF C9 85 F0 B0
```

Chapter 2

C0F8:36 C9 86 F0 1B C9 87 D0 C7
C100:F1 EE 36 C7 AD 36 C7 C9 1B
C108:06 D0 05 A9 01 8D 36 C7 70
C110:09 30 8D 5E 07 4C F2 C0 CB
C118:EE 37 C7 AD 37 C7 C9 04 25
C120:D0 05 A9 01 8D 37 C7 09 74
C128:30 8D 0E 07 4C F2 C0 A0 A9
C130:03 A9 00 99 32 C7 88 10 0B
C138:FA 20 FF C6 A9 00 8D 08 1E
C140:DC 8D 09 DC 8D 0A DC A9 7C
C148:A0 85 FD A9 04 85 FE A2 AE
C150:13 A0 26 B1 FD 30 04 A9 C7
C158:20 91 FD 88 D0 F5 A5 FD 40
C160:18 69 28 85 FD 90 02 E6 C4
C168:FE CA D0 E5 A0 02 B9 00 17
C170:00 99 00 08 C8 D0 F7 A9 FD
C178:00 8D 0E DC A2 40 BD A8 29
C180:C7 95 10 CA 10 F8 A9 84 38
C188:8D F8 07 A9 8C 8D F9 07 22
C190:A9 00 85 47 85 4E A0 18 CC
C198:88 99 00 D4 D0 FA A9 0F E8
C1A0:8D 18 D4 A9 80 8D 14 D4 5D
C1A8:A9 81 8D 12 D4 A0 07 A9 15
C1B0:0C 99 27 D0 88 10 FA A9 B6
C1B8:0E 8D 27 D0 8D 2D D0 A9 05
C1C0:05 8D 28 D0 8D 2E D0 A9 AC
C1C8:3C 8D 1C D0 A9 0B 8D 25 18
C1D0:D0 A9 0F 8D 26 D0 AD 37 E8
C1D8:C7 0A 0A 0A 0A 85 02 A9 B8
C1E0:60 38 E5 02 8D 07 DC 8D 4F
C1E8:06 DC A9 09 8D 0F DC A9 78
C1F0:3F 8D 15 D0 A9 01 2C 0F DF
C1F8:DC D0 FB A9 09 8D 0F DC B2
C200:AD 1E D0 85 45 29 C3 D0 7D
C208:03 4C DC C2 10 51 A5 45 40
C210:29 01 F0 35 A2 00 A0 0E 40
C218:20 5A C4 D0 2C A9 01 20 14
C220:EE C6 A2 01 B5 3E 30 04 3E
C228:4A 4C 2E C2 38 6A 18 75 E8
C230:30 10 08 C9 C0 B0 0A A9 F5
C238:C0 D0 06 C9 41 90 02 A9 A9
C240:40 95 30 CA F0 DE 4C 50 E9
C248:C2 A2 0E 20 D7 C4 D0 0F 1E
C250:AD 15 D0 29 7F 8D 15 D0 CB
C258:A0 00 84 1F 20 28 C5 A5 83
C260:45 29 40 F0 53 A5 45 29 CE
C268:02 F0 35 A2 02 A0 0C 20 C6
C270:5A C4 D0 2C A9 01 20 EE B1
C278:C6 A2 01 B5 3C 30 04 4A 7A
C280:4C 85 C2 38 6A 18 75 32 3A
C288:10 08 C9 C0 B0 0A A9 C0 1F

C290:D0 06 C9 41 90 02 A9 40 6D
 C298:95 32 CA F0 DE 4C A7 C2 18
 C2A0:A2 0C 20 D7 C4 D0 11 AD 35
 C2A8:15 D0 29 BF 8D 15 D0 A9 1A
 C2B0:00 85 1D A0 07 20 28 C5 14
 C2B8:A5 45 29 02 F0 0C A2 02 A6
 C2C0:20 D7 C4 D0 05 A0 01 4C EA
 C2C8:5E C5 A5 45 29 01 F0 0C 33
 C2D0:A2 00 20 D7 C4 D0 05 A0 3D
 C2D8:00 4C 5E C5 20 0B C6 C6 1B
 C2E0:40 D0 7A A9 07 85 40 A2 16
 C2E8:01 BD 00 DC 29 04 D0 05 2C
 C2F0:D6 41 4C FE C2 BD 00 DC 95
 C2F8:29 08 D0 0D F6 41 B5 41 69
 C300:29 0F 95 41 09 80 9D F8 25
 C308:07 BD 00 DC 29 01 D0 28 67
 C310:B4 41 8A 0A AA B5 30 18 D8
 C318:79 75 C7 C9 41 90 04 C9 6D
 C320:C0 90 02 95 30 B5 31 18 98
 C328:79 71 C7 C9 41 90 04 C9 7C
 C330:C0 90 02 95 31 8A 4A AA C8
 C338:CA F0 AE AD 00 DC 29 01 D8
 C340:85 02 AD 01 DC 29 01 F0 4F
 C348:04 A4 02 D0 0B 18 69 10 E3
 C350:E5 02 8D 0F D4 4C 5D C3 44
 C358:A9 00 8D 0F D4 A5 43 D0 EB
 C360:4E AD 15 D0 29 BF 8D 15 A2
 C368:D0 AD 00 DC 29 10 D0 41 FD
 C370:A0 00 20 43 C5 A9 80 85 DB
 C378:43 A5 10 18 69 20 85 1C 81
 C380:A5 11 69 01 85 1D A5 20 68
 C388:18 69 40 85 2C A5 21 69 7A
 C390:01 85 2D A4 41 B9 75 C7 8D
 C398:0A 0A 0A 0A 85 3C B9 71 8B
 C3A0:C7 0A 0A 0A 0A 85 3D AD FE
 C3A8:15 D0 09 40 8D 15 D0 C6 3D
 C3B0:43 A5 44 D0 4E AD 15 D0 FC
 C3B8:29 7F 8D 15 D0 AD 01 DC D3
 C3C0:29 10 D0 41 A0 07 20 43 B3
 C3C8:C5 A9 80 85 44 A5 12 18 FA
 C3D0:69 50 85 1E A5 13 69 01 01
 C3D8:85 1F A5 22 18 69 40 85 2E
 C3E0:2E A5 23 69 01 85 2F A4 05
 C3E8:42 B9 75 C7 0A 0A 0A 0A C1
 C3F0:85 3E B9 71 C7 0A 0A 0A 9D
 C3F8:0A 85 3F AD 15 D0 09 80 28
 C400:8D 15 D0 C6 44 AE 0A DC E9
 C408:AD 09 DC F0 09 E8 A9 60 F4
 C410:F8 38 ED 09 DC D8 85 02 C9
 C418:4A 4A 4A 4A 09 30 8D 3C A7
 C420:04 A5 02 29 0F 09 30 8D 72

Chapter 2

C428:3D 04 86 02 AD 36 C7 38 50
C430:E5 02 09 30 8D 3A 04 AD 5C
C438:0A DC CD 36 C7 F0 03 4C 6F
C440:F4 C1 A9 00 8D 15 D0 A0 EC
C448:02 B9 00 08 99 00 00 C8 57
C450:D0 F7 A9 01 8D 0E DC 4C 30
C458:8C C0 B5 10 38 F9 10 00 D9
C460:85 FD B5 11 F9 11 00 85 8D
C468:FE 90 0F A5 FD F9 51 C7 14
C470:A5 FE F9 52 C7 B0 5D 4C F8
C478:95 C4 B9 10 00 38 F5 10 13
C480:85 FD B9 11 00 F5 11 85 14
C488:FE A5 FD FD 51 C7 A5 FE 8E
C490:FD 52 C7 B0 3F B5 20 38 FA
C498:F9 20 00 85 FD B5 21 38 C0
C4A0:F9 21 00 85 FE 90 0F A5 C5
C4A8:FD F9 61 C7 A5 FE F9 62 D7
C4B0:C7 B0 21 4C D1 C4 B9 20 68
C4B8:00 38 F5 20 85 FD B9 21 C9
C4C0:00 F5 21 85 FE A5 FD FD CC
C4C8:61 C7 A5 FE FD 62 C7 B0 53
C4D0:03 A9 00 60 A9 01 60 A5 04
C4D8:45 0A 0A 85 02 A0 0A 06 CD
C4E0:02 90 05 20 5A C4 F0 09 03
C4E8:88 88 C0 02 D0 F1 A9 01 B3
C4F0:60 A9 01 85 02 C8 E8 B5 48
C4F8:30 30 07 4A 4A 4A 4A 4C 88
C500:08 C5 4A 4A 4A 4A 09 F0 6D
C508:18 79 30 00 10 08 C9 C0 F8
C510:B0 0A A9 C0 D0 06 C9 41 2B
C518:90 02 A9 40 99 30 00 CA FD
C520:88 C6 02 F0 D2 A9 00 60 8E
C528:A9 00 99 06 D4 99 47 00 B7
C530:99 04 D4 A9 0A 99 01 D4 4C
C538:A9 35 99 05 D4 A9 21 99 92
C540:04 D4 60 A9 00 99 06 D4 F0
C548:99 04 D4 A9 28 99 47 00 0D
C550:99 01 D4 A9 09 99 05 D4 AB
C558:A9 11 99 04 D4 60 A9 90 7C
C560:99 F8 07 B9 85 C7 8D 05 DE
C568:C6 E8 A9 01 85 48 A9 20 97
C570:20 EE C6 A0 07 A9 00 88 12
C578:99 0E D4 D0 FA A9 0A 8D 1C
C580:13 D4 A9 08 8D 0F D4 A9 7C
C588:81 8D 12 D4 A9 1E 85 46 DE
C590:20 0B C6 A9 01 2C 0F DC 16
C598:D0 FB A9 09 8D 0F DC C6 7A
C5A0:46 D0 ED A9 80 8D 12 D4 0F
C5A8:A9 8C 8D 13 D4 A9 11 8D 0C
C5B0:0F D4 A9 A0 8D 16 D4 A9 50
C5B8:84 8D 17 D4 A9 1F 8D 18 17

C5C0:D4 A9 81 8D 12 D4 A9 3C 9D
C5C8:8D 15 D0 A9 46 85 46 20 0A
C5D0:0B C6 20 F2 C5 A9 01 2C C9
C5D8:0F DC D0 FB A9 09 8D 0F 98
C5E0:DC C6 46 D0 EA A9 00 8D ED
C5E8:17 D4 A9 0F 8D 18 D4 4C 1E
C5F0:7C C1 C6 48 D0 14 A9 08 BA
C5F8:85 48 A0 0A AD 87 C7 49 72
C600:01 8D 87 C7 99 00 04 88 3C
C608:10 FA 60 A2 0E B5 30 10 4A
C610:1D C9 FE B0 3E 75 10 95 07
C618:10 B5 11 E9 00 95 11 C9 1E
C620:05 90 28 D0 2E B5 10 C9 98
C628:40 B0 28 4C 4B C6 C9 03 D7
C630:90 21 18 75 10 95 10 B5 55
C638:11 69 00 95 11 DD 89 C7 DC
C640:90 11 D0 07 B5 10 DD 88 17
C648:C7 90 08 B5 30 49 FF A8 89
C650:C8 94 30 B5 10 85 02 A0 04
C658:08 B5 11 C9 30 90 02 E9 C7
C660:30 26 49 06 02 2A C9 30 95
C668:90 04 E9 30 E6 02 88 D0 A0
C670:F2 A5 02 9D 00 D0 B5 31 DA
C678:10 1D C9 FE B0 3E 75 20 08
C680:95 20 B5 21 E9 00 95 21 45
C688:C9 11 90 28 D0 2E B5 20 9E
C690:C9 80 B0 28 4C B4 C6 C9 48
C698:03 90 21 18 75 20 95 20 E8
C6A0:B5 21 69 00 95 21 DD 99 05
C6A8:C7 90 11 D0 07 B5 20 DD 9A
C6B0:98 C7 90 08 B5 31 49 FF 14
C6B8:A8 C8 94 31 B5 21 85 02 B1
C6C0:B5 20 A0 06 46 02 6A 88 3D
C6C8:D0 FA 9D 01 D0 CA CA 30 B8
C6D0:03 4C 0D C6 A5 49 8D 10 7E
C6D8:D0 A4 47 F0 06 88 84 47 92
C6E0:8C 01 D4 A4 4E F0 06 88 A4
C6E8:84 4E 8C 08 D4 60 F8 18 90
C6F0:7D 32 C7 9D 32 C7 BD 33 FB
C6F8:C7 69 00 9D 33 C7 D8 A0 A9
C700:01 AD 34 C7 85 FD AD 35 33
C708:C7 85 FE A2 04 A9 00 06 B3
C710:FD 26 FE 2A CA D0 F8 09 3F
C718:30 99 28 04 C8 C0 05 D0 8F
C720:0C A0 23 AD 32 C7 85 FD D6
C728:AD 33 C7 85 FE C0 27 D0 C6
C730:DA 60 00 00 00 00 01 01 48
C738:00 06 00 00 11 0F 00 04 12
C740:03 00 00 15 0F 00 04 0C 2F
C748:00 00 15 0F 00 00 00 00 6B
C750:0F 30 03 30 03 80 04 80 79

Chapter 2

C758:04 80 04 80 04 F0 00 F0 67
C760:00 00 03 00 03 C0 04 C0 34
C768:04 C0 04 C0 04 00 01 00 D8
C770:01 FA FB FC FE 00 02 04 8E
C778:05 06 05 04 02 00 FE FC F7
C780:FB FA FB FC FE 3F 2E 85 F2
C788:F0 3C F0 3C 70 3B 70 3B 0E
C790:70 3B 70 3B 30 3F 30 3F 07
C798:00 3B 00 3B 00 39 00 39 C8
C7A0:00 39 00 39 00 3D 00 3D 44
C7A8:90 06 D0 3B 00 21 00 21 75
C7B0:00 21 00 21 00 00 00 00 9A
C7B8:C0 25 C0 25 40 16 00 20 D6
C7C0:C0 29 80 33 00 00 00 00 3E
C7C8:00 00 00 00 00 00 00 00 58
C7D0:00 00 00 00 00 00 00 00 60
C7D8:01 04 0C 00 00 55 40 40 81
C7E0:40 40 72 40 40 40 40 40 B6
C7E8:40 40 40 40 40 40 72 40 DC
C7F0:40 40 40 72 40 40 40 40 A3
C7F8:40 40 40 40 40 40 40 72 BA
C800:40 40 40 40 49 5D 30 30 1E
C808:30 30 5D 84 84 84 84 84 75
C810:84 84 84 84 84 84 5D 30 FE
C818:3A 30 30 5D 84 84 84 84 72
C820:84 84 84 84 84 84 84 5D 8A
C828:30 30 30 30 5D 6B 40 40 40
C830:40 40 71 40 40 40 40 40 E7
C838:40 40 40 40 40 40 71 40 2C
C840:40 40 40 71 40 40 40 40 E4
C848:40 40 40 40 40 40 40 71 0B
C850:40 40 40 40 73 5D 20 20 8F
C858:20 20 20 20 20 20 20 20 E9
C860:80 20 20 20 20 20 20 20 22
C868:20 20 20 20 20 20 20 20 F9
C870:20 20 20 20 20 20 20 20 02
C878:20 20 20 20 5D 5D 20 20 E8
C880:20 20 20 20 20 20 20 20 12
C888:20 20 20 20 20 20 20 20 1A
C890:20 20 20 20 20 20 20 20 22
C898:20 20 20 20 20 20 20 20 2A
C8A0:20 20 20 20 5D 5D 20 20 11
C8A8:20 80 20 20 20 20 13 20 38
C8B0:10 20 01 20 03 20 05 20 37
C8B8:20 20 01 20 12 20 05 20 BF
C8C0:0E 20 01 20 20 20 20 20 65
C8C8:20 20 20 20 5D 5D 20 20 39
C8D0:20 20 20 20 20 80 63 63 AD
C8D8:63 63 63 63 63 63 63 63 6A
C8E0:63 63 63 63 63 63 63 63 72
C8E8:63 63 63 20 20 20 20 20 55

Type It In

C8F0:81 20 20 20 5D 5D 20 20 12
C8F8:20 20 20 20 20 20 20 20 8A
C900:20 02 19 3A 20 02 12 19 31
C908:01 0E 20 06 09 0C 05 13 99
C910:20 20 20 20 20 20 20 20 A3
C918:20 20 20 20 5D 5D 20 20 8A
C920:20 20 20 81 20 20 20 20 C9
C928:20 20 20 20 20 20 20 20 BB
C930:20 20 20 20 20 20 20 20 C3
C938:83 20 20 20 20 20 20 20 7D
C940:20 20 20 20 5D 5D 20 20 B2
C948:20 20 20 20 20 20 20 20 DB
C950:20 20 20 20 20 20 82 20 A8
C958:20 20 20 20 20 20 20 20 EB
C960:20 20 20 80 20 80 20 60 BB
C968:20 20 20 20 5D 5D 20 20 DA
C970:20 20 20 20 20 20 20 20 04
C978:20 20 20 20 20 20 20 20 0C
C980:20 20 20 20 20 20 20 20 14
C988:80 20 20 20 20 20 20 20 4C
C990:20 20 20 20 5D 5D 20 20 03
C998:20 80 20 20 20 20 06 31 21
C9A0:20 2D 20 13 14 01 12 14 A1
C9A8:20 07 01 0D 05 20 20 20 08
C9B0:20 20 20 20 20 20 20 20 44
C9B8:20 20 20 20 5D 5D 20 20 2B
C9C0:20 20 20 20 20 20 20 81 B5
C9C8:20 20 20 20 20 20 20 20 5C
C9D0:20 20 20 20 20 20 20 20 64
C9D8:20 20 20 20 20 20 20 20 6C
C9E0:20 20 20 20 5D 5D 20 20 53
C9E8:20 20 20 20 20 20 06 33 5B
C9F0:20 2D 20 03 08 01 0E 07 7B
C9F8:05 20 13 10 05 05 04 20 DE
CA00:20 20 20 20 20 80 83 20 DD
CA08:20 20 20 20 5D 5D 20 83 DF
CA10:20 20 20 20 20 20 20 80 06
CA18:20 20 81 20 20 20 20 20 D9
CA20:20 20 20 20 20 20 20 20 B5
CA28:20 20 80 20 20 20 20 20 C9
CA30:20 20 20 20 5D 5D 20 81 06
CA38:20 20 20 20 20 20 06 35 AE
CA40:20 2D 20 03 08 01 0E 07 CC
CA48:05 20 14 09 0D 05 20 0C 44
CA50:09 0D 09 14 20 20 20 20 F1
CA58:20 20 20 20 5D 5D 20 20 CC
CA60:20 20 20 20 20 20 20 20 F5
CA68:20 20 80 20 20 20 20 20 0A
CA70:20 20 20 20 20 20 20 20 06
CA78:20 20 20 20 20 20 20 20 0E
CA80:20 20 20 20 5D 5D 20 20 F4

Chapter 2

CAB8:20 20 20 20 20 20 20 20 1E
CA90:20 20 20 20 20 20 20 20 26
CA98:20 20 20 80 20 20 20 20 34
CAA0:20 20 20 20 20 20 20 20 36
CAA8:20 20 80 20 5D 5D 20 20 29
CAB0:81 20 20 20 20 20 20 20 F6
CAB8:20 20 20 20 20 20 20 20 4E
CAC0:20 20 81 20 20 20 20 20 82
CAC8:82 20 20 20 20 20 20 20 8F
CAD0:20 20 20 80 5D 5D 20 20 4B
CAD8:20 20 20 20 20 20 20 20 6E
CAE0:20 20 20 13 10 05 05 04 66
CAE8:20 3D 20 35 20 81 20 20 9C
CAF0:20 20 80 20 20 81 20 20 18
CAF8:20 20 20 20 5D 5D 20 20 6D
CB00:20 20 20 20 20 20 20 20 97
CB08:20 20 20 20 20 20 20 20 9F
CB10:20 20 20 20 20 20 20 20 A7
CB18:20 20 20 20 20 20 20 20 AF
CB20:20 20 20 20 5D 5D 20 20 96
CB28:20 20 20 20 20 20 20 80 20
CB30:20 20 20 20 14 09 0D 05 C9
CB38:20 3D 20 32 3A 30 30 20 69
CB40:20 80 20 20 20 20 20 20 EF
CB48:20 20 20 20 5D 5D 20 83 22
CB50:20 20 20 20 20 20 20 20 E7
CB58:20 20 20 20 81 20 20 20 FA
CB60:80 20 20 20 20 20 20 20 28
CB68:20 20 20 20 20 20 20 20 FF
CB70:20 20 20 20 5D 5D 20 20 E6
CB78:20 20 20 20 20 20 20 20 10
CB80:20 20 20 20 20 20 20 20 18
CB88:20 20 20 20 20 20 20 20 20
CB90:20 20 20 80 20 20 20 20 2E
CB98:20 20 20 20 5D 4A 40 40 23
CBA0:40 40 40 40 40 40 40 40 38
CBA8:40 40 40 40 40 40 40 40 40
CBB0:40 40 40 40 40 40 40 40 48
CBB8:40 40 40 40 40 40 40 40 50
CBC0:40 40 40 40 4B 00 00 00 EE
CBC8:00 00 00 00 00 00 00 00 60
CBD0:00 00 00 00 00 80 80 92 FD
CBD8:93 94 95 91 91 01 80 00 BC
CBE0:01 03 C0 00 01 03 C0 00 67
CBE8:01 07 E0 00 01 07 E0 00 C4
CBF0:01 0F F0 00 01 0F F0 00 11
CBF8:01 1F F8 00 01 1F F8 00 6E
CC00:01 30 0C 00 24 30 00 02 8B
CC08:70 00 02 F0 00 01 01 F0 20
CC10:00 01 03 F0 00 01 07 F0 5C
CC18:00 01 0F F0 00 01 1F F0 16

Type It In

CC20:00 01 37 F0 00 02 B0 00 59
CC28:02 30 00 24 0C 00 02 3C B1
CC30:00 02 FC 00 01 03 F8 00 EF
CC38:01 0F F8 00 01 3F F8 00 2C
CC40:01 07 F0 00 01 01 F0 00 28
CC48:02 F0 00 02 60 00 02 60 A6
CC50:00 29 7F FE 00 01 1F FC 53
CC58:00 01 0F F8 00 01 0F F0 B6
CC60:00 01 07 E0 00 01 03 C0 F3
CC68:00 01 01 80 00 01 03 00 74
CC70:24 38 00 02 1E 00 02 1F 5E
CC78:C0 00 01 1F F0 00 01 1F 2D
CC80:FE 00 01 1F F0 00 01 1F 54
CC88:C0 00 01 1E 00 02 38 00 FC
CC90:24 03 00 02 01 80 00 01 28
CC98:03 C0 00 01 07 E0 00 01 B0
CCA0:0F F0 00 01 0F F8 00 01 6B
CCA8:1F FC 00 01 7F FE 00 27 40
CCB0:C0 00 02 C0 00 01 01 E0 DD
CCB8:00 01 03 E0 00 01 0F E0 04
CCC0:00 01 7F F0 00 01 1F F0 CC
CCC8:00 01 07 F0 00 01 01 F8 91
CCD0:00 02 78 00 02 18 00 24 8E
CCD8:30 00 02 B0 00 01 37 F0 39
CCE0:00 01 1F F0 00 01 0F F0 C0
CCE8:00 01 07 F0 00 01 03 F0 AD
CCF0:00 01 01 F0 00 02 F0 00 E3
CCF8:02 70 00 02 30 00 23 30 C7
CD00:0C 00 01 1F F8 00 01 1F 9C
CD08:F8 00 01 0F F0 00 01 0F C9
CD10:F0 00 01 07 E0 00 01 07 C4
CD18:E0 00 01 03 C0 00 01 03 7F
CD20:C0 00 01 01 80 00 20 0C 9C
CD28:00 02 0D 00 02 0F EC 00 0C
CD30:01 0F F8 00 01 0F F0 00 55
CD38:01 0F E0 00 01 0F C0 00 F9
CD40:01 0F 80 00 01 0F 00 02 76
CD48:0E 00 02 0C 00 1E 03 00 6A
CD50:02 03 00 02 07 80 00 01 09
CD58:07 C0 00 01 07 F0 00 01 B4
CD60:0F FE 00 01 0F F8 00 01 B0
CD68:0F E0 00 01 1F 80 00 01 CF
CD70:1E 00 02 18 00 25 C0 00 F2
CD78:01 01 80 00 01 03 C0 00 7A
CD80:01 07 E0 00 01 0F F0 00 A0
CD88:01 1F F0 00 01 3F F8 00 81
CD90:01 7F FE 00 2D 1C 00 02 48
CD98:78 00 01 03 F8 00 01 0F 99
CDA0:F8 00 01 7F F8 00 01 0F A9
CDA8:F8 00 01 03 F8 00 02 78 55
CDB0:00 02 1C 00 2C 7F FE 00 AD

Chapter 2

CDB8:01 3F F8 00 01 1F F0 00 2A
CDC0:01 0F F0 00 01 07 E0 00 A4
CDC8:01 03 C0 00 01 01 80 00 CA
CDD0:02 C0 00 20 30 00 02 3C 61
CDD8:00 02 3F 00 02 1F C0 00 EA
CDE0:01 1F F0 00 01 1F FC 00 61
CDE8:01 0F E0 00 01 0F 80 00 2A
CDF0:01 0F 00 02 06 00 02 06 2B
CDF8:00 1E 0C 00 02 0E 00 02 E7
CE00:0F 00 02 0F 80 00 01 0F 6B
CE08:C0 00 01 0F E0 00 01 0F 2F
CE10:F0 00 01 0F F8 00 01 0F 10
CE18:EC 00 01 0D 00 02 0C 00 3D
CE20:21 08 10 00 01 04 32 00 CE
CE28:01 06 14 00 01 14 A4 00 EB
CE30:01 18 CC 00 01 03 00 02 04
CE38:6D 1C 00 01 08 4A 00 02 0F
CE40:C0 00 01 09 30 00 01 01 73
CE48:00 03 80 00 1D 60 00 02 23
CE50:F0 00 02 60 00 3A FF 00 95
CE58:01 0F BE F0 0E FF F0 3F B2
CE60:F9 7C 3F F9 BC 3D 6A B8 0A
CE68:F5 7B DA B5 BA 6A F5 B6 B8
CE70:A6 F6 B6 A6 FB DA 56 FA 53
CE78:6A 5A 3A AA 58 3A A5 A8 73
CE80:16 96 58 0B 56 60 0B 59 2E
CE88:A0 00 01 AA 00 0C FF 00 71
CE90:01 0F EF B0 0B FF B0 3D 72
CE98:7F FC 3D B5 AC 2F E5 B8 DE
CEA0:FF A6 FA FF AB DA E5 AA 86
CEA8:69 F6 EA AA FA EF 96 FB 81
CEB0:75 96 3A B5 98 3A 65 98 62
CEB8:25 9A 58 09 55 60 0A 55 C0
CEC0:A0 00 01 AA 00 0C FF 00 A9
CEC8:01 03 FF F0 0E D7 F0 0E 76
CED0:DB FC 3B FF AC 3B EA 68 93
CED8:EF AE BA EF B5 AA FE D5 9C
CEE0:A6 FE D5 E6 F5 96 D6 F6 69
CEE8:EB E6 3B EF 58 3A AA 58 F5
CEF0:2A 99 98 09 99 60 0A 55 65
CEF8:A0 00 01 AA 00 0C FF 00 E1
CF00:01 0F FB F0 0E FF F0 3B FF
CF08:FF FC 3B 5A AC 3F 6E 98 CB
CF10:6F 6D AE BF B6 AA FE 6A 5D
CF18:66 F6 BA B6 DB AE 96 FE 31
CF20:B5 56 3A B5 98 39 A5 98 60
CF28:2A E5 98 09 5A 60 0A 55 B7
CF30:A0 00 01 AA 00 0B 00 00 17
CF38:00 10 00 00 00 00 00 00 DB
CF40:00 18 00 00 00 00 20 04 2A
CF48:00 3A 5C 00 20 04 00 10 23

Type It In

CF50:49 18 18 92 08 00 33 CC 3A
CF58:33 CC 33 CC 33 CC 66 99 2B
CF60:66 99 66 99 66 99 0F 0E C5
CF68:0E 0E 0E 0F 0E 0E 0E 0E 18
CF70:0E 0E 0E 0E 0E 0E 0E 0F 11
CF78:04 04 04 04 0F 05 05 05 77
CF80:05 05 05 05 05 05 05 05 20
CF88:0F 05 05 05 05 0F 00 00 46

Program 2

Saloon Shootout

Make sure you have a copy of "MLX" entered and saved to tape or disk before you begin typing in "Saloon Shootout." Load and run MLX, and answer the prompts with

Starting Address: 2800

Ending Address: 3BC7

When you've finished typing, be sure to save a copy of the data before you exit MLX.

Saloon Shootout

For mistake-proof program entry, use "MLX: Machine Language Entry Program" to type in this program.

```
2800:4C 1A 28 00 00 00 00 00 02
2808:00 00 00 00 00 00 00 00 58
2810:00 00 00 00 00 00 00 00 60
2818:00 00 A9 00 8D 5C 03 8D 0F
2820:5D 03 A9 93 20 D2 FF 20 BA
2828:B2 33 20 4C 2F 20 5D 34 50
2830:20 AD 2E 20 70 38 20 18 80
2838:2E 20 FC 29 CE 0B 28 D0 9D
2840:0C AD 0C 28 8D 0B 28 20 0F
2848:08 30 20 5F 2A CE 09 28 69
2850:D0 0C AD 0A 28 8D 09 28 14
2858:20 8D 30 20 2D 2A CE 61 35
2860:03 D0 09 AD 62 03 8D 61 FD
2868:03 20 F7 2A CE 0D 28 D0 AF
2870:09 AD 0E 28 8D 0D 28 20 06
2878:F7 30 AD 0D 28 C9 0A B0 84
2880:0F AD 66 03 C9 01 D0 08 BC
2888:AD 11 FF 29 4F 8D 11 FF 59
2890:CE 0F 28 D0 09 AD 10 28 65
2898:8D 0F 28 20 40 2D CE 18 E6
28A0:28 D0 09 AD 19 28 8D 18 D1
28A8:28 20 DD 37 20 EE 28 CE 20
28B0:57 03 D0 88 CE 54 03 20 FD
28B8:6E 29 AD 54 03 F0 08 AD 1F
28C0:58 03 F0 03 4C 3C 28 A9 99
28C8:00 8D 11 FF A2 17 A0 08 59
28D0:18 20 F0 FF A9 BD A0 37 10
28D8:20 08 34 20 2D 2A 20 6E 84
28E0:29 20 FC 29 20 E4 FF C9 5E
```

Type It In

28E8:0D D0 F9 4C 22 28 A2 00 EE
28F0:CA D0 FD 60 18 6D 52 03 BE
28F8:8D 52 03 AD 53 03 69 00 59
2900:8D 53 03 AD 63 03 D0 21 13
2908:AD 52 03 38 E9 F4 8D 04 EB
2910:28 AD 53 03 E9 01 0D 04 ED
2918:28 90 0E AD 54 03 18 69 87
2920:32 8D 54 03 A9 01 8D 63 79
2928:03 4C 6E 29 8D 03 28 AD E5
2930:52 03 38 ED 03 28 8D 52 78
2938:03 AD 53 03 E9 00 8D 53 CF
2940:03 B0 08 A9 00 8D 52 03 B9
2948:8D 53 03 4C 6E 29 18 6D 11
2950:58 03 C9 24 90 02 A9 23 0E
2958:8D 58 03 60 8D 04 28 AD 68
2960:58 03 38 ED 04 28 B0 02 A9
2968:A9 00 8D 58 03 60 8A 48 BD
2970:98 48 A9 92 20 D2 FF A9 75
2978:05 20 D2 FF A2 06 A0 00 1E
2980:18 20 F0 FF AE 52 03 AD 77
2988:53 03 20 13 34 A2 0A A0 5B
2990:00 18 20 F0 FF AE 54 03 62
2998:A9 00 20 13 34 A0 00 AD C6
29A0:54 03 C9 19 D0 02 A0 F1 6A
29A8:AE 56 03 D0 05 8C 19 FF E1
29B0:D0 03 8C 20 D0 A9 12 20 31
29B8:D2 FF A2 12 A0 02 18 20 47
29C0:F0 FF AD 46 03 0A 0A 18 12
29C8:69 49 A8 20 D2 FF C8 98 F9
29D0:8D 04 28 20 D2 FF A9 D9 B5
29D8:A0 37 20 08 34 AC 04 28 52
29E0:C8 98 20 D2 FF C8 C0 60 F3
29E8:D0 04 38 E9 3E A8 98 20 30
29F0:D2 FF A9 92 20 D2 FF 68 BF
29F8:A8 68 AA 60 AD 5C 03 38 32
2A00:ED 52 03 8D 03 28 AD 5D 8A
2A08:03 ED 53 03 0D 03 28 B0 69
2A10:0B A0 01 B9 52 03 99 5C FB
2A18:03 88 10 F7 A2 0E A0 00 20
2A20:18 20 F0 FF AE 5C 03 AD 41
2A28:5D 03 4C 13 34 AD 5A 03 B6
2A30:18 69 C4 85 04 AD 5B 03 6C
2A38:69 03 85 05 AE 58 03 E8 C8
2A40:A0 00 A9 BC CA F0 02 10 14
2A48:02 A9 20 91 04 20 30 2D 53
2A50:AD 56 03 D0 02 A9 71 91 A9
2A58:02 C8 C0 23 D0 E4 60 AD B2
2A60:4A 03 D0 3A CE 4E 03 F0 FE
2A68:01 60 20 EC 2A 20 FB 2F 21
2A70:29 07 18 69 07 8D 4C 03 BE
2A78:A9 03 8D 4B 03 EE 4A 03 34

Chapter 2

2A80:A2 04 20 FB 2F 29 03 C9 D8
2A88:03 F0 F7 A8 B9 E9 2A A8 96
2A90:8C 4F 03 18 20 F0 FF A9 4F
2A98:38 A0 37 20 08 34 AD 4C D2
2AA0:03 F0 29 CE 4C 03 D0 C1 96
2AA8:A2 04 AC 4F 03 18 20 F0 83
2AB0:FF A9 00 8D 59 03 A2 5A BE
2AB8:A0 37 20 FB 2F 10 09 A9 64
2AC0:01 8D 59 03 A2 76 A0 37 BB
2AC8:8A 20 08 34 CE 4B 03 D0 29
2AD0:98 A9 00 8D 4A 03 A2 04 5C
2AD8:AC 4F 03 20 A7 35 AD 59 80
2AE0:03 F0 05 A9 0A 20 5C 29 E0
2AE8:60 0D 17 21 20 FB 2F 29 1E
2AF0:1F 09 20 8D 4E 03 60 AD E0
2AF8:5E 03 C9 0B F0 17 C9 0A A8
2B00:D0 12 A9 00 8D 5F 03 A9 12
2B08:01 8D 60 03 EE 5E 03 A9 1F
2B10:65 8D 62 03 60 AD 5A 03 6A
2B18:18 69 D8 85 04 AD 5B 03 D8
2B20:69 02 85 05 AC 5F 03 A9 3F
2B28:20 91 04 C8 91 04 AD 5F 57
2B30:03 18 6D 60 03 8D 5F 03 D1
2B38:F0 29 C9 0E F0 25 A2 00 CC
2B40:A8 AD 60 03 10 02 A2 02 62
2B48:BD 69 2B 91 04 C8 BD 6A 7F
2B50:2B 91 04 20 30 2D AD 56 0B
2B58:03 D0 02 A9 71 91 02 88 9D
2B60:91 02 60 A9 00 8D 5E 03 9C
2B68:60 1B 1C 1D 1E AC 47 03 40
2B70:B1 04 C9 85 90 31 C9 89 98
2B78:B0 2D 38 E9 85 AA BD FB 66
2B80:2C 8D 03 28 A5 04 38 ED CE
2B88:03 28 85 04 A5 05 E9 00 70
2B90:85 05 A2 00 20 B0 2C A9 05
2B98:0A 20 F4 28 AD 5E 03 C9 D3
2BA0:0A B0 03 EE 5E 03 60 C9 01
2BA8:84 F0 08 C9 89 90 05 C9 7D
2BB0:8D B0 01 60 C9 A0 F0 4C 1F
2BB8:C9 A2 B0 48 C9 8D 90 44 21
2BC0:AD 47 03 38 E9 13 AA BD 52
2BC8:D2 2C 18 6D 50 03 C9 05 94
2BD0:90 03 38 E9 05 AA BD 3C 60
2BD8:03 29 7F 4A 4A CD 46 03 A8
2BE0:D0 0A CE 46 03 D0 10 A9 85
2BE8:23 20 4E 29 A9 05 8D 46 F7
2BF0:03 20 70 38 4C 75 2C A9 9C
2BF8:0A 20 F4 28 A9 1C 9D 3C B2
2C00:03 4C 75 2C C9 66 D0 08 EF
2C08:A9 0A 20 2C 29 4C 75 2C 10
2C10:C9 A2 90 1A C9 B4 B0 16 42

2C18:C9 AB B0 08 A9 64 20 2C 22
2C20:29 4C 75 2C A9 00 8D 59 53
2C28:03 A9 19 20 F4 28 C9 1B 88
2C30:90 26 C9 1F B0 22 AD 60 4F
2C38:03 49 FF 18 69 01 8D 60 B0
2C40:03 AD 62 03 38 E9 14 30 C3
2C48:03 8D 62 03 A9 65 38 ED 43
2C50:62 03 20 F4 28 4C 75 2C 77
2C58:C9 B6 D0 19 A9 00 8D 65 BC
2C60:03 A9 01 8D 66 03 EE 64 1F
2C68:03 AD 64 03 C9 03 90 05 EA
2C70:A9 00 8D 64 03 AE 16 28 BC
2C78:EE 16 28 A5 04 9D C3 3B 86
2C80:A5 05 9D D7 3B AD 47 03 40
2C88:9D EB 3B B1 04 C9 1B 90 3B
2C90:06 C9 1F B0 02 A9 20 9D E1
2C98:FF 3B 20 30 2D B1 02 9D 98
2CA0:13 3C AD 56 03 D0 02 A9 B5
2CA8:71 91 02 A9 84 91 04 60 CB
2CB0:AC 47 03 BD DB 2C 91 04 24
2CB8:C8 E8 BD DB 2C 91 04 E8 BD
2CC0:98 18 69 27 A8 BD DB 2C 2B
2CC8:91 04 E8 C8 BD DB 2C 91 DB
2CD0:04 60 00 00 00 01 01 01 4A
2CD8:02 02 02 89 8A 8B 8C 8D B4
2CE0:8E 8F 90 91 92 93 94 95 31
2CE8:96 97 98 99 9A 9B 9C 9D 39
2CF0:9E 9F A1 66 66 66 66 20 D4
2CF8:20 20 20 00 01 28 29 AE 17
2D00:16 28 F0 23 CA BD C3 3B CF
2D08:85 04 BD D7 3B 85 05 BC 12
2D10:EB 3B BD FF 3B 91 04 20 2F
2D18:30 2D BD 13 3C 91 02 CA B5
2D20:10 E3 A9 00 8D 16 28 AD 73
2D28:11 FF 29 1F 8D 11 FF 60 33
2D30:A5 04 18 6D 14 28 85 02 86
2D38:A5 05 6D 15 28 85 03 60 63
2D40:20 4C 2E 4A B0 0D AE 11 50
2D48:28 E0 10 F0 06 EE 11 28 36
2D50:20 18 2E 4A B0 0D AE 11 53
2D58:28 E0 02 F0 06 CE 11 28 04
2D60:20 18 2E 4A B0 14 48 AD 4F
2D68:47 03 F0 0D A2 00 20 DB 47
2D70:2D CE 47 03 A2 04 20 DB 6F
2D78:2D 68 4A B0 16 48 AD 47 4C
2D80:03 C9 1F F0 0D A2 00 20 D4
2D88:DB 2D EE 47 03 A2 04 20 39
2D90:DB 2D 68 4A B0 3F AD 17 CA
2D98:28 D0 34 AD 58 03 F0 2F 7C
2DA0:A9 01 20 5C 29 20 33 2E 38
2DAB:AD 5A 03 18 69 57 85 04 0A

Chapter 2

2DB0:AD 5B 03 69 00 85 05 AE 7E
2DB8:11 28 F0 10 A5 04 18 69 9B
2DC0:28 85 04 A5 05 69 00 85 BE
2DC8:05 CA D0 F0 20 6D 2B A9 38
2DD0:01 8D 17 28 60 A9 00 8D AB
2DD8:17 28 60 AD 5A 03 18 69 28
2DE0:FF 85 04 AD 5B 03 69 02 B3
2DE8:85 05 AC 47 03 BD 10 2E AE
2DF0:91 04 E8 98 18 69 28 A8 1B
2DF8:BD 10 2E 91 04 E8 98 18 22
2E00:69 28 A8 BD 10 2E 91 04 6C
2E08:E8 C8 BD 10 2E 91 04 60 E3
2E10:A0 A0 A0 A0 80 81 82 83 95
2E18:48 A9 FF A0 FF 99 FF 23 96
2E20:C8 C0 10 F0 09 CC 11 28 E7
2E28:D0 F3 A9 96 D0 EF 20 6E 7D
2E30:2E 68 60 AD 56 03 D0 09 0E
2E38:AD 11 FF 09 40 8D 11 FF 9A
2E40:60 A9 80 8D 04 D4 A9 81 68
2E48:8D 04 D4 60 AD 56 03 F0 CA
2E50:04 AD 00 DC 60 A9 FD 8D 1B
2E58:08 FF EA EA AD 08 FF 8D DF
2E60:03 28 29 80 08 AD 03 28 9A
2E68:28 D0 02 29 EF 60 AD 5A 96
2E70:03 18 69 77 85 02 AD 5B E3
2E78:03 69 00 85 03 A0 00 A2 46
2E80:00 A9 20 EC 11 28 D0 02 E6
2E88:A9 1F 91 02 A5 02 18 6D A6
2E90:14 28 85 FC A5 03 6D 15 AA
2E98:28 85 FD AD 56 03 D0 02 67
2EA0:A9 71 91 FC 20 E9 30 E8 22
2EA8:E0 11 D0 D5 60 AD 56 03 9A
2EB0:F0 1E A9 00 A0 17 99 00 D6
2EB8:D4 88 10 FA A9 0F 8D 18 10
2EC0:D4 A9 19 8D 13 D4 A9 15 42
2EC8:8D 05 D4 A9 28 8D 01 D4 B0
2ED0:A9 00 8D 08 28 8D 16 28 FF
2ED8:8D 5E 03 A9 7D 8D 0B 28 EE
2EE0:8D 0C 28 A9 05 8D 0A 28 41
2EE8:8D 09 28 8D 46 03 8D 66 EB
2EF0:03 A9 41 8D 0D 28 8D 0E 6C
2EF8:28 A9 0F 8D 0F 28 8D 10 D2
2F00:28 A9 FA 8D 18 28 8D 19 AA
2F08:28 A9 0A 8D 47 03 8D 11 71
2F10:28 A2 04 20 DB 2D A9 FF 94
2F18:8D 50 03 A9 00 8D 4A 03 1A
2F20:8D 52 03 8D 53 03 8D 63 38
2F28:03 A9 00 8D 64 03 A9 FF CD
2F30:8D 65 03 A9 65 8D 61 03 D0
2F38:8D 62 03 A9 C8 8D 54 03 19
2F40:A9 23 8D 58 03 20 EC 2A 10

Type It In

2F48:20 6E 29 60 AD 56 03 D0 1B
2F50:03 4C D7 2F 78 A9 33 85 87
2F58:01 20 76 2F A9 37 85 01 37
2F60:58 A9 18 8D 18 D0 A9 D8 61
2F68:8D 16 D0 A9 0F 8D 22 D0 8B
2F70:A9 02 8D 23 D0 60 A2 07 5C
2F78:A0 00 A9 00 85 02 85 04 9F
2F80:A9 D0 85 03 A9 20 85 05 A6
2F88:B1 02 8E 03 28 E0 04 B0 BF
2F90:07 AE 56 03 D0 02 49 FF 3A
2F98:AE 03 28 91 04 88 D0 E8 F9
2FA0:E6 03 E6 05 E0 04 D0 09 22
2FA8:AD 56 03 D0 04 A9 D0 85 CE
2FB0:03 CA 10 D4 A0 00 B9 9B A6
2FB8:38 99 00 24 B9 93 39 99 04
2FC0:F8 24 B9 B3 3A 99 D0 21 12
2FC8:88 D0 EB A0 1F B9 93 3A 68
2FD0:99 D8 20 88 10 F7 60 AD 8D
2FD8:12 FF 29 FB 8D 12 FF AD 87
2FE0:13 FF 29 03 09 20 8D 13 15
2FE8:FF A9 D8 8D 07 FF A9 62 93
2FF0:8D 17 FF A9 41 8D 16 FF E2
2FF8:4C 76 2F AD 07 28 0A 0A D2
3000:38 6D 07 28 8D 07 28 60 74
3008:20 FF 2C EE 08 28 AD 5A 83
3010:03 18 69 70 85 02 85 04 6F
3018:AD 5B 03 69 01 85 03 85 C6
3020:05 A9 03 8D 03 28 A2 02 A6
3028:A0 0E B1 02 C9 85 90 04 3C
3030:C9 89 90 02 A9 20 C8 91 FA
3038:02 88 88 10 ED C8 A9 20 D3
3040:91 02 20 E9 30 CA D0 E0 BB
3048:20 E9 30 CE 03 28 D0 D6 57
3050:AD 08 28 4A B0 33 20 FB C1
3058:2F 30 2E 20 FB 2F 29 03 16
3060:C9 03 F0 25 AA A5 04 18 E2
3068:7D 8A 30 85 04 A5 05 69 B2
3070:00 85 05 A0 00 A9 85 91 20
3078:04 C8 A9 86 91 04 A0 28 B0
3080:A9 87 91 04 C8 A9 88 91 99
3088:04 60 00 F0 78 AD 08 28 C4
3090:4A B0 55 A2 09 AD 5A 03 CD
3098:18 69 E6 85 04 AD 5B 03 25
30A0:69 02 85 05 A5 04 38 E9 CE
30A8:28 85 02 A5 05 E9 00 85 6E
30B0:03 A0 01 B1 02 E0 09 D0 6C
30B8:0B C9 85 D0 07 48 A9 05 80
30C0:20 2C 29 68 C9 84 D0 02 EB
30C8:A9 20 91 04 88 10 E4 A5 6C
30D0:04 38 E9 28 85 04 A5 05 8D
30D8:E9 00 85 05 CA D0 C5 A0 F4

Chapter 2

30E0:01 A9 20 91 02 88 10 F9 95
30E8:60 A5 02 18 69 28 85 02 9D
30F0:A5 03 69 00 85 03 60 AD B8
30F8:5A 03 18 69 FD 85 02 AD 98
3100:5B 03 69 01 85 03 A9 02 9B
3108:8D 06 28 AD 06 28 8D 03 81
3110:28 A2 02 AD 03 28 18 69 9C
3118:28 A8 8D 04 28 B1 02 AC 63
3120:03 28 91 02 AD 04 28 8D BB
3128:03 28 CA D0 E6 A8 A2 A0 3C
3130:20 FB 2F C9 32 B0 02 A2 1F
3138:B5 8A 91 02 CE 06 28 10 59
3140:CA CE 66 03 F0 01 60 AD B2
3148:64 03 C9 02 D0 03 4C ED 10
3150:31 A8 0A AA BD 7A 32 85 23
3158:04 BD 7B 32 85 05 B9 AF 22
3160:33 CD 65 03 B0 05 A9 FF 99
3168:8D 65 03 EE 65 03 AD 65 32
3170:03 0A A8 B1 04 38 E9 01 DC
3178:0A AE 56 03 D0 1D AA BD 94
3180:F6 31 8D 0E FF AD 12 FF 18
3188:29 FC 1D F7 31 8D 12 FF C5
3190:AD 11 FF 09 10 8D 11 FF 77
3198:4C B6 31 AA BD 39 32 4A 21
31A0:08 8D 0F D4 BD 38 32 28 F4
31A8:6A 8D 0E D4 A9 10 8D 12 6D
31B0:D4 A9 11 8D 12 D4 AD 5A 7C
31B8:03 18 69 9A 85 02 AD 5B 64
31C0:03 69 02 85 03 98 48 A9 4C
31C8:B6 A0 08 91 02 88 10 FB 17
31D0:8A 4A 4A 4A C9 08 90 02 8A
31D8:A9 07 A8 A9 B8 91 02 C8 5A
31E0:A9 B9 91 02 68 A8 C8 B1 02
31E8:04 8D 66 03 60 AD 11 FF 89
31F0:29 4F 8D 11 FF 60 54 02 AA
31F8:6C 02 83 02 98 02 AD 02 CC
3200:C0 02 D2 02 E3 02 F3 02 D0
3208:02 03 10 03 1E 03 2A 03 B4
3210:36 03 42 03 4C 03 56 03 E6
3218:60 03 69 03 71 03 79 03 58
3220:81 03 88 03 8F 03 95 03 FD
3228:9B 03 A1 03 A6 03 AB 03 1B
3230:B0 03 B4 03 B9 03 BD 03 CC
3238:87 21 86 23 A2 25 DF 27 3C
3240:3E 2A C1 2C 6B 2F 3C 32 0C
3248:39 35 63 38 BE 3B 4B 3F 3F
3250:0F 43 0C 47 45 4B BF 4F 29
3258:7D 54 83 59 D6 5E 79 64 1E
3260:73 6A C7 70 7C 77 97 7E 88
3268:1E 86 18 8E 8B 96 7E 9F BC
3270:FA A8 06 B3 AC BD F3 C8 85

Type It In

3278:E6 D4 80 32 02 33 AE 33 26
3280:19 01 1B 02 19 01 1B 02 3A
3288:19 02 16 01 17 02 16 01 BA
3290:17 02 16 02 14 01 16 02 B6
3298:14 01 16 02 14 02 0F 08 F8
32A0:16 01 14 02 13 01 16 02 BD
32A8:14 02 19 08 12 01 16 02 FD
32B0:19 01 1D 02 1B 02 19 08 C0
32B8:19 01 1B 02 19 01 1B 02 72
32C0:19 02 16 01 17 02 16 01 F2
32C8:17 02 16 02 14 01 16 02 EE
32D0:14 01 16 02 14 02 0F 08 31
32D8:14 01 16 02 14 01 19 02 43
32E0:16 02 14 01 16 02 14 01 45
32E8:19 02 16 02 14 01 16 02 10
32F0:14 01 19 02 14 02 12 02 B1
32F8:0D 01 0F 02 0D 01 0A 02 A8
3300:06 04 09 01 10 01 09 01 33
3308:0D 01 10 02 08 01 10 01 BC
3310:08 01 0B 01 10 05 09 01 D3
3318:10 01 09 01 0D 01 10 02 86
3320:08 01 10 01 08 01 0B 01 38
3328:10 03 04 02 05 04 04 02 3A
3330:04 02 05 04 04 04 09 01 3D
3338:09 01 0C 01 15 01 09 01 B4
3340:09 01 0C 01 15 01 15 01 D4
3348:15 01 18 01 21 01 15 01 C4
3350:15 01 18 01 21 01 21 02 E5
3358:21 02 21 02 21 01 21 02 65
3360:1C 01 1E 01 19 01 1C 01 EE
3368:1E 02 15 02 17 01 19 01 11
3370:15 01 17 01 19 01 15 01 8C
3378:17 01 19 01 15 01 17 02 BA
3380:15 04 15 02 15 02 15 02 12
3388:15 01 15 02 10 01 12 01 26
3390:0D 01 10 01 12 02 09 02 7C
3398:0B 01 0D 01 09 01 0B 01 D9
33A0:0D 01 09 01 0B 01 0D 01 76
33A8:09 01 0B 02 09 05 00 3F F0
33B0:54 00 AD 28 03 C9 ED F0 85
33B8:2B A9 00 8D 12 28 A9 08 84
33C0:8D 13 28 A9 00 8D 14 28 D8
33C8:A9 FC 8D 15 28 A9 03 8D C1
33D0:10 FF A9 C8 8D 0F FF A9 53
33D8:0F 8D 11 FF A9 00 A2 00 DE
33E0:A0 0C D0 1A A9 00 8D 12 D0
33E8:28 A9 D8 8D 13 28 A9 00 4E
33F0:8D 14 28 A9 D4 8D 15 28 F1
33F8:A9 01 A2 00 A0 04 8D 56 4F
3400:03 8E 5A 03 8C 5B 03 60 41
3408:AE 56 03 F0 03 4C 1E AB FD

Chapter 2

3410:4C 88 90 8E 03 28 8D 04 93
3418:28 A0 FE C8 C8 AD 03 28 54
3420:38 F9 53 34 AD 04 28 F9 98
3428:54 34 B0 EF 98 4A 8D 05 EA
3430:28 A9 04 38 ED 05 28 AA 99
3438:F0 08 A9 20 20 D2 FF CA 69
3440:D0 F8 AE 03 28 AD 04 28 7D
3448:AC 56 03 F0 03 4C CD BD AE
3450:4C 5F A4 0A 00 64 00 E8 66
3458:03 10 27 FF FF A9 93 20 19
3460:D2 FF AD 12 28 85 04 AD 16
3468:13 28 85 05 A9 0F AE 56 A2
3470:03 D0 02 A9 6F A2 04 A0 18
3478:00 91 04 88 D0 FB E6 05 97
3480:CA D0 F6 A9 00 AE 56 03 66
3488:F0 08 8D 20 D0 8D 21 D0 EE
3490:F0 06 8D 15 FF 8D 19 FF 5E
3498:A9 13 8D 03 28 AE 03 28 A6
34A0:A0 07 18 20 F0 FF A9 CB C6
34A8:A0 35 20 08 34 EE 03 28 BE
34B0:AD 03 28 C9 17 D0 E6 A2 BE
34B8:17 A0 03 18 20 F0 FF A9 25
34C0:EF A0 35 20 08 34 A2 03 4B
34C8:A0 07 18 20 F0 FF A9 10 33
34D0:8D 03 28 A9 16 A0 36 20 20
34D8:08 34 CE 03 28 D0 F4 A2 6D
34E0:00 A0 05 18 20 F0 FF A9 02
34E8:2A A0 36 20 08 34 A9 11 CC
34F0:8D 03 28 AE 03 28 A0 08 D2
34F8:18 20 F0 FF A9 1B A0 36 C4
3500:20 08 34 AD 03 28 38 E9 F0
3508:03 8D 03 28 C9 0A B0 E3 F5
3510:A2 0B A0 19 18 20 F0 FF 57
3518:A9 5C A0 36 20 08 34 A2 12
3520:0B A0 1A 18 20 F0 FF A9 6B
3528:78 A0 36 20 08 34 A2 0C 22
3530:A0 1D 18 20 F0 FF A9 F6 09
3538:A0 36 20 08 34 A9 0D 8D F4
3540:03 28 A2 04 AC 03 28 20 AC
3548:A7 35 AD 03 28 18 69 0A 38
3550:8D 03 28 C9 28 90 EB AD EC
3558:5A 03 85 04 AD 5B 03 85 08
3560:05 A0 09 A2 3A 8A 91 04 E3
3568:C8 E8 E0 56 D0 F7 A0 31 CB
3570:A2 56 8A 91 04 C8 E8 E0 22
3578:58 D0 F7 A0 4B A2 59 8A 6E
3580:91 04 C8 CA E0 57 D0 F7 78
3588:A2 04 A0 00 18 20 F0 FF 7C
3590:A9 92 A0 37 20 08 34 A2 28
3598:13 A0 06 18 20 F0 FF A9 65
35A0:B2 A0 35 20 08 34 60 18 1F

Type It In

35A8:20 F0 FF A9 1C A0 37 4C 18
35B0:08 34 96 12 A9 11 9D 9D 8A
35B8:A9 20 11 9D 9D 9D A9 20 D2
35C0:20 11 9D 9D 9D 9D A9 20 E3
35C8:20 20 00 96 12 20 20 20 26
35D0:20 20 20 20 20 20 20 20 3B
35D8:20 20 20 20 20 20 20 20 43
35E0:20 20 20 20 20 20 20 20 4B
35E8:20 20 20 20 20 20 00 12 05
35F0:B7 B7 B7 B7 B7 B7 B7 B7 5B
35F8:B7 B7 B7 B7 B7 B7 B7 B7 63
3600:B7 B7 B7 B7 B7 B7 B7 B7 6C
3608:B7 B7 B7 B7 B7 B7 B7 B7 74
3610:B7 B7 B7 B7 B7 00 92 A5 41
3618:11 9D 00 B8 B8 B8 B8 B8 D2
3620:B8 B8 B8 B8 B8 B8 B8 B8 8C
3628:B8 00 96 12 3D 3A 11 9D 77
3630:9D 3B 92 12 3D 3A 11 9D 40
3638:9D 3B 3E 92 B7 B7 B7 B7 F6
3640:B7 B7 B7 B7 B7 B7 B7 B7 AC
3648:B7 B7 B7 B7 B7 B7 B7 B7 B4
3650:B7 B7 B7 B7 B7 B7 B7 B7 BC
3658:B7 B7 B7 00 05 12 A9 11 5A
3660:9D 20 11 9D 20 11 9D 20 40
3668:11 9D 20 11 9D 20 11 9D 07
3670:20 20 11 9D 9D 20 20 00 9E
3678:1F 12 20 20 20 20 20 20 E0
3680:20 20 20 11 9D 9D 9D 9D 56
3688:9D 9D 9D 9D 9D 20 20 20 86
3690:20 20 20 20 20 20 11 9D 5C
3698:9D 9D 9D 9D 9D 9D 9D 9D 05
36A0:20 20 20 20 20 20 20 20 0D
36A8:20 11 9D 9D 9D 9D 9D 9D 33
36B0:9D 9D 9D 20 20 20 20 20 EA
36B8:20 20 20 20 11 9D 9D 9D 1B
36C0:9D 9D 9D 9D 9D 9D 20 20 B4
36C8:20 20 20 20 20 20 20 11 26
36D0:11 9D 9D 9D 9D 9D 9D 9D F6
36D8:9D 20 20 20 20 20 20 20 04
36E0:20 20 11 9D 9D 9D 9D 9D 9D
36E8:9D 9D 9D 9D 20 20 20 20 FA
36F0:20 20 20 20 20 00 05 12 98
36F8:20 35 20 11 9D 9D 9D 20 96
3700:20 20 11 9D 9D 9D 35 20 70
3708:20 11 11 9D 9D 9D 9D 9D 03
3710:9D 36 36 36 36 36 36 36 32
3718:36 36 37 00 9E 12 20 C2 56
3720:20 11 9D 9D 9D C3 DB C3 E7
3728:11 9D 9D 9D 20 C2 20 11 71
3730:9D 9D 9D 92 20 20 20 00 73
3738:9E 12 DF 92 20 12 A9 11 4D

Chapter 2

3740:9D 9D 9D 20 92 20 12 20 F3
3748:11 9D 9D 9D CD 92 20 12 3F
3750:CE 11 9D 9D 9D 92 DF 20 0F
3758:A9 00 05 12 23 24 25 11 62
3760:9D 9D 9D 26 27 28 11 9D B4
3768:9D 9D 29 2A 2B 11 9D 9D 4B
3770:9D 92 20 20 20 00 05 12 75
3778:2C 2D 2E 11 9D 9D 9D 2F EC
3780:30 31 11 9D 9D 9D 32 33 4A
3788:34 11 9D 9D 9D 92 20 20 7A
3790:20 00 1E 53 43 4F 52 45 49
3798:11 11 11 11 9D 9D 9D 9D 43
37A0:54 49 4D 45 11 11 11 11 89
37A8:9D 9D 9D 9D 48 49 47 48 19
37B0:11 11 11 11 9D 9D 9D 9D 5B
37B8:43 41 52 44 00 05 50 52 AE
37C0:45 53 53 20 52 45 54 55 B8
37C8:52 4E 20 54 4F 20 50 4C 25
37D0:41 59 20 41 47 41 49 4E 6E
37D8:00 11 9D 9D 00 EE 50 03 78
37E0:AD 50 03 C9 05 D0 05 A9 56
37E8:00 8D 50 03 AD 5A 03 18 E9
37F0:69 82 85 04 AD 5B 03 69 EF
37F8:01 85 05 20 30 2D A0 09 6C
3800:AD 56 03 D0 02 A9 71 91 75
3808:02 88 10 FB A0 31 91 02 4C
3810:88 C0 27 D0 F9 A9 00 8D EA
3818:05 28 A9 03 8D 04 28 AD F4
3820:50 03 8D 03 28 AE 03 28 85
3828:BD 3C 03 10 07 29 7F 9D 61
3830:3C 03 A9 18 AA AC 05 28 70
3838:20 B3 2C AD 05 28 18 69 68
3840:03 8D 05 28 EE 03 28 AD 3A
3848:03 28 C9 05 D0 05 A9 00 BB
3850:8D 03 28 CE 04 28 D0 CD 6A
3858:20 FB 2F 29 03 AA BD 3C CA
3860:03 C9 1C F0 05 09 80 9D 42
3868:3C 03 60 04 08 0C 10 14 A8
3870:98 48 8A 48 A0 04 A9 00 7D
3878:99 3C 03 88 10 FA A0 04 5F
3880:20 FB 2F C9 05 B0 F9 AA 0C
3888:BD 3C 03 D0 F3 B9 6B 38 E9
3890:9D 3C 03 88 10 EA 68 AA 6F
3898:68 A8 60 96 96 96 96 96 AF
38A0:96 96 96 96 96 96 96 11
38A8:96 96 96 95 55 95 95 95 F7
38B0:55 65 A9 6A 9A 6A AA 6A 3F
38B8:56 55 56 6C FF 7F 7E FE 35
38C0:FF 7F 66 0C 33 C0 4C 73 47
38C8:40 7F 7F C0 30 C0 D4 11 74
38D0:11 D1 D1 7F 7F 7F 7F 7F E8

Type It In

38D8:7F 7F 15 D1 D1 D5 D4 D0 09
38E0:D0 D0 40 0C 00 C0 0C 43 15
38E8:00 73 43 00 00 C0 C4 01 2C
38F0:11 D1 00 30 40 43 33 4C 23
38F8:0F 3C 15 D1 11 04 C0 D0 AA
3900:10 00 00 7F E7 C3 99 99 8D
3908:81 99 99 FE FB F1 FB EF 54
3910:DF BF 7F FE FD FB F7 DF F1
3918:8F DF 7F E7 C3 99 99 81 F1
3920:99 99 FE 7F 99 93 87 87 4F
3928:93 99 FF FE FB F1 FB EF 4A
3930:DF BF 7F FE FD FB F7 DF 12
3938:8F DF 7F FF 99 93 87 87 0C
3940:93 99 FE 7F 81 9D 9D 9D 16
3948:95 81 FB FE FB F1 FB EF E4
3950:DF BF 7F FE FD FB F7 DF 32
3958:8F DF 7F 7F 81 9D 9D 9D CD
3960:95 81 FB 7F E1 F3 F3 F3 30
3968:F3 B3 83 FE FB F1 FB EF B1
3970:DF BF 7F FE FD FB F7 DF 52
3978:8F DF 7F 7F E1 F3 F3 F3 4D
3980:F3 B3 83 FF 73 6D 6D 6D E3
3988:6D 33 FE FE FB F1 FB EF DD
3990:DF BF 7F FE FD FB F7 DF 72
3998:8F DF 7F FF FF FF FF FF BA
39A0:FF FF FF FF 73 6D 6D 6D AC
39A8:6D 33 FE FF 80 80 80 80 06
39B0:80 80 80 FF 80 80 80 80 1B
39B8:80 80 80 FF 00 63 F7 FF 19
39C0:FF FF FF FF 01 01 81 81 C3
39C8:81 81 81 80 84 83 81 81 4B
39D0:81 81 81 FF FF FF 00 22 B6
39D8:22 08 08 81 91 E1 41 41 4F
39E0:41 41 41 81 81 80 9F BF 91
39E8:FF FF FF 41 3E 80 7F 80 E2
39F0:FF FF FF 41 41 81 79 7D F7
39F8:FF FF FF FF 80 80 81 81 F5
3A00:81 81 81 FF 63 F7 88 80 52
3A08:80 80 80 FF 01 81 C1 C1 40
3A10:C1 C1 C1 91 8F 81 81 81 2E
3A18:81 83 81 FF FF 00 63 42 67
3A20:08 FF FF C5 F9 41 41 41 8D
3A28:51 F9 C1 80 9F A0 C0 C0 C5
3A30:C0 C0 FF FF 7F BE 1C 0C 70
3A38:04 00 FF 81 79 85 03 03 B1
3A40:03 03 FF FF FF FF E7 E7 AE
3A48:FF FF FF 63 B1 D8 EF F7 BB
3A50:FB FD FE 00 80 C0 E0 F0 DB
3A58:F8 FC FE 1C 8E C7 E0 F0 70
3A60:F8 FC FE E3 71 38 1F 0F 68
3A68:07 03 01 00 00 00 00 00 41

Chapter 2

3A70:00 00 80 01 00 00 00 00 05
3A78:00 00 00 00 00 18 3C 3C 02
3A80:3C 3C 7E C0 E0 70 38 1C 53
3A88:0E 07 03 FF FF C0 C0 C0 6B
3A90:C0 C0 C0 00 00 00 0F 1F EA
3A98:FF 08 00 00 00 00 90 F8 29
3AA0:FC 20 00 00 00 00 09 1F CC
3AA8:3F 04 00 00 00 00 F0 F8 98
3AB0:FF 10 00 00 00 08 02 02 4F
3AB8:0B 2F BF 00 00 00 02 AA 25
3AC0:FF FE AE 00 00 00 00 AA 75
3AC8:FF AA 00 00 00 00 00 AA 92
3AD0:FF AA 00 00 00 00 00 AA 9A
3AD8:FE AA 00 14 15 15 15 55 35
3AE0:54 50 00 55 40 45 51 54 F6
3AE8:40 15 00 55 51 44 40 44 78
3AF0:44 55 00 55 45 45 45 45 41
3AF8:40 55 00 55 40 44 44 44 18
3B00:40 55 00 55 40 44 44 44 21
3B08:40 55 00 55 45 41 40 44 3D
3B10:45 55 00 50 14 15 15 15 B7
3B18:14 50 00 15 50 51 54 55 C3
3B20:50 15 00 55 11 51 50 11 D8
3B28:11 55 00 55 10 11 11 11 C9
3B30:10 55 00 55 10 11 11 11 51
3B38:10 55 00 55 10 14 14 14 6E
3B40:14 55 00 55 10 51 51 51 25
3B48:50 55 00 55 11 11 11 11 91
3B50:10 55 00 55 10 14 14 14 86
3B58:14 55 00 51 15 55 55 51 3D
3B60:51 40 00 40 50 50 50 56 4E
3B68:56 16 00 00 00 00 00 AA 3A
3B70:FF AA 00 00 00 00 00 AA 3C
3B78:FF AA 00 00 00 00 00 AA 44
3B80:FF AA 00 00 00 00 80 AA 4D
3B88:FF BF BA 00 00 08 20 80 27
3B90:E0 F8 FE BE BE BE BE BE AE
3B98:BE BE AA B8 20 00 00 00 FF
3BA0:00 00 00 BE BE BE BE BE 30
3BA8:BE BE AA 2E 08 00 00 00 A6
3BB0:00 00 00 B8 20 00 00 00 B3
3BB8:00 00 00 BE BE BE BE BE 48
3BC0:BE BE AA 00 00 00 00 00 9B

Program 3

Prisonball

Once you have a copy of "MLX" typed in and saved to disk or tape, load and run it. When you run MLX, you'll be asked for a starting address and an ending address for the data you'll be entering. For "Prisonball," those are

Starting Address: 0801
Ending Address: 0F90

Be sure you save a copy (preferably more than one) of the finished program to tape or disk before you quit MLX.

Prisonball

For mistake-proof program entry, use "MLX: Machine Language Entry Program" to type in this program.

```
0801:FF FF 0A 00 9E 32 30 36 A6
0809:31 00 00 00 A9 00 8D BA D4
0811:0F 20 E4 0D 20 BB 0A 20 42
0819:CA 0D 20 16 0E 20 39 0E A8
0821:20 3F 09 AD BA 0F D0 08 C8
0829:A9 01 8D BA 0F 4C 65 08 28
0831:CE FD 02 D0 0C AD FE 02 8C
0839:8D FD 02 20 E9 0C 20 BE 50
0841:08 CE FB 02 D0 0C AD FC B7
0849:02 8D FB 02 20 C3 0B 20 A3
0851:9B 09 20 E1 FF D0 D9 AD 38
0859:00 DC 2D 01 DC 29 10 D0 D2
0861:F6 4C 31 08 AD 00 DC 2D FA
0869:01 DC 29 1F C9 1F D0 F4 A9
0871:A9 B0 8D 14 04 8D 16 04 FB
0879:8D 17 04 A9 03 8D 15 D0 7A
0881:A2 18 A0 0C 18 20 F0 FF E0
0889:A9 7F A0 0F 20 1E AB 78 9C
0891:A9 00 85 C6 AD 00 DC 2D E7
0899:01 DC 29 10 F0 0B AD 00 96
08A1:DC C9 7F F0 EF A9 00 F0 A8
08A9:02 A9 01 85 F7 0A 0A 0A A3
08B1:18 69 08 8D FB 02 8D FC 02
08B9:02 58 4C 12 08 A5 A2 C9 71
08C1:3C D0 37 A9 00 85 A2 AD AE
08C9:17 04 C9 B0 F0 06 CE 17 FE
08D1:04 4C FB 08 A9 B9 8D 17 5D
08D9:04 AD 16 04 C9 B0 F0 06 53
```

Chapter 2

08E1:CE 16 04 4C FB 08 A9 B5 2D
08E9:8D 16 04 AD 14 04 C9 B0 96
08F1:D0 05 68 68 4C 65 08 CE 16
08F9:14 04 AD 14 04 C9 B0 D0 85
0901:12 AD 16 04 C9 B2 D0 0B 50
0909:AD 17 04 C9 B0 D0 04 EE 94
0911:20 D0 60 A9 00 8D 20 D0 55
0919:60 A6 06 BD 27 D0 29 0F 7F
0921:C9 03 F0 19 AA AC B8 0F 11
0929:B9 4E 0F 18 7D B4 0F 9D 89
0931:B4 0F BD B6 0F 69 00 9D 40
0939:B6 0F 20 3F 09 60 A0 07 74
0941:AE B5 0F AD B7 0F 20 56 65
0949:09 A0 22 AE B4 0F AD B6 2B
0951:0F 20 56 09 60 8C B9 0F 06
0959:86 FB 85 FC A2 08 A0 FF A4
0961:C8 A5 FB 48 38 FD 91 09 2B
0969:85 FB A5 FC 48 FD 92 09 2A
0971:85 FC 90 05 68 68 4C 61 C6
0979:09 68 85 FC 68 85 FB 98 94
0981:09 B0 AC B9 0F 99 00 04 58
0989:EE B9 0F CA CA 10 CF 60 A6
0991:01 00 0A 00 64 00 E8 03 5D
0999:10 27 A2 04 86 06 BD DF B9
09A1:02 38 E9 2A 29 F8 48 85 E5
09A9:FB A9 00 85 FC 06 FB 26 9A
09B1:FC 06 FB 26 FC 68 18 65 C4
09B9:FB 85 FB A5 FC 69 00 85 18
09C1:FC BD D7 02 85 FE BD CF 50
09C9:02 85 FD A0 03 46 FE 66 9D
09D1:FD 88 D0 F9 A5 FD 38 E9 3E
09D9:02 85 FD A5 FE E9 00 85 8D
09E1:FE A5 FB 18 65 FD 85 FB 08
09E9:A5 FC 65 FE 85 FC A5 FC 13
09F1:18 69 D8 85 FC A0 00 B1 F9
09F9:FB 29 0F A2 03 CA 30 68 6C
0A01:DD 40 0F D0 F8 8E B8 0F 85
0A09:A5 FC 38 E9 D4 85 FC B1 3D
0A11:FB A2 03 CA 30 52 DD 45 A4
0A19:0F D0 F8 8A 0A A8 A6 06 F6
0A21:A5 FD 38 E9 0B 4A 4A DD 21
0A29:96 0F F0 3C 9D 96 0F BD 51
0A31:BF 02 10 01 C8 B9 48 0F 84
0A39:A0 00 91 FB 20 74 0A C9 40
0A41:20 D0 1A A5 FD 38 E9 0B E6
0A49:4A 4A AA FE AA 0F BD AA 12
0A51:0F C9 2E D0 08 A9 00 9D B6
0A59:AA 0F 20 9C 0C 20 1A 09 72
0A61:A6 06 20 65 0C 4C 6B 0A 17
0A69:A6 06 CA E0 01 F0 03 4C D7
0A71:9D 09 60 48 8A 38 E9 02 32

Type It In

ØA79:A8 B9 B8 ØA A8 A9 1F 99 C9
ØA81:Ø1 D4 A9 11 99 Ø5 D4 A9 C5
ØA89:8Ø 99 Ø4 D4 A9 81 99 Ø4 9C
ØA91:D4 68 6Ø 48 8A 38 E9 Ø2 C5
ØA99:A8 B9 B8 ØA A8 68 ØA ØA 2D
ØAA1:ØA 18 69 ØA 99 Ø1 D4 A9 B2
ØAA9:13 99 Ø5 D4 A9 2Ø 99 Ø4 AØ
ØAB1:D4 A9 21 99 Ø4 D4 6Ø ØØ 8C
ØAB9:Ø7 ØE A9 ØØ AØ 17 99 ØØ 9E
ØAC1:D4 88 1Ø FA A9 ØF 8D 18 DØ
ØAC9:D4 A9 FF 8D ØF D4 A9 8Ø 2B
ØAD1:8D 12 D4 A9 14 8D FD Ø2 3B
ØAD9:8D FE Ø2 A9 ØØ 85 A2 AØ 4B
ØAE1:Ø4 99 AA ØF 99 B4 ØF 88 EA
ØAE9:1Ø F7 AØ Ø2 A2 4B AD 1B FØ
ØAF1:D4 1Ø Ø2 A2 B4 8A 99 C1 A3
ØAF9:Ø2 A9 Ø6 99 98 ØF 88 1Ø F5
ØBØ1:EB 6Ø BD D7 Ø2 DØ 17 BD 99
ØBØ9:CF Ø2 C9 ØA DØ 26 A9 14 E7
ØB11:9D 96 ØF A9 4D 9D CF Ø2 9A
ØB19:A9 Ø1 9D D7 Ø2 6Ø BD CF 52
ØB21:Ø2 C9 4E DØ ØF A9 14 9D 66
ØB29:96 ØF A9 ØB 9D CF Ø2 A9 ØE
ØB31:ØØ 9D D7 Ø2 6Ø BD D7 Ø2 75
ØB39:DØ 2E BD BF Ø2 1Ø 28 BD 55
ØB41:CF Ø2 C9 1B FØ Ø4 C9 45 1B
ØB49:DØ 1D AD DF Ø2 38 FD DF 8F
ØB51:Ø2 C9 F5 BØ Ø4 C9 ØF BØ BA
ØB59:ØE 2Ø 9C ØB A9 Ø1 9D 27 76
ØB61:DØ 2Ø 94 ØA 4C 93 ØB 6Ø 42
ØB69:BD BF Ø2 3Ø FA BD CF Ø2 Ø2
ØB71:C9 3D FØ Ø4 C9 13 DØ EF 46
ØB79:AD EØ Ø2 38 FD DF Ø2 C9 9F
ØB81:F5 BØ Ø4 C9 ØF BØ EØ 2Ø F8
ØB89:9C ØB A9 ØØ 9D 27 DØ 2Ø 31
ØB91:94 ØA A9 14 9D 96 ØF 2Ø 7Ø
ØB99:65 ØC 6Ø C9 ØØ DØ ØB AD 15
ØBA1:Ø4 DC 1Ø Ø4 A9 FF DØ Ø2 24
ØBA9:A9 Ø1 2Ø AB ØE Ø8 1Ø Ø3 47
ØBB1:2Ø AB ØE ØA ØA 18 69 ØA B2
ØBB9:28 1Ø Ø3 2Ø AB ØE 9D C7 E2
ØBC1:Ø2 6Ø A2 Ø4 2Ø Ø3 ØB 2Ø C8
ØBC9:17 ØC 2Ø 45 ØC C9 31 FØ A1
ØBD1:Ø4 C9 E5 DØ Ø3 2Ø 83 ØC D1
ØBD9:BD BF Ø2 1Ø ØE BD CF Ø2 Ø9
ØBE1:C9 14 DØ 26 BD D7 Ø2 DØ 8Ø
ØBE9:21 FØ ØC BD CF Ø2 C9 44 88
ØBF1:DØ 18 BD D7 Ø2 FØ 13 BD 63
ØBF9:DF Ø2 C9 57 9Ø Ø4 C9 BE 16
ØCØ1:9Ø Ø8 A9 14 9D 96 ØF 2Ø 5F
ØCØ9:65 ØC 2Ø 36 ØB CA EØ Ø1 84
ØC11:FØ Ø3 4C C5 ØB 6Ø AØ ØØ 63

Chapter 2

0C19:BD BF 02 10 02 A0 FF 8C 60
0C21:F8 02 8C F9 02 BD EF 02 50
0C29:18 7D BF 02 9D EF 02 BD 33
0C31:CF 02 6D F8 02 9D CF 02 17
0C39:BD D7 02 29 01 6D F9 02 AC
0C41:9D D7 02 60 A0 00 BD C7 AC
0C49:02 10 02 A0 FF 8C FA 02 DA
0C51:BD E7 02 18 7D C7 02 9D B0
0C59:E7 02 BD DF 02 6D FA 02 59
0C61:9D DF 02 60 BD BF 02 20 97
0C69:AB 0E 9D BF 02 BD CF 02 33
0C71:85 06 20 17 0C BD CF 02 3C
0C79:C5 06 F0 F6 20 17 0C 4C 45
0C81:17 0C BD C7 02 20 AB 0E 52
0C89:9D C7 02 BD DF 02 85 06 96
0C91:20 45 0C BD DF 02 C5 06 01
0C99:F0 F6 60 8A 48 A9 33 85 71
0CA1:FB A9 33 85 FD A9 04 85 05
0CA9:FC A9 D8 85 FE 8A 0A 0A 5E
0CB1:A8 BD 40 0F 85 F9 A2 16 F5
0CB9:A9 A0 91 FB C8 91 FB 88 CD
0CC1:A5 F9 91 FD C8 91 FD 88 4E
0CC9:A5 FB 18 69 28 85 FB A5 42
0CD1:FC 69 00 85 FC A5 FD 18 AD
0CD9:69 28 85 FD A5 FE 69 00 3D
0CE1:85 FE CA 10 D3 68 AA 60 CC
0CE9:A5 F7 F0 26 A2 01 BD 00 E7
0CF1:DC 4A B0 0D BD DF 02 C9 2D
0CF9:3B F0 13 DE DF 02 4C 0F EA
0D01:0D 4A B0 0A BD DF 02 C9 26
0D09:DC F0 03 FE DF 02 CA 10 CA
0D11:DD 60 78 AD 02 DC 48 A9 D9
0D19:C0 8D 02 DC A9 80 8D 00 6F
0D21:DC A0 80 EA 88 D0 FC A2 B4
0D29:01 BD 19 D4 C9 3B B0 04 44
0D31:A9 3B D0 06 C9 DC 90 02 4E
0D39:A9 DC 48 8A 0A 0A 0A 18 B5
0D41:69 BD 85 FB A9 0F 69 00 4C
0D49:85 FC BD BB 0F A8 68 91 56
0D51:FB C0 07 D0 07 A9 00 9D 04
0D59:BB 0F F0 03 FE BB 0F A9 12
0D61:00 85 FD 85 FE A0 07 A5 23
0D69:FD 18 71 FB 85 FD A5 FE E4
0D71:69 00 85 FE 88 10 F0 A5 EC
0D79:FD 46 FE 6A 46 FE 6A 46 F3
0D81:FE 6A 9D DF 02 CA 10 A1 64
0D89:68 8D 02 DC 58 60 A9 01 E1
0D91:8D 19 D0 A2 0E A0 07 A9 A7
0D99:00 85 02 B9 CF 02 9D 00 B2
0DA1:D0 B9 DF 02 9D 01 D0 B9 FA

Type It In

ØDA9:D7 Ø2 4A 26 Ø2 CA CA 88 35
ØDB1:1Ø E9 A5 Ø2 8D 1Ø DØ A9 1B
ØDB9:FA 8D 12 DØ AD ØD DC 29 88
ØDC1:Ø1 FØ Ø3 4C 31 EA 4C BC 48
ØDC9:FE A9 7F 8D ØD DC A9 8F 55
ØDD1:8D 14 Ø3 A9 ØD 8D 15 Ø3 7E
ØDD9:A9 1B 8D 11 DØ A9 81 8D 1Ø
ØDE1:1A DØ 6Ø AØ 7F B9 B1 ØE A7
ØDE9:99 4Ø Ø3 88 1Ø F7 A9 ØE 8B
ØDF1:8D F8 Ø7 8D F9 Ø7 A9 ØD 17
ØDF9:AØ Ø2 99 FA Ø7 88 1Ø FA 3D
ØEØ1:A9 1F 8D 15 DØ A9 Ø3 8D 7D
ØEØ9:1D DØ A2 Ø2 A9 32 9D C9 77
ØE11:Ø2 CA 1Ø F8 6Ø AØ Ø4 B9 B9
ØE19:31 ØF 99 CF Ø2 B9 36 ØF 34
ØE21:99 D7 Ø2 B9 3B ØF 99 DF Ø5
ØE29:Ø2 B9 34 ØE 99 27 DØ 88 AF
ØE31:1Ø E5 6Ø Ø1 ØØ Ø3 Ø3 Ø3 FF
ØE39:A9 93 2Ø D2 FF A9 ØB 8D 8A
ØE41:21 DØ A9 ØØ 8D 2Ø DØ AØ 86
ØE49:27 A9 AØ 99 ØØ Ø4 99 CØ 15
ØE51:Ø7 A9 ØF 99 ØØ D8 99 CØ 2E
ØE59:DB 88 1Ø ED AØ C8 A9 AØ 82
ØE61:99 ØØ Ø4 99 27 Ø4 99 F8 D9
ØE69:Ø6 99 1F Ø7 A9 ØF 99 ØØ FF
ØE71:D8 99 27 D8 99 F8 DA 99 D2
ØE79:1F DB 98 38 E9 28 A8 C9 BD
ØE81:D8 DØ DB A2 Ø4 2Ø 9C ØC C9
ØE89:CA 1Ø FA A2 ØØ AØ Ø1 18 35
ØE91:2Ø FØ FF A9 51 AØ ØF 2Ø DF
ØE99:1E AB A2 18 AØ ØF 18 2Ø 17
ØEA1:FØ FF A9 74 AØ ØF 2Ø 1E 52
ØEA9:AB 6Ø 49 FF 18 69 Ø1 6Ø A5
ØEB1:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ CD
ØEB9:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ D5
ØEC1:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ DD
ØEC9:ØØ ØØ ØØ ØØ 18 ØØ ØØ 3C E2
ØED1:ØØ ØØ 3C ØØ ØØ 18 ØØ ØØ D5
ØED9:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ F5
ØEE1:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ FD
ØEE9:ØØ ØØ ØØ ØØ ØØ ØØ ØØ 7F 85
ØEF1:CØ ØØ Ø3 CØ ØØ Ø3 CØ ØØ 68
ØEF9:Ø3 CØ ØØ Ø3 CØ ØØ Ø3 CØ C4
ØFØ1:ØØ Ø3 CØ ØØ Ø3 CØ ØØ Ø3 16
ØFØ9:CØ ØØ Ø3 CØ ØØ Ø3 CØ ØØ 81
ØF11:Ø3 CØ ØØ Ø3 CØ ØØ Ø3 CØ DD
ØF19:ØØ Ø3 CØ ØØ Ø3 CØ ØØ Ø3 2E
ØF21:CØ ØØ Ø3 CØ ØØ Ø3 CØ ØØ 99
ØF29:Ø3 CØ ØØ Ø3 CØ ØØ Ø3 Ø8 3D
ØF31:21 1F ØC ØC ØC ØØ Ø1 ØØ 4C
ØF39:ØØ ØØ 82 82 82 8C 96 ØA 4D

Chapter 2

0F41:03 0D 03 0A A0 E1 61 E1 56
0F49:61 20 20 20 20 0A 14 1E 95
0F51:9B 12 53 43 4F 52 45 20 CE
0F59:20 20 20 20 20 20 20 20 77
0F61:54 49 4D 45 20 33 3A 30 6C
0F69:30 20 20 20 20 53 43 4F D1
0F71:52 45 00 50 52 49 53 4F BC
0F79:4E 42 41 4C 4C 00 50 52 91
0F81:45 53 53 20 46 49 52 45 C4
0F89:42 55 54 54 4F 4E 00 00 A1

Program 4

Q-Bird

You must have a copy of "MLX" typed in and saved to disk or tape before you start entering "Q-Bird." Load and run MLX; then answer the prompts for the starting and ending addresses with

Starting Address: 0801

Ending Address: 1910

Save a copy (several is safer) of the completed program to tape or disk before you exit MLX.

Q-Bird

For mistake-proof program entry, use "MLX: Machine Language Entry Program" to type in this program.

```
0801:0B 08 00 00 9E 32 30 36 EC
0809:31 00 00 00 A9 00 8D 20 3A
0811:D0 20 AD 14 20 1B 08 4C 52
0819:C1 0B A9 93 20 D2 FF A9 31
0821:08 20 D2 FF A9 8E 20 D2 32
0829:FF A9 04 85 02 A9 0E 38 87
0831:E5 02 85 D3 A5 02 85 D6 B9
0839:20 10 E5 A0 06 A2 0E BD B8
0841:61 08 20 D2 FF CA 10 F7 78
0849:88 10 F2 A9 0D 20 D2 FF 29
0851:20 D2 FF E6 02 E6 02 A5 E9
0859:02 C9 10 D0 D0 4C 70 08 8C
0861:1D 91 B7 B7 98 9D 9D 9D EA
0869:11 A9 92 20 A9 12 05 A0 01
0871:09 A9 00 85 02 A6 02 BD 35
0879:AD 08 C9 FF F0 08 20 D2 56
0881:FF E6 02 4C 76 08 88 10 45
0889:03 4C 8F 13 E8 E6 02 BD F5
0891:AD 08 85 D3 E8 E6 02 BD 0D
0899:AD 08 85 D6 48 8A 48 98 36
08A1:48 20 10 E5 68 A8 68 AA 9F
08A9:68 4C 82 08 FF 21 05 1C 7C
08B1:51 2D 42 49 52 44 FF 21 57
08B9:06 A3 A3 A3 A3 A3 A3 FF 57
08C1:23 08 1F 42 59 FF 22 0A 86
08C9:4D 49 4B 45 FF 21 0C 53 80
08D1:45 44 4F 52 45 FF 04 11 E7
08D9:1E 48 49 53 43 4F 52 45 AA
08E1:3A 99 34 36 33 37 30 FF 36
```

Chapter 2

08E9:05 13 1E 53 43 4F 52 45 7B
08F1:3A 99 34 36 33 37 30 FF 46
08F9:19 11 1E 4C 45 56 45 4C BD
0901:3A 99 30 31 FF 19 13 1E 58
0909:4C 49 56 45 53 3A 99 FF 69
0911:A9 97 20 D2 FF A0 02 A9 3F
0919:A4 20 2E 09 A0 10 A9 A3 18
0921:20 2E 09 A0 14 A9 A4 20 AA
0929:2E 09 4C 42 09 84 D6 A0 EA
0931:00 84 D3 48 20 10 E5 68 D8
0939:A2 26 20 D2 FF CA 10 FA 9D
0941:60 A9 1F 85 D3 A9 03 85 FA
0949:D6 20 10 E5 A2 0C A9 A7 6F
0951:20 D2 FF A9 9D 20 D2 FF D5
0959:A9 11 20 D2 FF CA 10 EE EF
0961:60 A9 FF 8D 1C D0 A9 00 5E
0969:8D 15 D0 8D 1D D0 8D 17 D8
0971:D0 A9 07 8D 25 D0 4C 7A 8F
0979:0B 03 C0 00 0E B0 F0 3A 39
0981:AF AC 36 6A AC D6 AA B0 CA
0989:3A AA B0 0E AA C0 03 67 20
0991:00 03 77 00 0D DD C0 03 B7
0999:33 00 00 00 00 00 00 00 45
09A1:00 00 00 00 00 00 00 00 B3
09A9:00 00 00 00 00 00 00 00 BB
09B1:00 00 00 00 00 00 00 00 C3
09B9:00 03 C3 00 0E BE F0 3A 8C
09C1:AA EC D6 6B AC 36 AB B0 3C
09C9:DA AE B0 3E AA C0 03 67 B4
09D1:00 03 77 00 03 77 00 0D 96
09D9:DD C0 03 33 00 00 00 00 9E
09E1:00 00 00 00 00 00 00 00 F3
09E9:00 00 00 00 00 00 00 00 FB
09F1:00 00 00 00 00 00 00 00 04
09F9:00 00 0F 00 3C 3A C0 EB 26
0A01:EA B0 EA A6 70 3A AA 5C 9C
0A09:3A AA B0 0E AA C0 03 67 A1
0A11:00 03 77 00 0D DD C0 03 39
0A19:33 00 00 00 00 00 00 00 C6
0A21:00 00 00 00 00 00 00 00 35
0A29:00 00 00 00 00 00 00 00 3D
0A31:00 00 00 00 00 00 00 00 45
0A39:00 03 0F 00 3E FA C0 EE 3E
0A41:AA B0 EB A6 5C 3B AA 70 54
0A49:3A EA 9C 0E AA F0 03 67 30
0A51:00 03 77 00 03 77 00 0D 18
0A59:DD C0 03 33 00 00 00 00 20
0A61:00 00 00 00 00 00 00 00 75
0A69:00 00 00 00 00 00 00 00 7D
0A71:00 00 00 00 00 00 00 00 85
0A79:00 03 C0 00 0E B0 00 36 CF

Type It In

0A81:6C 00 EA AC 00 FE AC 00 49
0A89:EA AC 00 3E B0 00 EA AC 2A
0A91:00 35 AC 00 0D AC 00 0D B0
0A99:AC 00 0D AC 00 3A AC 00 B2
0AA1:EA B0 00 3E B0 00 0E B0 8D
0AA9:00 03 AC 00 00 F0 00 00 D7
0AB1:00 00 00 00 00 00 00 C5
0AB9:00 03 C0 00 0E B0 00 36 10
0AC1:6C 00 EA AC 00 EA AC 00 39
0AC9:3A B0 00 0E AC 00 3A AC 8E
0AD1:00 E5 AC 00 3D AC 00 0D 9E
0AD9:AC 00 0D AC 00 3A B0 00 FA
0AE1:EA B0 00 EE B0 00 3E B0 39
0AE9:00 03 B0 00 00 C0 00 00 D7
0AF1:00 00 00 00 00 00 00 06
0AF9:00 0F 00 00 3A C0 00 E6 8D
0B01:70 00 EA AC 00 EA FC 00 1D
0B09:EA AC 00 3A F0 00 EA AC 6D
0B11:00 E9 70 00 E9 C0 00 E9 EB
0B19:C0 00 E9 C0 00 EA B0 00 E5
0B21:3A AC 00 3A F0 00 3A C0 DF
0B29:00 EB 00 00 3C 00 00 00 1C
0B31:00 00 00 00 00 00 00 47
0B39:00 0F 00 00 3A C0 00 E6 CE
0B41:70 00 EA AC 00 EA AC 00 BC
0B49:3A B0 00 EA C0 00 EA B0 E3
0B51:00 E9 6C 00 E9 F0 00 E9 6C
0B59:C0 00 E9 C0 00 3A B0 00 63
0B61:3A AC 00 3A EC 00 3A F0 30
0B69:00 3B 00 00 0C 00 00 00 AE
0B71:00 00 00 00 00 00 00 87
0B79:00 A9 00 8D 26 D0 A2 07 93
0B81:BD 93 0B 9D 27 D0 BD 9B 2A
0B89:0B 9D F8 07 CA 10 F1 4C E2
0B91:A3 0B 05 04 04 04 09 02 61
0B99:06 0E 8C 86 84 86 8A 88 0C
0BA1:80 8E A2 0F BD B1 0B 9D 49
0BA9:00 D0 CA 10 F7 60 00 00 8F
0BB1:78 66 6D 47 FD 47 1D 97 9E
0BB9:00 00 00 00 A9 9E 00 00 97
0BC1:20 62 09 4C 80 12 4C E9 35
0BC9:18 AD 34 03 C9 07 D0 0F 29
0BD1:A9 00 8D 34 03 EE 35 03 F2
0BD9:AD 35 03 C9 05 F0 03 4C 4F
0BE1:D8 15 20 2A 0C F0 03 4C C6
0BE9:A9 11 20 C0 11 29 0F 49 BD
0BF1:0F 85 03 29 01 F0 08 A9 69
0BF9:00 8D 00 80 4C 8D 0C A5 D1
0C01:03 29 02 F0 08 A9 20 8D E8
0C09:00 80 4C 8D 0C A5 03 29 C9

Chapter 2

0C11:04 F0 08 A9 40 8D 00 80 BB
0C19:4C 8D 0C A5 03 29 08 F0 54
0C21:BE A9 60 8D 00 80 4C 8D 10
0C29:0C A9 00 8D 35 03 AD 36 D1
0C31:03 60 8D 21 D0 A9 00 A2 76
0C39:CB 9D 34 03 CA E0 FF D0 FF
0C41:F8 60 EE 36 03 AE 36 03 71
0C49:E0 1A D0 08 A9 00 8D 36 91
0C51:03 4C 15 16 8A 18 6D 00 91
0C59:80 AA BD F3 0C AA BD 76 50
0C61:0C 18 6D 0C D0 8D 0C D0 99
0C69:BD 81 0C 18 6D 0D D0 8D 92
0C71:0D D0 4C CF 17 00 00 00 83
0C79:00 FF FF FF 00 01 01 01 98
0C81:00 FF 01 00 00 FF 01 00 BB
0C89:00 FF 01 00 AD 00 80 F0 21
0C91:0C C9 60 F0 08 A9 81 8D B4
0C99:FE 07 4C 43 0C A9 83 8D 4C
0CA1:FE 07 4C 43 0C CE FE 07 59
0CA9:4C E0 0B A9 04 85 FE A0 F0
0CB1:00 84 FD A2 04 AD 3A 03 23
0CB9:18 65 FD 85 FD A5 FE 69 3D
0CC1:00 85 FE CA 10 EF A5 FD 51
0CC9:18 6D 3B 03 85 FD A5 FE 4F
0CD1:69 00 85 FE 60 00 00 AD EF
0CD9:0D D0 38 E9 31 8D 3A 03 89
0CE1:AD 0C D0 4A 4A 4A 18 69 A7
0CE9:0C 8D 3B 03 20 AC 0C 4C 1B
0CF1:8C 0D 00 01 01 01 01 01 B2
0CF9:01 01 01 09 09 09 09 0B
0D01:09 09 09 08 08 08 08 08 FB
0D09:08 08 08 00 00 00 00 2A
0D11:00 00 00 04 04 04 04 04 A7
0D19:04 04 04 06 06 06 06 06 71
0D21:06 06 06 02 02 02 02 02 BE
0D29:02 02 02 00 00 00 00 05
0D31:00 00 00 05 05 05 05 05 E6
0D39:05 05 05 04 04 04 04 04 34
0D41:04 04 04 06 06 06 06 06 99
0D49:06 06 06 00 00 00 00 00 A8
0D51:00 00 00 09 09 09 09 09 83
0D59:09 09 09 08 08 08 08 08 54
0D61:08 08 08 0A 0A 0A 0A 0A B9
0D69:0A 0A 0A 00 00 00 00 00 4C
0D71:00 00 A5 FE 18 69 D4 85 C5
0D79:FE A0 02 B1 FD 4C 48 12 5A
0D81:20 01 17 91 FD 88 10 F3 0E
0D89:4C A6 0C B1 FD C9 E9 F0 EB
0D91:E1 4C 46 14 EE 37 03 AD C1
0D99:37 03 C9 02 D0 2C A9 00 F3
0DA1:8D 37 03 EE 38 03 AD 38 01

Type It In

ØDA9:Ø3 C9 Ø1 DØ 1D A9 ØØ 8D Ø2
ØDB1:38 Ø3 A2 Ø1 FE 3C Ø3 BD B9
ØDB9:3C Ø3 DD CD ØD DØ Ø8 A9 BØ
ØDC1:ØØ 9D 3C Ø3 2Ø EA ØD 4C ØE
ØDC9:BC ØE 6Ø 15 11 18 1C 2Ø 64
ØDD1:A2 18 A9 ØØ 9D ØØ D4 CA D9
ØDD9:1Ø FA A9 FF 8D ØF D4 A9 EB
ØDE1:8Ø 8D 12 D4 A9 ØØ 4C 7Ø 85
ØDE9:17 BD 4Ø Ø3 DØ ØB 2Ø 5E 88
ØDF1:ØE ØA ØA ØA ØA ØA 2Ø 47 77
ØDF9:ØE BD 5Ø Ø3 8D Ø1 8Ø FE 35
ØEØ1:4Ø Ø3 BC 4Ø Ø3 CØ 1E DØ C1
ØEØ9:Ø6 A9 ØØ 9D 4Ø Ø3 6Ø 86 C1
ØE11:Ø5 Ø6 Ø5 8E 4F Ø3 98 18 8A
ØE19:6D Ø1 8Ø A8 B9 F3 ØC DØ 4D
ØE21:ØA 48 BD F8 Ø7 29 FE 9D 14
ØE29:F8 Ø7 68 AA A4 Ø5 BD 76 66
ØE31:ØC 18 79 ØØ DØ 99 ØØ DØ 46
ØE39:BD 81 ØC 18 79 Ø1 DØ 99 A2
ØE41:Ø1 DØ AE 4F Ø3 6Ø 9D 5Ø Ø2
ØE49:Ø3 C9 ØØ FØ ØA C9 6Ø FØ 91
ØE51:Ø6 A9 85 9D F8 Ø7 6Ø A9 B3
ØE59:87 9D F8 Ø7 6Ø 2Ø B3 ØE 29
ØE61:AD 3A Ø3 38 E9 2C 8D 3A 1C
ØE69:Ø3 AD 3B Ø3 4A 4A 4A 18 32
ØE71:69 ØC 8D 3B Ø3 86 FC 2Ø F6
ØE79:AC ØC A6 FC AØ ØØ B1 FD F9
ØE81:C9 E9 DØ D9 AD FF Ø3 6Ø 88
ØE89:86 Ø5 2Ø 97 17 29 Ø3 8D 98
ØE91:FF Ø3 AA BD AB ØE 18 79 DE
ØE99:Ø1 DØ 8D 3A Ø3 BD AF ØE 3C
ØEA1:18 79 ØØ DØ 8D 3B Ø3 A6 3B
ØEA9:Ø5 6Ø F2 12 Ø2 Ø2 12 F2 ØF
ØEB1:EA 1A 86 Ø5 Ø6 Ø5 A4 Ø5 7D
ØEB9:4C 89 ØE E8 EØ Ø4 FØ Ø3 AA
ØEC1:4C B5 ØD 4C CB ØD A2 ØØ AF
ØEC9:BD 7A Ø9 9D ØØ 2Ø BD 7A D4
ØED1:ØA 9D ØØ 21 BD E7 ØE 9D B3
ØED9:ØØ 22 BD A7 ØF 9D CØ 22 43
ØEE1:CA DØ E5 4C D1 ØD ØØ FC D8
ØEE9:ØØ Ø3 AB ØØ ØE 5A CØ 39 DØ
ØEF1:AA BØ 3A AA BØ EA A9 AC B2
ØEF9:EA AA AC EA AA AC 39 AA 9F
ØFØ1:BØ 39 69 BØ ØE 56 CØ Ø3 4C
ØFØ9:AB ØØ ØØ FC ØØ ØØ ØØ ØØ CC
ØF11:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 2F
ØF19:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ 37
ØF21:ØØ ØØ ØØ ØØ ØØ ØØ ØØ FC 3C
ØF29:ØØ Ø3 AB ØØ ØE 96 CØ 3A Ø4
ØF31:A9 BØ 3A AA BØ EA AA AC 75
ØF39:EA AA AC EA AA AC 39 AA EØ

0F41:B0 39 6A B0 0E 5A C0 03 BC
 0F49:AB 00 00 FC 00 00 0D
 0F51:00 00 00 00 00 00 6F
 0F59:00 00 00 00 00 00 77
 0F61:00 00 00 00 00 00 7F
 0F69:00 00 00 00 00 00 8A
 0F71:F3 FC 3E AE AB EF FA 65 F3
 0F79:EA AA A9 EA EA AF 3F AA DA
 0F81:F0 3A AF 00 0F F0 00 00 D8
 0F89:00 00 00 00 00 00 A7
 0F91:00 00 00 00 00 00 AF
 0F99:00 00 00 00 00 00 B7
 0FA1:00 00 00 00 00 00 BF
 0FA9:00 00 00 00 00 00 7D
 0FBI:BC 00 0E AB FC 3F AE AB 98
 0FB9:EF FA 65 EA AA A9 EA EA A6
 0FCl:AF 3F AA F0 0E AB 00 0E 19
 0FC9:AC 00 3A F0 00 0F 00 00 D0
 0FD1:00 00 00 00 00 00 EF
 0FD9:00 00 00 00 00 00 F7
 0FE1:00 00 00 00 00 00 FF
 0FE9:00 00 FC 00 03 AB 00 0E 7C
 0FEF:66 C0 3E AA F0 35 A9 70 06
 0FF9:35 99 70 0D 55 C0 03 57 03
 1001:00 03 9B 00 03 AB 00 0E 2A
 1009:6C 00 0E AF C0 39 BE B0 35
 1011:3A BA C0 39 BE B0 0E AA 28
 1019:B0 0E 6A C0 03 AB 00 00 35
 1021:FC 00 00 00 00 03 FF C5
 1029:00 0E 66 C0 3A AA B0 35 B8
 1031:99 70 35 55 70 0D 55 C0 59
 1039:03 57 00 03 9B 00 03 9B 5F
 1041:00 03 AB 00 03 9B 00 0E 2C
 1049:AC 00 0E 6C 00 0E AC 00 D9
 1051:0E 6C 00 0E AC 00 03 6B 4B
 1059:00 03 AB 00 03 9B 00 00 36
 1061:EA C0 00 3F 00 00 00 1B
 1069:00 00 00 00 00 00 4A
 1071:EC 00 03 AB 00 03 6B 00 06
 1079:03 6B 00 03 5B 00 03 9B A2
 1081:00 00 FC 00 00 00 00 41
 1089:00 00 00 00 00 00 00 A9
 1091:00 00 00 00 00 00 00 B1
 1099:00 00 00 00 00 00 00 B9
 10A9:D0 18 69 08 38 ED F6 03 50
 10A9:D0 18 69 08 38 ED F6 03 50
 10B1:F0 08 90 03 A9 00 60 A9 46
 10B9:20 60 AD 00 D0 18 69 02 73
 10C1:38 ED F5 03 90 03 A9 40 8C
 10C9:60 A9 60 AD F0 03 D0 9E
 10D1:06 20 36 11 8D 02 80 EE 39

10D9:F0 03 AE F0 03 E0 20 F0 E4
 10E1:2B 8A 18 6D 02 80 AA BD 39
 10E9:F3 0C AA D0 08 AD F8 07 59
 10F1:29 8C 8D F8 07 BD 76 0C 33
 10F9:18 6D 00 D0 8D 00 D0 BD 5A
 1101:81 0C 18 6D 01 D0 8D 01 28
 1109:D0 4C 4E 11 A9 00 8D F0 DA
 1111:03 4C 0A 11 EE F1 03 AD 0D
 1119:F1 03 C9 0A D0 EB A9 00 58
 1121:8D F1 03 EE F2 03 AD F2 C7
 1129:03 C9 05 D0 DC A9 00 8D 08
 1131:F2 03 4C CD 10 A9 8D 8D C3
 1139:F8 07 4C A7 10 AD 0C D0 BD
 1141:8D F5 03 AD 0D D0 8D F6 A0
 1149:03 AD 00 DC 60 EE FA 03 DD
 1151:AD FA 03 C9 06 D0 0F A9 41
 1159:00 8D FA 03 EE FB 03 AD 89
 1161:FB 03 C9 05 F0 28 4C D2 5F
 1169:11 A9 00 8D FB 03 EE 08 29
 1171:D0 AD 08 D0 C9 01 D0 EE 58
 1179:20 97 17 29 07 C9 06 B0 23
 1181:F7 0A 0A 0A 0A 18 69 4C D3
 1189:8D 09 D0 4C 67 11 EE FC ED
 1191:03 AD FC 03 C9 20 D0 D1 B2
 1199:AD FC 07 49 01 8D FC 07 86
 11A1:A9 00 8D FC 03 4C 6A 11 49
 11A9:AD FE 03 C9 01 F0 03 4C 7D
 11B1:43 0C EE FD 03 AD FD 03 04
 11B9:29 01 F0 F3 4C 46 0C A9 4B
 11C1:00 8D FE 03 AD 00 DC 29 A7
 11C9:10 D0 03 EE FE 03 4C 3E 52
 11D1:11 EE F3 03 AD F3 03 C9 F3
 11D9:05 D0 0F A9 00 8D F3 03 50
 11E1:EE F4 03 AD F4 03 C9 0A 45
 11E9:F0 03 4C C7 0B A9 00 8D D7
 11F1:F4 03 EE 0B D0 4C 39 12 1A
 11F9:C9 01 F0 21 C9 48 90 1A 1C
 1201:CE 0A D0 EE EF 03 AD EF EE
 1209:03 C9 08 D0 0D AD FD 07 51
 1211:49 01 8D FD 07 A9 00 8D 18
 1219:EF 03 4C EB 11 20 97 17 8D
 1221:29 07 C9 07 B0 F7 AA A9 A9
 1229:00 18 69 18 CA 10 FA 18 A6
 1231:69 58 8D 0A D0 4C 01 12 3E
 1239:AD 0B D0 C9 A8 D0 B9 A9 53
 1241:01 8D 0B D0 4C 1E 12 29 DF
 1249:0F C9 00 D0 03 4C 89 0D DD
 1251:85 04 E6 04 C9 0F D0 0D 8F
 1259:EE D0 03 AD D0 03 C9 7E 09
 1261:D0 03 4C 69 12 4C 81 0D A0
 1269:A5 04 91 FD 20 01 17 4C F2

Chapter 2

1271:55 13 AD E1 03 8D 56 12 E5
1279:8D 4B 12 4C 82 13 60 A9 09
1281:06 85 D3 A9 16 85 D6 20 B3
1289:10 E5 A2 00 BD 9B 12 20 24
1291:D2 FF E8 E0 3A D0 F5 4C 97
1299:3B 15 08 8E 9E 55 53 45 C0
12A1:20 53 54 49 43 4B 20 54 A5
12A9:4F 20 53 45 4C 45 43 54 8E
12B1:20 4C 45 56 45 4C 0D 20 9C
12B9:20 20 20 20 20 20 50 52 70
12C1:45 53 53 20 54 52 49 47 8F
12C9:47 45 52 20 54 4F 20 53 A2
12D1:54 41 52 54 20 48 15 29 75
12D9:10 D0 0B 20 4A 13 A9 40 CF
12E1:8D 15 D0 4C 24 17 AD 00 C9
12E9:DC 29 0F 49 0F 29 03 C9 2A
12F1:01 D0 0D AD E1 03 C9 0F 05
12F9:F0 DA EE E1 03 4C 0F 13 C3
1301:C9 02 D0 D0 AD E1 03 C9 78
1309:02 90 C9 CE E1 03 AD E1 D2
1311:03 C9 0A B0 0E 18 69 30 4B
1319:8D C8 06 A9 30 8D C7 06 E0
1321:4C D5 12 18 69 26 8D C8 6E
1329:06 A9 31 8D C7 06 4C D5 80
1331:12 EE E3 03 D0 FB EE E4 02
1339:03 AD E4 03 C9 FF D0 F6 FF
1341:A9 08 8D E4 03 AD 00 DC E9
1349:60 A2 4F A9 20 9D 70 07 2C
1351:CA 10 FA 60 EE E3 03 D0 24
1359:FB EE E4 03 D0 F6 EE E5 2C
1361:03 AD E5 03 C9 02 D0 EC 46
1369:A9 00 8D E5 03 8D 15 D0 BD
1371:4C 9C 13 AD E1 03 D0 03 E1
1379:EE E1 03 20 73 12 4C A9 18
1381:13 EE 4B 12 C9 0F D0 05 A8
1389:A9 00 8D 4B 12 60 A9 30 80
1391:20 D2 FF A9 35 20 D2 FF E6
1399:4C 11 09 20 1B 08 AD E1 83
13A1:03 C9 0F F0 03 EE E1 03 47
13A9:A9 60 8D 21 13 8D 2F 13 C0
13B1:20 0F 13 A9 4C 8D 21 13 96
13B9:8D 2F 13 20 E0 13 20 C0 2B
13C1:16 A9 FF 8D 15 D0 AD E1 5F
13C9:03 85 04 A9 00 A2 CB 9D AD
13D1:34 03 CA E0 FF D0 F8 A5 15
13D9:04 8D E1 03 4C 62 15 A9 91
13E1:0E 85 D3 A9 16 85 D6 20 1A
13E9:4A 13 20 10 E5 A2 00 BD 76
13F1:0E 14 20 D2 FF E8 E0 0A C4
13F9:D0 F5 A9 60 8D 6E 13 20 AD
1401:55 13 A9 8D 8D 6E 13 20 13

Type It In

1409:BA 14 4C 73 12 9E 47 45 33
1411:54 20 52 45 41 44 59 EE C6
1419:D4 03 D0 FB EE D5 03 AD C8
1421:D5 03 C9 03 D0 F1 A9 00 FF
1429:8D D5 03 EE 0D D0 AD 0D F0
1431:D0 49 FF 8D 0F D4 AD 0D 21
1439:D0 C9 FF D0 DA A9 00 8D 54
1441:1B D0 4C 68 14 A9 40 8D 90
1449:1B D0 4C 18 14 AD D0 03 3A
1451:85 FC A9 60 8D DD 13 20 E0
1459:A9 13 A9 4C 8D DD 13 A5 C4
1461:FC 8D D0 03 4C C7 0B 8D DA
1469:15 D0 AD 18 07 C9 30 D0 18
1471:15 AD 17 07 C9 30 D0 03 96
1479:4C FC 14 A9 39 8D 18 07 5B
1481:CE 17 07 4C 8A 14 CE 18 D6
1489:07 A9 40 8D 15 D0 A9 A9 69
1491:8D 0C D0 A9 9E 8D 0D D0 4E
1499:A9 80 8D FE 07 AD 17 07 7C
14A1:8D 90 13 AD 18 07 8D 95 7F
14A9:13 4C 4E 14 A9 30 8D 90 33
14B1:13 A9 35 8D 95 13 4C C7 A6
14B9:0E A9 0C 85 D3 A9 16 85 24
14C1:D6 20 10 E5 A2 00 BD D5 24
14C9:14 20 D2 FF E8 E0 17 D0 28
14D1:F5 4C EC 14 05 43 48 41 ED
14D9:4E 47 45 20 54 4F 20 12 D7
14E1:E9 20 92 E9 11 9D 9D 9D CF
14E9:98 B7 B7 AD E1 03 18 69 D2
14F1:01 8D 86 DB 8D 87 DB 8D 5C
14F9:88 DB 60 20 4A 13 A9 0E 6B
1501:85 D3 A9 16 85 D6 20 10 51
1509:E5 A2 00 BD 31 15 20 D2 9B
1511:FF E8 E0 0A D0 F5 EE D7 46
1519:03 D0 FB EE D8 03 D0 F6 D2
1521:EE D9 03 AD D9 03 C9 05 E7
1529:D0 EC A2 00 9A 4C 0D 08 73
1531:9F 47 41 4D 45 20 4F 56 99
1539:45 52 A9 0E 8D 15 D0 A9 BC
1541:60 8D CB 0D 4C D5 12 20 47
1549:95 0D EE EC 03 D0 F8 EE 6A
1551:ED 03 AD ED 03 C9 09 D0 E9
1559:EE A9 00 8D ED 03 4C 37 89
1561:13 A9 4C 8D CB 0D A9 10 D7
1569:38 ED E1 03 4A 8D 7E 15 32
1571:EE 7E 15 4C C7 0B EE C1 24
1579:03 AD C1 03 C9 01 D0 F6 E3
1581:A9 00 8D C1 03 EE 34 03 8D
1589:4C CA 0B 20 97 17 C9 B0 4D
1591:B0 F9 29 F8 18 69 06 8D 47
1599:0D 50 20 97 17 29 F8 18 43

Chapter 2

15A1:69 01 8D 0C 50 20 D8 0C F3
15A9:A0 00 B1 FD C9 E9 D0 DB AD
15B1:AD 0C 50 8D 0E D0 AD 0D B4
15B9:50 8D 0F D0 A9 4C 8D F0 E8
15C1:0C 4C CD 15 A9 60 8D F0 EA
15C9:0C 4C 0A 16 A9 D0 8D E3 3F
15D1:0C 8D DA 0C 4C C7 0B AD C6
15D9:C5 03 F0 03 4C 37 16 EE 50
15E1:CA 03 AD CA 03 C9 70 D0 85
15E9:17 A9 00 8D CA 03 EE CB EE
15F1:03 AD CB 03 D0 0A AD 0F CB
15F9:D0 F0 08 A9 00 8D 0F D0 89
1601:4C 72 16 20 C5 15 4C 01 D0
1609:16 A9 50 8D E3 0C 8D DA D2
1611:0C 4C 8C 15 AD 0C D0 CD 46
1619:0E D0 D0 17 AD 0D D0 CD 1D
1621:0F D0 D0 0F A2 02 20 EC 5E
1629:16 A9 7F 8D C5 03 A9 00 21
1631:8D 0F D0 4C D8 0C EE C6 62
1639:03 AD C6 03 D0 30 EE C7 48
1641:03 AD C7 03 C9 16 D0 26 F1
1649:A9 00 8D C7 03 EE C8 03 E0
1651:AD C8 03 C9 0B 90 08 AD DB
1659:2D D0 49 08 8D 2D D0 AD 6A
1661:C8 03 C9 10 D0 08 A9 00 E6
1669:8D C8 03 8D C5 03 4C D5 70
1671:16 EE 2E D0 A2 0A BD 00 EF
1679:D0 18 7D 9F 16 38 ED 0C 37
1681:D0 C9 05 B0 12 BD 01 D0 8E
1689:18 7D A0 16 38 ED 0D D0 FA
1691:C9 05 B0 03 4C 68 17 CA 27
1699:CA 10 DB 4C 95 0D 04 0B 63
16A1:FF 0A FF 0A FF 0A 05 05 28
16A9:03 04 A9 60 8D 6E 13 A9 89
16B1:00 8D 0F D4 20 55 13 A9 96
16B9:8D 8D 6E 13 4C 46 14 EE A1
16C1:95 13 AD 95 13 C9 3A D0 91
16C9:08 A9 30 8D 95 13 EE 90 AA
16D1:13 4C 7A 0B EE C1 03 AD CC
16D9:C1 03 C9 04 D0 08 A9 00 1B
16E1:8D C1 03 4C C7 0B 4C D8 46
16E9:15 A2 03 FE 03 07 BD 03 4C
16F1:07 C9 3A D0 08 A9 30 9D 4D
16F9:03 07 CA 10 EE 4C 16 17 AF
1701:C0 00 D0 0E 84 FC A4 04 EF
1709:20 EA 16 88 C0 01 D0 F8 F1
1711:A4 FC A5 04 60 A2 04 BD 19
1719:03 07 9D F3 08 CA 10 F7 01
1721:4C 34 17 A2 04 A9 30 9D 54
1729:03 07 CA 10 FA 20 73 12 46
1731:4C 74 13 A2 00 38 BD B4 40
1739:06 FD 03 07 30 09 E9 01 35

Type It In

1741:10 13 E8 E0 05 D0 EE A2 53
1749:04 BD 03 07 9D B4 06 9D 23
1751:E3 08 CA 10 F4 60 AD 36 88
1759:03 C9 06 90 07 C9 15 B0 7F
1761:03 4C 98 16 4C AB 16 AD 83
1769:FE 03 D0 EA 4C AB 16 A9 87
1771:30 8D 7C 17 A2 00 AD 1B A7
1779:D4 9D 00 50 E8 D0 F7 EE E7
1781:7C 17 AD 7C 17 C9 50 D0 82
1789:EB A9 30 8D B0 17 A9 00 2C
1791:8D AF 17 4C B2 17 AD AF 17
1799:17 18 69 01 8D AF 17 AD 9D
17A1:B0 17 69 00 C9 4F D0 02 4A
17A9:A9 30 8D B0 17 AD CF 30 B4
17B1:60 A9 E1 8D 14 D4 A9 21 F7
17B9:8D 12 D4 A9 0F 8D 18 D4 1C
17C1:A9 00 8D 05 D4 8D FE 2F D0
17C9:8D FF 2F 4C F9 18 AE 36 2D
17D1:03 BD DB 17 8D 0F D4 4C 7C
17D9:E0 0B 00 80 7C 78 74 70 62
17E1:6E 6C 6A 68 66 00 00 00 69
17E9:00 00 00 00 00 00 00 00 18
17F1:00 00 00 00 A9 08 8D 09 B1
17F9:D4 8D 0A D4 A9 77 8D 06 D0
1801:D4 8D 0D D4 A9 41 8D 0B 66
1809:D4 A9 4C 8D DD 13 A9 00 FE
1811:8D 0F D4 4C C2 18 AD FE FB
1819:2F C9 01 D0 1B AE FF 2F 43
1821:BD BA 18 8D 08 D4 A9 28 C9
1829:8D 01 D4 A9 81 8D 04 D4 B4
1831:A9 80 8D 04 D4 4C B7 18 A7
1839:C9 0F D0 28 A9 40 8D 0B 23
1841:D4 A9 41 8D 0B D4 A9 1E 64
1849:8D 01 D4 EE FF 2F AE FF 24
1851:2F BD BA 18 8D 08 D4 A9 41
1859:81 8D 04 D4 A9 80 8D 04 E9
1861:D4 4C B7 18 C9 10 D0 0F C6
1869:AE FF 2F BD BA 18 18 69 82
1871:02 8D 08 D4 4C B7 18 C9 8F
1879:18 D0 23 A9 40 8D 0B D4 0C
1881:A9 41 8D 0B D4 AE FF 2F C9
1889:BD BA 18 18 69 04 8D 08 4A
1891:D4 A9 81 8D 04 D4 A9 80 E6
1899:8D 04 D4 4C B7 18 C9 1E C0
18A1:D0 14 EE FF 2F A9 00 8D CA
18A9:FE 2F AD FF 2F C9 08 D0 5C
18B1:05 A9 00 8D FF 2F 4C D3 D0
18B9:18 04 08 0C 10 14 10 0C B5
18C1:08 78 A9 D9 8D 14 03 A9 53
18C9:18 8D 15 03 58 A9 00 4C F1
18D1:33 0C EE FE 2F 4C 31 EA 64
18D9:EE FD 2F AD FD 2F 29 01 C1

Chapter 2

18E1:D0 03 4C 31 EA 4C 17 18 A6
18E9:AD 8D 02 D0 03 4C 77 15 EE
18F1:A9 00 8D 0F D4 4C E9 18 5D
18F9:A9 2B 8D 18 03 A9 15 8D 73
1901:19 03 A9 00 8D 10 D0 8D 91
1909:1B D0 4C F5 17 00 00 00 9E

Program 5

Bump-N-Run

Once you have a copy of "MLX" typed in and saved to tape or disk, load and run it. You'll have to provide a starting address and an ending address. For "Bump-N-Run," the addresses are

Starting Address: C000

Ending Address: C69F

Save a copy to disk or tape when you've finished typing, and before you quit MLX.

Bump-N-Run

For mistake-proof program entry, use "MLX: Machine Language Entry Program" to type in this program.

```
C000:A9 00 8D 20 D0 8D 21 D0 D9
C008:8D 20 C7 8D 21 C7 AD 16 C3
C010:03 8D 18 03 AD 17 03 8D 07
C018:19 03 A9 14 8D 18 D0 A9 75
C020:80 8D 91 02 A9 93 20 D2 46
C028:FF A9 00 A2 80 9D 80 3F F8
C030:CA 10 FA A9 18 8D 99 3F 7E
C038:8D A2 3F A9 3C 8D 9C 3F 3C
C040:8D 9F 3F A2 33 BD 4C C6 72
C048:9D C4 3F CA 10 F7 20 1A 19
C050:C5 78 A9 05 8D 14 03 A9 C4
C058:C2 8D 15 03 58 20 80 C2 78
C060:A9 FF 8D 18 C7 A9 00 8D 5C
C068:15 C7 8D 16 C7 8D 17 C7 E3
C070:8D 1F C7 8D 19 C7 8D 1A 6F
C078:C7 A2 05 A9 00 9D 00 C7 FF
C080:9D 06 C7 A9 5A 9D 0C C7 0F
C088:CA 10 F0 AD 16 C7 29 03 91
C090:C9 03 D0 50 AD 15 C7 D0 F8
C098:48 AE 00 C7 AC 01 C7 AD 0D
C0A0:02 C7 8D 00 C7 AD 03 C7 89
C0A8:8D 01 C7 8E 02 C7 8C 03 5E
C0B0:C7 AD 17 C7 F0 1B C9 01 69
C0B8:F0 0D A9 01 8D 17 C7 A9 3D
C0C0:0B 8D 15 D0 4C D1 C0 A9 AF
C0C8:03 8D 17 C7 A9 13 8D 15 58
C0D0:D0 A2 03 BD 06 C7 9D 0C 36
C0D8:C7 CA 10 F7 A9 01 8D 15 F3
```

Chapter 2

C0E0:C7 4C 2A C1 AD 17 C7 D0 E4
C0E8:18 AD 04 C7 CD 19 C7 D0 12
C0F0:0B AD 05 C7 CD 1A C7 D0 B7
C0F8:03 4C 01 C1 A9 FF 8D 18 CB
C100:C7 A9 00 8D 15 C7 AD 16 E3
C108:C7 2D 18 C7 29 07 C9 05 38
C110:D0 0A A9 01 8D 17 C7 A9 C5
C118:0B 8D 15 D0 C9 06 D0 0A 46
C120:A9 03 8D 17 C7 A9 13 8D F4
C128:15 D0 A2 00 8E 14 C7 A9 BC
C130:09 8D 12 C7 AD 01 DC 8D 13
C138:13 C7 20 DF C2 A2 02 8E 6C
C140:14 C7 A9 12 8D 12 C7 AD 08
C148:00 DC 8D 13 C7 20 DF C2 27
C150:A2 04 8E 14 C7 A9 24 8D F3
C158:12 C7 AD 04 C7 8D 19 C7 3B
C160:AD 05 C7 8D 1A C7 BD 0C 45
C168:C7 DD 06 C7 F0 06 DE 0C ED
C170:C7 4C 7E C1 A9 5A 9D 0C D4
C178:C7 A0 02 20 E4 C3 EE 14 72
C180:C7 AE 14 C7 BD 0C C7 DD 1E
C188:06 C7 F0 06 DE 0C C7 4C 82
C190:02 C2 A9 5A 9D 0C C7 A0 ED
C198:02 20 20 C4 AE 1F C7 F0 E7
C1A0:61 A9 00 8D 15 D0 FE 1F 21
C1A8:C7 A2 01 18 20 F0 FF AE CD
C1B0:1F C7 BD 1F C7 0A AA BD D8
C1B8:80 C6 20 D2 FF E8 BD 80 FE
C1C0:C6 20 D2 FF A2 64 A0 FF F1
C1C8:88 D0 FD CA D0 F8 AE 1F 18
C1D0:C7 BD 1F C7 C9 0F D0 27 5B
C1D8:A2 0C A0 10 18 20 F0 FF E8
C1E0:A2 2C 20 E3 C5 A2 0E A0 78
C1E8:0B 18 20 F0 FF A2 41 20 38
C1F0:E3 C5 AD 01 DC 0D 00 DC 95
C1F8:29 10 D0 F6 4C 00 C0 4C CE
C200:5D C0 4C 8B C0 AD 1E D0 70
C208:8D 16 C7 29 03 C9 03 D0 7B
C210:0E A2 00 20 22 C2 E8 20 55
C218:22 C2 A9 0F 8D 19 D0 4C 44
C220:31 EA BD 00 D0 DD 02 D0 83
C228:F0 55 10 2B DE 00 D0 DE A7
C230:06 D0 10 0C E0 01 F0 08 A4
C238:A9 F6 2D 10 D0 8D 10 D0 A4
C240:FE 02 D0 FE 08 D0 D0 37 2C
C248:E0 01 F0 33 A9 12 0D 10 8F
C250:D0 8D 10 D0 4C 7F C2 FE 95
C258:00 D0 FE 06 D0 D0 0C E0 15
C260:01 F0 08 A9 09 0D 10 D0 AB
C268:8D 10 D0 DE 02 D0 DE 08 D9
C270:D0 10 0C E0 01 F0 08 A9 77

Type It In

C278:ED 2D 10 D0 8D 10 D0 60 FD
C280:A9 4A 8D 00 D0 8D 06 D0 B8
C288:A9 90 8D 01 D0 8D 03 D0 5C
C290:8D 05 D0 8D 07 D0 8D 09 B0
C298:D0 8D 0B D0 A9 0F 8D 02 FE
C2A0:D0 8D 08 D0 A9 AC 8D 0A 25
C2A8:D0 8D 04 D0 A9 12 8D 10 48
C2B0:D0 A9 FF 8D F8 07 8D F9 DA
C2B8:07 A9 FE 8D FA 07 8D FB EF
C2C0:07 8D FC 07 A9 04 8D 27 DC
C2C8:D0 A9 07 8D 28 D0 A9 01 B3
C2D0:8D 29 D0 8D 2A D0 8D 2B 35
C2D8:D0 A9 07 8D 15 D0 60 BD 55
C2E0:0C C7 DD 06 C7 F0 0C DE 73
C2E8:0C C7 4E 13 C7 4E 13 C7 C6
C2F0:4C 00 C3 A9 5A 9D 0C C7 D8
C2F8:20 AD C3 A0 00 20 E4 C3 8A
C300:EE 14 C7 AE 14 C7 BD 0C 2F
C308:C7 DD 06 C7 F0 0C DE 0C A9
C310:C7 4E 13 C7 4E 13 C7 4C 88
C318:27 C3 A9 5A 9D 0C C7 20 CB
C320:AD C3 A0 00 20 20 C4 AD 3C
C328:14 C7 4D 17 C7 D0 7D AE F1
C330:14 C7 4E 13 C7 B0 08 BD 7D
C338:1B C7 F0 0B 4C AC C3 A9 54
C340:00 9D 1B C7 4C AC C3 BD 69
C348:00 C7 8D 05 C7 BD 06 C7 CC
C350:8D 0B C7 BD FF C6 8D 04 70
C358:C7 BD 05 C7 8D 0A C7 A9 1E
C360:00 9D FF C6 9D 00 C7 9D D5
C368:05 C7 9D 06 C7 BD 00 D0 7E
C370:8D 05 D0 8D 0B D0 BD FF 0A
C378:CF 8D 04 D0 8D 0A D0 AD BC
C380:12 C7 2D 10 D0 F0 0B AD B7
C388:10 D0 09 24 8D 10 D0 4C 4A
C390:9A C3 AD 10 D0 29 DB 8D 7D
C398:10 D0 A9 07 8D 15 D0 A9 0E
C3A0:00 8D 17 C7 AD 12 C7 49 79
C3A8:FF 8D 18 C7 60 AC 14 C7 B8
C3B0:B9 00 C7 D0 05 A2 00 4C 1A
C3B8:C5 C3 29 80 F0 05 A2 01 23
C3C0:4C C5 C3 A2 02 BC F9 C5 3F
C3C8:4E 13 C7 B0 03 BC FC C5 0B
C3D0:4E 13 C7 B0 03 BC FF C5 19
C3D8:AE 14 C7 98 18 7D 00 C7 BD
C3E0:9D 00 C7 60 BD 00 C7 F0 A4
C3E8:33 29 80 F0 1A BD 01 D0 0E
C3F0:D9 02 C6 B0 06 20 0E C5 5C
C3F8:4C 1C C4 DE 01 D0 DE 07 44
C400:D0 20 EA C4 4C 1F C4 BD C9
C408:01 D0 C8 D9 02 C6 90 06 4F

Chapter 2

C410:20 0E C5 4C 89 C4 FE 01 09
C418:D0 FE 07 D0 20 EA C4 60 4E
C420:BD 00 C7 F0 60 10 31 BD F3
C428:FF CF D9 06 C6 B0 13 AD 0E
C430:10 D0 2D 12 C7 D0 0B 20 74
C438:B4 C4 B0 06 20 0E C5 4C D4
C440:85 C4 DE FF CF DE 05 D0 6E
C448:10 3F AD 12 C7 49 FF 2D 11
C450:10 D0 8D 10 D0 4C 89 C4 58
C458:BD FF CF C8 D9 06 C6 90 4C
C460:13 AD 10 D0 2D 12 C7 F0 20
C468:0B 20 8D C4 B0 06 20 0E 69
C470:C5 4C 89 C4 FE FF CF FE 04
C478:05 D0 D0 09 AD 12 C7 0D B5
C480:10 D0 8D 10 D0 20 EA C4 9A
C488:60 20 EA C4 60 E0 05 D0 55
C490:57 AD 17 C7 D0 52 BD FF DB
C498:CF C9 36 10 06 20 DB C4 71
C4A0:4C B3 C4 C9 38 30 3F A9 1D
C4A8:9C 20 D2 FF A9 01 8D 1F 6E
C4B0:C7 A0 0A 60 E0 05 D0 30 7A
C4B8:AD 17 C7 D0 2B BD FF CF 05
C4C0:C9 24 30 06 20 DB C4 4C E4
C4C8:DA C4 C9 20 10 18 A9 9E FE
C4D0:20 D2 FF A9 02 8D 1F C7 06
C4D8:A0 23 60 AD 05 D0 C9 7B DC
C4E0:30 06 C9 A6 10 02 38 60 01
C4E8:18 60 BD 00 C7 30 0D C9 31
C4F0:52 30 05 A9 52 9D 00 C7 BB
C4F8:9D 06 C7 60 C9 AE 10 05 FF
C500:A9 AE 9D 00 C7 49 FF 9D C0
C508:06 C7 FE 06 C7 60 BD 00 04
C510:C7 49 FF 9D 00 C7 FE 00 C8
C518:C7 60 A9 81 20 D2 FF A2 DB
C520:00 A0 02 18 20 F0 FF A9 04
C528:B0 20 D2 FF A2 21 20 C7 10
C530:C5 A9 AE 20 D2 FF A2 02 BE
C538:20 F0 FF A9 ED 20 D2 FF 40
C540:A2 21 20 C7 C5 A9 BD 20 56
C548:D2 FF A2 01 C8 20 F0 FF 4A
C550:A2 0A 20 E3 C5 A2 01 A0 4D
C558:0F 20 F0 FF A2 21 20 E3 4F
C560:C5 A2 01 A0 1C 20 F0 FF E4
C568:A2 15 20 E3 C5 A9 99 20 F4
C570:D2 FF A2 03 A0 02 20 F0 28
C578:FF A9 B0 20 D2 FF A2 21 83
C580:20 C7 C5 A9 AE 20 D2 FF FC
C588:A2 16 20 F0 FF A9 AD 20 20
C590:D2 FF A2 21 20 C7 C5 A9 41
C598:BD 20 D2 FF A2 04 20 F0 BB
C5A0:FF A9 05 20 D0 C5 A2 10 2C

Type It In

C5A8:20 F0 FF A9 05 20 D0 C5 2B
C5B0:A2 04 A0 25 20 F0 FF A9 63
C5B8:05 20 D0 C5 A2 10 20 F0 CB
C5C0:FF A9 05 20 D0 C5 60 A9 61
C5C8:C0 20 D2 FF CA 10 F8 60 FF
C5D0:8D 22 C7 A9 DD 20 D2 FF 54
C5D8:E8 18 20 F0 FF CE 22 C7 39
C5E0:10 F1 60 BC 0A C6 CA BD 87
C5E8:0A C6 20 D2 FF CA BD 0A 0D
C5F0:C6 20 D2 FF CA 88 D0 F6 53
C5F8:60 00 01 FF FF FF FE 01 D3
C600:02 01 4F D2 48 D9 2C 2D 15
C608:25 34 30 20 3A 45 4C 50 0D
C610:52 55 50 9C 09 30 20 3A 73
C618:57 4F 4C 4C 45 59 9E 09 49
C620:4E 55 52 2D 4E 2D 50 4D 5C
C628:55 42 9F 0A 52 45 56 4F 29
C630:20 45 4D 41 47 9F 09 54 FB
C638:45 53 45 52 20 4F 54 20 12
C640:45 52 49 46 20 53 53 45 CC
C648:52 50 9A 13 18 00 00 18 70
C650:00 00 18 00 00 18 00 00 41
C658:3C 00 00 66 00 00 C3 00 F1
C660:01 81 80 3F 00 FC 3F 00 45
C668:FC 01 81 80 00 C3 00 00 FB
C670:66 00 00 3C 00 00 18 00 25
C678:00 18 00 00 18 00 00 18 E4
C680:A0 30 A0 31 A0 32 A0 33 D3
C688:A0 34 A0 35 A0 36 A0 37 31
C690:A0 38 A0 39 31 30 31 31 02
C698:31 32 31 33 31 34 31 35 96

Program 6

Ringside Karate

After you've typed in "MLX" and saved it to tape or disk, load and run the program. It will ask you to provide two addresses before you can begin entering "Ringside Karate." The addresses are

Starting Address: 3000

Ending Address: 45BF

When you've finished typing in the game (it will probably take you more than one typing session), make sure you save a copy to tape or disk before exiting MLX.

Ringside Karate

For mistake-proof program entry, use "MLX: Machine Language Entry Program" to type in this program.

```
3000:00 0F C0 00 15 50 00 3F 65
3008:F0 00 3F A0 00 3F B0 00 31
3010:3B A8 00 3A A0 00 2A A0 D5
3018:00 2A 00 00 3A 00 00 5E 33
3020:70 01 57 5C 01 75 57 01 37
3028:5D 55 03 57 5A 00 D5 AA 8B
3030:00 36 A8 00 1D 70 00 15 F2
3038:70 00 15 70 00 3F F0 0F 68
3040:00 00 00 00 00 00 00 00 A0
3048:00 00 00 00 00 00 00 00 A8
3050:00 00 00 00 00 00 00 00 B0
3058:00 00 00 00 00 00 02 00 BC
3060:00 02 80 00 0A C0 00 6B 10
3068:00 00 6C 00 00 70 00 00 18
3070:C0 00 00 00 00 00 00 00 31
3078:00 00 00 00 00 00 00 00 B E3
3080:00 15 70 00 55 5C 00 55 A5
3088:5C 01 55 57 01 55 57 01 84
3090:55 57 05 5F 55 05 70 15 BC
3098:05 70 15 15 C0 05 15 C0 90
30A0:05 15 C0 05 17 00 01 17 03
30A8:00 01 17 00 01 17 0F 01 AF
30B0:17 3F C1 2B FF FE 3F C0 92
30B8:3F 00 00 00 00 00 00 0F C7
30C0:00 00 00 00 00 00 00 00 21
30C8:00 00 00 00 00 00 00 00 29
30D0:00 00 C0 00 00 C0 00 00 4C
```

Type It In

30D8:C0 00 00 70 00 00 70 00 81
30E0:00 70 00 00 70 00 00 70 51
30E8:00 00 70 00 00 70 00 00 19
30F0:70 00 00 AC 00 00 FC 00 4E
30F8:00 00 00 00 00 00 00 0F 68
3100:00 15 70 00 55 5C 00 55 27
3108:5F 01 55 57 01 55 57 01 87
3110:55 57 01 5F 55 01 70 15 AD
3118:01 70 15 01 70 05 01 70 D3
3120:05 01 70 05 01 5C 01 01 20
3128:5C 01 00 57 01 00 5B C1 EE
3130:00 6F F1 00 BF FE 03 F0 9D
3138:0F 00 00 00 00 00 00 0F 31
3140:00 00 00 00 00 00 00 00 A2
3148:FC 00 01 55 00 03 FF 00 AA
3150:03 FA 00 03 FB 00 03 BA C3
3158:00 03 AA 00 02 AA 00 02 8D
3160:A0 00 03 A0 00 05 E7 00 61
3168:15 75 00 1D 55 00 1D 55 BE
3170:00 17 55 00 15 A9 00 35 C7
3178:A9 00 0F 57 00 01 57 00 B9
3180:00 00 00 00 00 00 00 00 E2
3188:00 00 00 00 00 00 00 00 EA
3190:00 00 00 00 00 80 00 00 F4
3198:00 00 00 00 00 00 00 00 FA
31A0:00 00 00 00 FF C2 80 55 64
31A8:6A 80 55 6A 80 5F C0 00 B4
31B0:70 00 00 C0 00 00 00 00 57
31B8:00 00 00 00 00 00 00 00 1B
31C0:00 01 57 00 03 FF 00 01 67
31C8:57 00 01 57 00 01 57 00 1F
31D0:01 55 00 01 55 00 01 55 1B
31D8:00 01 5D 00 01 5C 00 01 A1
31E0:5C 00 05 5C 00 15 70 00 0D
31E8:55 C0 01 57 00 05 5C 00 88
31F0:15 70 0F 0A 70 FF 02 AF B3
31F8:FF 0F F0 00 00 00 00 00 3D
3200:00 00 00 00 00 00 00 00 64
3208:00 00 00 00 C0 00 00 70 E2
3210:00 00 5C 00 00 57 00 00 5D
3218:55 C0 00 15 C0 00 35 C0 D9
3220:00 35 C0 00 05 C0 00 05 1A
3228:C0 00 05 C0 00 05 C0 00 2F
3230:C5 C0 00 C5 C0 00 FA B0 B0
3238:00 FF F0 00 00 00 00 00 BA
3240:00 3F 00 00 55 40 00 FF 20
3248:C0 00 FE 80 00 FE C0 00 72
3250:EE A0 00 EA 80 00 AA 80 DC
3258:00 A8 00 00 A8 00 00 EA 17
3260:D5 00 7A D5 01 5F 55 01 8D
3268:55 50 05 55 50 05 55 50 13

Chapter 2

3270:15 D5 54 17 C5 57 15 A5 2C
3278:5D 05 A1 75 00 00 D5 00 04
3280:00 00 00 00 00 00 00 00 E4
3288:00 00 00 00 00 00 00 00 EC
3290:A0 00 00 A0 00 02 B0 00 B8
3298:6A C0 00 6B 00 00 6C 00 F1
32A0:00 70 00 00 C0 00 18 00 57
32A8:01 58 00 15 58 00 55 5C BE
32B0:05 55 5C 15 55 F0 55 5C 3F
32B8:00 55 70 00 55 C0 00 00 2E
32C0:00 03 55 00 00 55 00 00 E5
32C8:55 00 00 55 00 01 55 00 DB
32D0:01 57 00 01 5C 00 01 50 D0
32D8:00 05 70 00 05 70 00 05 7B
32E0:70 00 15 C0 00 15 C0 00 02
32E8:15 C0 00 15 C0 00 17 00 8D
32F0:00 57 00 00 57 00 00 AB 91
32F8:0F 00 FF FF 00 00 00 00 E4
3300:57 00 00 7C 00 00 70 00 BA
3308:00 C0 00 00 00 00 00 00 9E
3310:00 00 00 00 00 00 00 00 76
3318:00 00 00 00 00 00 00 00 7E
3320:00 00 00 00 00 00 00 00 86
3328:00 00 00 00 00 00 00 00 8E
3330:00 00 00 0F FC 00 FF FF 6F
3338:FF FC 0F FF 00 00 00 00 BF
3340:00 FC 00 01 55 00 03 FF A6
3348:00 03 FA 00 03 FB 00 03 D9
3350:BA 80 03 AA 00 02 AA 00 9C
3358:02 A0 00 03 A0 00 01 E0 FF
3360:00 05 75 55 05 55 55 05 39
3368:D5 FF 05 75 C0 01 5F C0 3B
3370:03 55 6A 00 D5 6A 00 55 A8
3378:C0 00 55 C0 00 FF C0 00 77
3380:00 00 00 00 00 00 00 00 E6
3388:00 00 00 00 B0 00 00 AC 21
3390:00 00 B0 00 00 B0 00 00 CF
3398:B0 00 00 B0 00 00 B0 00 C3
33A0:00 B0 00 00 B0 00 00 00 B8
33AB:00 00 00 00 00 F0 00 00 D2
33B0:A0 00 00 80 00 00 20 00 AF
33B8:00 00 00 00 00 00 00 0B 2A
33C0:00 15 7F 00 55 55 00 55 B1
33C8:55 00 55 55 01 55 55 01 E2
33D0:55 FF 01 5F 00 01 5C 00 B4
33D8:01 70 00 01 70 00 01 70 E1
33E0:00 01 70 00 01 70 00 01 60
33E8:70 00 01 70 00 01 70 00 93
33F0:01 70 0F 02 B3 FF 03 FF 99
33F8:F0 00 00 00 00 00 00 0F E6
3400:00 00 00 00 00 00 50 00 09

Type It In

3408:00 50 00 00 50 00 00 50 57
3410:00 00 50 00 00 50 00 00 C3
3418:50 00 00 A0 00 00 20 00 F2
3420:00 20 00 00 00 00 00 00 90
3428:00 00 00 00 00 00 00 00 90
3430:00 00 00 C0 00 00 F0 00 86
3438:00 00 00 00 00 00 00 00 A0
3440:00 00 00 00 00 00 00 00 A8
3448:00 00 00 00 00 00 00 00 B0
3450:00 00 00 00 00 00 00 00 B8
3458:00 00 00 00 00 00 00 00 C0
3460:00 00 00 00 00 00 00 00 C8
3468:03 F0 00 05 54 00 0F FC 9C
3470:00 0F E8 00 0F EC 00 0E F3
3478:EA 00 0E A8 00 0A A8 02 1E
3480:00 00 00 00 00 00 00 00 E8
3488:00 00 00 00 00 00 00 00 F0
3490:00 00 00 00 00 00 00 00 F8
3498:00 00 00 00 00 00 00 00 01
34A0:00 00 00 00 00 00 00 00 09
34A8:00 00 00 00 00 00 00 00 11
34B0:00 00 00 00 00 00 00 00 19
34B8:00 00 00 00 00 00 00 02 23
34C0:00 0A 80 00 0E 80 00 07 35
34C8:97 00 05 D5 00 15 55 00 F9
34D0:55 55 00 55 5C 01 5D 5C 8C
34D8:03 C5 5C 0D 75 5C 0D 5F 27
34E0:FF 35 55 55 35 D5 55 35 77
34E8:F5 55 35 C5 55 B5 C0 00 A7
34F0:B5 C0 FF BA CF FF FF FF 8E
34F8:FF 00 00 00 00 00 00 02 63
3500:00 00 00 00 00 00 00 00 6A
3508:00 C0 00 00 70 00 00 5F 85
3510:00 00 55 A8 00 15 A8 00 55
3518:0F 00 00 35 C0 00 D5 70 7F
3520:00 55 50 00 55 5C 00 50 56
3528:57 00 40 15 C0 00 15 A0 68
3530:FF 05 A8 FF 01 FC FF FF EC
3538:FC 00 00 00 00 00 00 02 23
3540:00 00 00 00 00 00 00 00 AA
3548:00 00 00 00 00 00 00 00 B2
3550:00 00 00 00 00 00 00 00 BA
3558:00 00 00 00 00 00 00 00 C2
3560:00 00 00 00 00 00 00 00 CA
3568:00 00 00 00 00 00 00 00 D2
3570:00 00 00 00 00 00 00 00 DA
3578:00 00 00 00 00 00 00 00 E2
3580:00 00 00 00 00 00 00 00 EA
3588:00 00 00 00 00 00 00 00 F2
3590:00 00 00 00 00 00 00 00 FA
3598:00 00 00 00 00 00 00 00 03

Chapter 2

35A0:00 00 00 00 00 00 00 00 00 0B
35A8:00 00 00 00 00 00 00 00 00 13
35B0:00 00 00 00 00 00 00 00 00 1B
35B8:00 00 00 00 00 00 00 00 00 23
35C0:00 00 00 00 00 00 00 00 00 2B
35C8:00 00 00 00 00 00 00 00 00 33
35D0:00 00 00 00 00 00 00 00 00 3B
35D8:00 00 00 00 80 00 1E A0 24
35E0:00 DE A1 55 DE AB 55 DF BC
35E8:EA D5 DF AB 5F DF ED 55 A0
35F0:FF FF 55 00 03 D5 00 00 75
35F8:FF 00 00 00 00 00 00 00 63
3600:00 00 00 00 00 00 00 00 6C
3608:00 00 00 00 00 00 00 00 74
3610:00 00 00 00 00 00 07 00 8A
3618:00 15 C0 00 55 70 01 55 A5
3620:70 D5 5D 70 D5 71 70 D7 1A
3628:C1 70 D5 7F 70 75 55 F2 3B
3630:5D 55 5E 57 F5 5A D6 AD 56
3638:5A F6 AF FF 3F F0 00 00 43
3640:00 00 00 00 00 00 00 00 AC
3648:00 00 00 00 00 00 00 00 B4
3650:00 00 00 00 00 00 00 00 BC
3658:00 00 00 00 00 00 00 00 C4
3660:B0 00 02 B0 00 00 AC 00 C9
3668:00 2B 00 00 0B 00 00 03 FA
3670:00 00 03 00 00 00 00 00 3D
3678:00 00 00 00 00 00 00 0B EF
3680:03 F0 00 05 54 00 0F FC B8
3688:00 0A FC 00 0E FC 00 2A A5
3690:EC 00 0A AC 00 0A A8 00 F8
3698:00 A8 00 00 A4 00 07 9F 02
36A0:C0 1F 7F C0 7F F7 F0 FF EE
36A8:DF F0 AF 7F F0 AA FD C0 1E
36B0:2A 77 C0 03 DF C0 03 FF 60
36B8:00 03 FF 00 01 57 00 0F 5A
36C0:00 00 00 00 00 00 00 00 2D
36C8:00 00 00 00 00 00 00 00 35
36D0:00 00 00 00 00 00 00 00 3D
36D8:00 00 00 00 00 02 00 00 4D
36E0:02 00 00 02 00 00 02 00 72
36E8:00 02 00 00 02 00 00 02 E7
36F0:00 00 02 00 00 05 00 00 B1
36F8:0F 00 00 00 00 00 00 0F FB
3700:02 AB 00 0A AA C0 0A AA 12
3708:C0 2A AA B0 2A AA B0 2A 49
3710:AA B0 AA FA AC AB 02 AC C9
3718:AB 02 AC AC 00 AB AC 00 45
3720:AB AC 00 AB B0 00 2B B0 D6
3728:00 2B B0 00 2B B0 F0 2B A0
3730:B3 FC 2B 7F FF 17 FC 03 6E

Type It In

3738:FF 00 00 00 00 00 00 0F B5
3740:02 AB 00 0A AA C0 0A AA 52
3748:C0 2A AA B0 2A AA B0 2A 89
3750:AA B0 AA FA B0 AB 02 B0 2E
3758:AB 02 B0 AC 02 B0 AC 02 2C
3760:B0 AC 02 B0 B0 0A B0 B0 5D
3768:0A B0 B0 2A C0 BF 1A C0 BA
3770:BF C6 C0 7F FD C0 F0 0F 64
3778:C0 00 00 00 00 00 00 0F 56
3780:00 00 00 00 00 00 00 00 EE
3788:00 00 00 00 00 00 00 00 F6
3790:00 00 00 00 00 00 00 02 01
3798:00 00 00 00 00 00 00 00 07
37A0:00 00 00 00 02 81 55 02 D1
37A8:AB FF 02 AB FF 00 01 5F 49
37B0:00 00 07 00 00 01 00 00 04
37B8:00 00 00 00 00 00 00 00 27
37C0:00 00 00 00 00 00 3F 00 AD
37C8:00 55 40 00 FF C0 00 AF 47
37D0:C0 00 EF C0 00 AE C0 00 E5
37D8:AA C0 00 AA 80 00 0A 80 10
37E0:00 0A 40 00 79 F0 00 F7 61
37E8:FC 00 FF DF 00 FF DF 00 93
37F0:FF 7F 00 EA FF 00 EA F4 B8
37F8:00 3F 50 00 3F F0 00 00 FE
3800:00 00 00 00 00 00 00 00 70
3808:00 00 00 00 00 00 00 00 78
3810:00 02 00 00 0A 00 00 2A 7B
3818:00 00 AA 00 00 AB 00 00 8C
3820:AC 00 00 AC 00 00 AC 00 0B
3828:00 AC 00 00 AC 00 00 AC D5
3830:00 00 AC 00 00 AC 00 01 E9
3838:5F 00 03 FF 00 00 00 00 B8
3840:2A B0 00 15 70 00 2A B0 CB
3848:00 2A B0 00 2A B0 00 AA 18
3850:B0 00 AA B0 00 AA B0 00 85
3858:BA B0 00 CA B0 00 0A B0 49
3860:00 0A AC 00 02 AB 00 00 A7
3868:AA C0 00 2A B0 00 0A AC 47
3870:F0 02 A8 FF 02 5C FF F5 66
3878:7C 00 0F FC 00 00 00 00 D8
3880:00 00 00 00 00 00 00 00 F0
3888:00 00 00 00 00 00 00 00 F8
3890:0A 00 00 0A C0 00 02 B0 61
3898:00 00 AD 00 00 2B 00 00 6B
38A0:0B 00 00 01 1B C0 00 1A 9C
38A8:BC 00 1A AB 00 0A AA F0 E3
38B0:0A AA AC 00 AA AB 00 0A 74
38B8:AA 00 02 AA 00 00 AA 00 BE
38C0:03 F0 00 05 54 00 0F FC FC
38C8:00 0A FC 00 0E FC 00 2A E9

Chapter 2

38D0:EC 00 0A AC 00 0A A8 00 3D
38D8:00 A8 00 00 A8 00 55 A9 0D
38E0:00 FD A7 C0 FF 5F C0 5F 30
38E8:FF F0 0F FF F0 0F FF FC 38
38F0:3F FF FC 7F FF 7C 9F FA C4
38F8:FC A7 FA F4 A9 FC 50 00 62
3900:00 00 2A 00 00 0A 00 00 DF
3908:02 00 00 00 00 00 00 00 7B
3910:00 00 00 00 00 00 00 00 82
3918:00 00 00 00 00 00 00 00 8A
3920:00 00 00 00 00 00 00 00 92
3928:00 00 00 00 00 00 00 00 9A
3930:00 00 00 00 0F FC FF FF 0F
3938:FF FF C0 3F 00 00 00 00 B6
3940:AA C0 00 AA C0 00 AA C0 FE
3948:00 AA C0 00 AA B0 00 2A BF
3950:B0 00 0A B0 00 0A B0 00 F0
3958:02 AC 00 02 AC 00 02 AC 2D
3960:00 00 AB 00 00 AB 00 00 F6
3968:AB 00 00 AB 00 00 2B 00 C1
3970:00 2A 00 00 2A 00 FF D5 94
3978:00 FF FF 00 00 00 00 00 EA
3980:00 00 00 00 00 00 00 00 F2
3988:00 00 00 00 00 00 0B 00 11
3990:00 2B 00 00 0B 00 00 0B 31
3998:00 00 0B 00 00 0B 00 00 98
39A0:0B 00 00 0B 00 00 0B 00 5F
39A8:00 00 00 00 00 00 00 0F 2A
39B0:00 00 0A 00 00 02 00 00 6C
39B8:08 00 00 00 00 00 00 0B 3A
39C0:00 3F 00 00 55 40 00 FF AE
39C8:C0 00 AF C0 00 EF C0 02 E0
39D0:AE C0 00 AA C0 00 AA 80 51
39D8:00 0A 80 00 0A 40 55 59 33
39E0:F0 FF F7 F0 FF FF F0 55 11
39E8:FD F0 00 F7 F0 35 5F F0 22
39F0:AF FF 70 AF FF F0 05 55 67
39F8:C0 03 FF C0 01 55 C0 00 77
3A00:00 00 00 00 00 00 00 00 74
3A08:0A 00 00 0A 00 00 0A 00 36
3A10:00 0A 00 00 0A 00 00 0A 61
3A18:00 00 0A 00 00 05 00 00 E1
3A20:04 00 00 04 00 00 00 00 D6
3A28:00 00 00 00 00 00 00 00 9C
3A30:00 00 00 00 00 03 00 00 B0
3A38:0F 00 00 00 00 00 00 00 34
3A40:FE AB 00 AA AA C0 AA AA 22
3A48:C0 AA AA C0 AA AA B0 C0 4B
3A50:AA B0 C0 0A B0 C0 0A B0 4C
3A58:C0 02 B0 C0 02 B0 00 02 A4
3A60:B0 00 02 B0 00 02 B0 00 E1

3A68:02 B0 00 02 B0 00 02 B0 64
 3A70:3F C2 B0 FF FD 70 C0 FF 7E
 3A78:F0 00 00 00 00 00 00 0F 74
 3A80:00 00 00 00 00 00 00 00 F4
 3A88:00 00 00 00 00 00 00 00 FC
 3A90:00 00 00 00 00 00 00 00 05
 3A98:00 00 00 00 00 00 00 00 0D
 3AA0:00 00 00 00 00 00 00 00 15
 3AA8:00 00 00 00 00 00 00 00 1D
 3AB0:00 00 00 00 00 00 00 00 25
 3AB8:00 00 00 00 00 00 00 0B 38
 3AC0:00 00 00 00 00 00 00 00 35
 3AC8:00 00 00 00 00 00 00 00 3D
 3AD0:00 00 00 00 00 00 00 00 45
 3AD8:00 00 00 00 00 00 00 00 4D
 3AE0:00 00 00 00 00 00 00 0F 64
 3AE8:C0 00 15 50 00 3F F0 00 44
 3AF0:2B F0 00 3B F0 00 AB B0 7A
 3AF8:00 2A B0 00 2A A0 00 02 E3
 3B00:00 00 00 00 00 00 00 00 76
 3B08:00 00 00 01 00 00 07 00 9C
 3B10:28 0F 00 2A AD 00 2A A4 67
 3B18:00 00 F0 00 00 AC 00 02 61
 3B20:AB 00 0A AA 00 0A AA 00 D5
 3B28:2A FA 00 AB 02 05 AB 00 A8
 3B30:15 AC 0F 3F B0 FF 3F FF 36
 3B38:FF 00 00 00 00 00 00 02 B0
 3B40:02 A0 00 02 90 00 5E 70 B1
 3B48:00 FD F4 00 FF FD 00 5F 34
 3B50:FF 40 0F FF C0 0F F7 1C 07
 3B58:0F F1 1C 0F FE B0 F5 7A 68
 3B60:B4 AA AA AC AA AB AC AA 04
 3B68:AF AC AA A3 AC 00 03 AE 8B
 3B70:FF F3 AE FF FF 5E FF FF 33
 3B78:FF 00 00 00 00 00 00 02 F0
 3B80:00 00 00 00 00 00 00 00 F6
 3B88:00 00 00 00 00 00 00 00 FE
 3B90:00 00 00 00 00 00 00 00 07
 3B98:00 00 00 00 00 00 00 00 0F
 3BA0:00 00 00 00 00 00 00 00 17
 3BA8:00 00 00 00 00 00 00 00 1F
 3BB0:00 00 00 00 00 00 00 00 27
 3BB8:00 00 00 00 00 00 00 0B 3A
 3BC0:00 00 00 00 00 00 00 00 37
 3BC8:00 00 00 00 00 00 00 00 3F
 3BD0:00 00 00 00 00 00 00 00 47
 3BD8:00 00 00 00 00 00 00 00 4F
 3BE0:00 00 00 00 00 00 00 00 57
 3BE8:00 00 00 00 00 00 00 00 5F
 3BF0:00 00 00 00 00 00 00 00 67
 3BF8:00 00 00 00 00 00 00 00 6F

Chapter 2

3C00:00 00 00 00 00 00 00 00 78
3C08:00 00 00 00 00 00 00 00 80
3C10:00 00 00 00 00 00 2C 00 E0
3C18:00 AB 00 02 AA C0 02 AA A2
3C20:BF 02 BA A9 02 8E A9 02 8A
3C28:83 E9 02 FE A9 4F AA A5 92
3C30:7A AA 9A 5A AF EA 5A BA 22
3C38:A5 FF FA AF 00 0F FC 00 14
3C40:00 00 00 00 00 00 00 00 B8
3C48:00 00 00 00 00 00 00 00 C0
3C50:00 00 00 00 00 00 00 00 C8
3C58:00 00 00 00 02 00 00 0A EA
3C60:B4 FF CA B7 FF 6A B7 FD 1F
3C68:AB F7 5F 6A F7 FF DB F7 B6
3C70:FF FF FF FF C0 00 FF 00 EE
3C78:00 00 00 00 00 00 00 00 F0
3C80:EA EA 20 A0 40 20 93 40 21
3C88:A9 00 8D 3C 03 8D 3D 03 17
3C90:8D 3E 03 8D 3F 03 A9 FF F1
3C98:8D 15 D0 8D 1C D0 A9 01 88
3CA0:8D 25 D0 A9 00 8D 26 D0 31
3CA8:A2 00 A9 0A 9D 27 D0 E8 5C
3CB0:E0 08 D0 F6 A9 C0 8D F8 89
3CB8:07 A9 C1 8D F9 07 A9 C2 32
3CC0:8D FA 07 A9 C3 8D FB 07 8D
3CC8:A9 D9 8D FC 07 A9 DA 8D 30
3CD0:FD 07 A9 DB 8D FE 07 A9 1D
3CD8:DC 8D FF 07 A9 B4 8D 05 D3
3CE0:D0 8D 07 D0 8D 0D D0 8D E2
3CE8:0F D0 A9 A0 8D 01 D0 8D FB
3CF0:03 D0 8D 09 D0 8D 0B D0 05
3CF8:A9 51 8D 00 D0 8D 04 D0 E1
3D00:A9 69 8D 02 D0 8D 06 D0 15
3D08:A9 B4 8D 08 D0 8D 0C D0 5C
3D10:A9 CC 8D 0A D0 8D 0E D0 8E
3D18:AD 00 DC 29 10 D0 35 A5 6B
3D20:FC D0 31 AD 00 DC 29 0F 23
3D28:C9 0A D0 08 A9 E2 20 CE 8C
3D30:3E 4C BF 3D C9 09 D0 08 C4
3D38:A9 EA 20 CE 3E 4C BF 3D 13
3D40:C9 0B D0 08 A9 DE 20 CE D4
3D48:3E 4C BF 3D A9 E6 20 CE B8
3D50:3E 4C BF 3D AD 00 DC 29 19
3D58:10 F0 04 A9 00 85 FC AD EF
3D60:00 DC 29 0F C9 0F D0 03 57
3D68:4C BF 3D 38 C9 08 90 30 E3
3D70:AD 0E D0 18 C9 28 B0 03 34
3D78:4C BF 3D AD 08 D0 38 ED 6D
3D80:00 D0 C9 14 B0 03 4C BF 93
3D88:3D A2 05 CE 08 D0 CE 0A 03
3D90:D0 CE 0C D0 CE 0E D0 CA D0

3D98:D0 F1 EE FF 07 4C BF 3D FB
 3DA0:AD 0E D0 38 C9 EB 90 03 35
 3DAB:4C BF 3D A2 05 EE 08 D0 CF
 3DB0:EE 0A D0 EE 0C D0 EE 0E BD
 3DB8:D0 CA D0 F1 EE FF 07 AD BA
 3DC0:01 DC 29 10 D0 35 A5 FB BB
 3DC8:D0 31 AD 01 DC 29 0F C9 31
 3DD0:06 D0 08 A9 C9 20 AE 3E 88
 3DD8:4C 6C 3E C9 07 D0 08 A9 2E
 3DE0:C5 20 AE 3E 4C 6C 3E C9 5A
 3DE8:05 D0 08 A9 D1 20 AE 3E 60
 3DF0:4C 6C 3E A9 CD 20 AE 3E 99
 3DF8:4C 6C 3E AD 01 DC 29 10 35
 3E00:F0 04 A9 00 85 FB AD 01 A3
 3E08:DC 29 0F C9 0F D0 03 4C C9
 3E10:6C 3E 38 C9 08 90 25 AD 70
 3E18:00 D0 18 C9 2D B0 03 4C E6
 3E20:6C 3E A2 05 CE 00 D0 CE ED
 3E28:02 D0 CE 04 D0 CE 06 D0 92
 3E30:CA D0 F1 EE FA 07 EE FA 40
 3E38:07 4C 6C 3E AD 08 D0 38 24
 3E40:ED 00 D0 C9 14 B0 03 4C 20
 3E48:6C 3E AD 00 D0 38 C9 DC 18
 3E50:90 03 4C 6C 3E A2 05 EE 9B
 3E58:00 D0 EE 02 D0 EE 04 D0 22
 3E60:EE 06 D0 CA D0 F1 EE FA C3
 3E68:07 EE FA 07 A2 64 20 BA 95
 3E70:3F AD 1E D0 AA 29 12 C9 B0
 3E78:12 F0 73 8A 29 48 C9 48 97
 3E80:F0 6C 8A 29 18 C9 18 F0 7D
 3E88:65 8A 29 42 C9 42 F0 5E 3B
 3E90:A5 FB A8 A9 C0 20 AE 3E B0
 3E98:98 85 FB A5 FC A8 A9 D9 54
 3EA0:20 CE 3E 98 85 FC A2 32 C9
 3EA8:20 BA 3F 4C 18 3D A2 00 8B
 3EB0:9D F8 07 18 69 01 E8 E0 9E
 3EB8:04 D0 F5 A9 01 85 FB AD 88
 3EC0:F8 07 C9 C0 F0 07 C9 D5 CD
 3EC8:F0 03 20 6A 40 60 A2 00 F1
 3ED0:9D FC 07 18 69 01 E8 E0 BF
 3ED8:04 D0 F5 A9 01 85 FC AD AA
 3EE0:FC 07 C9 EE F0 E7 C9 D9 5A
 3EE8:F0 E3 20 6A 40 60 AD F8 59
 3EF0:07 C9 CD F0 70 AD FC 07 67
 3EF8:C9 E6 F0 69 AD F8 07 C9 F1
 3F00:C0 D0 51 AD FC 07 C9 D9 89
 3F08:F0 4A 20 44 40 A2 0A CE 49
 3F10:00 D0 CE 02 D0 CE 04 D0 57
 3F18:CE 06 D0 CA D0 F1 EE FA 6D
 3F20:07 EE FA 07 A9 D5 20 AE 41
 3F28:3E A9 D9 20 CE 3E 20 10 2D

Chapter 2

3F30:40 A2 FF 20 BA 3F A2 FF 91
3F38:20 BA 3F A2 FF 20 BA 3F BC
3F40:A9 03 CD 3F 03 F0 03 4C 30
3F48:B4 3C A9 E8 CD 3E 03 D0 32
3F50:F6 4C E5 3F AD FC 07 C9 47
3F58:D9 D0 0D AD F8 07 C9 C0 AC
3F60:D0 06 4C 90 3E 4C 90 3E DD
3F68:AD FC 07 C9 D9 D0 F6 AD 28
3F70:F8 07 C9 C0 F0 EF 20 44 3E
3F78:40 A2 0A EE 08 D0 EE 0A 5B
3F80:D0 EE 0C D0 EE 0E D0 CA CD
3F88:D0 F1 A9 EE 20 CE 3E A9 72
3F90:C0 20 AE 3E 20 FC 3F A2 47
3F98:FF 20 BA 3F A2 FF 20 BA 7A
3FA0:3F A2 FF 20 BA 3F A9 03 92
3FA8:CD 3D 03 F0 03 4C B4 3C BB
3FB0:A9 E8 CD 3C 03 D0 F6 4C 51
3FB8:C3 3F A0 FF 88 D0 FD CA 4B
3FC0:D0 F8 60 A2 05 A0 0B 18 F4
3FC8:20 F0 FF A0 00 B9 9F 45 09
3FD0:C9 00 F0 06 20 D2 FF C8 C7
3FD8:D0 F3 A9 00 85 C6 A5 C6 4B
3FE0:F0 FC 4C 82 3C A2 05 A0 DF
3FE8:0B 18 20 F0 FF A0 00 B9 42
3FF0:AC 45 C9 00 F0 E4 20 D2 7E
3FF8:FF C8 D0 F3 18 AD 3C 03 F5
4000:69 64 8D 3C 03 AD 3D 03 10
4008:69 00 8D 3D 03 4C 21 40 8E
4010:18 AD 3E 03 69 64 8D 3E 36
4018:03 AD 3F 03 69 00 8D 3F 43
4020:03 A2 08 A0 22 18 20 F0 78
4028:FF AD 3D 03 AE 3C 03 20 78
4030:CD BD A2 0D A0 22 18 20 0A
4038:F0 FF AD 3F 03 AE 3E 03 2D
4040:20 CD BD 60 A9 0F 8D 18 BE
4048:D4 8D 0C D4 A9 07 8D 0D F6
4050:D4 A9 1C 8D 08 D4 A9 00 E8
4058:8D 07 D4 A9 81 8D 0B D4 C3
4060:A2 0A 20 BA 3F A9 80 8D 93
4068:0B D4 A9 E1 8D 00 D4 A9 B6
4070:1E 8D 01 D4 A9 44 8D 05 4F
4078:D4 A9 00 8D 06 D4 A9 07 84
4080:8D 18 D4 A9 81 8D 04 D4 22
4088:A2 0A 20 BA 3F A9 80 8D BB
4090:04 D4 60 A2 00 A9 00 9D C2
4098:00 D4 E8 E0 18 D0 F8 60 CF
40A0:A9 00 8D 21 D0 A9 0F 8D 92
40A8:20 D0 A0 00 B9 DC 40 20 63
40B0:D2 FF C8 D0 F7 B9 DC 41 62
40B8:20 D2 FF C8 D0 F7 B9 DC 41
40C0:42 20 D2 FF C8 D0 F7 B9 F7

Type It In

40C8:DC 43 20 D2 FF C8 D0 F7 76
40D0:B9 DC 44 F0 06 20 D2 FF 53
40D8:C8 D0 F5 60 93 20 12 05 FC
40E0:20 20 20 20 20 20 20 20 61
40E8:20 20 20 20 20 20 20 20 69
40F0:20 20 20 20 20 20 20 20 71
40F8:20 20 20 20 20 20 20 20 79
4100:20 20 20 20 20 20 92 20 67
4108:12 20 92 20 20 20 20 20 D1
4110:20 20 20 20 12 97 20 20 FF
4118:20 20 20 20 20 20 20 20 9A
4120:20 92 20 20 20 20 20 20 3F
4128:20 20 20 20 12 05 20 92 40
4130:20 20 20 20 20 20 20 12 A4
4138:20 20 92 20 20 20 20 20 09
4140:20 20 20 20 12 97 20 20 30
4148:20 20 20 20 20 20 20 20 CA
4150:20 92 20 20 20 20 20 20 6F
4158:20 20 20 20 12 05 20 92 70
4160:20 20 20 20 20 20 20 12 D4
4168:20 20 92 20 20 20 20 20 39
4170:20 20 20 20 DF 12 20 20 B8
4178:20 20 20 20 20 20 20 92 6D
4180:A9 20 20 20 20 20 20 20 C7
4188:20 20 20 12 05 20 92 20 36
4190:20 20 20 20 20 20 12 20 F6
4198:20 92 20 20 20 20 20 20 B7
41A0:20 20 20 20 20 20 20 20 23
41A8:20 20 20 20 20 20 20 20 2B
41B0:20 20 20 20 20 20 20 A0 B3
41B8:12 05 20 92 20 20 20 20 94
41C0:20 20 20 12 20 20 92 20 47
41C8:20 20 20 20 20 20 20 20 4B
41D0:20 20 20 20 20 20 20 20 53
41D8:20 20 20 20 20 20 20 20 5B
41E0:20 20 20 20 20 12 05 20 F4
41E8:92 20 12 57 48 49 54 45 C9
41F0:92 20 12 20 20 92 20 20 B4
41F8:20 20 20 20 20 20 20 20 7B
4200:20 20 20 20 20 20 20 20 84
4208:20 20 20 20 20 20 20 20 8C
4210:20 20 20 20 12 20 92 20 09
4218:12 53 43 4F 52 45 92 20 C4
4220:12 20 20 92 20 20 20 20 C4
4228:20 20 20 20 20 20 20 20 AC
4230:20 20 20 20 20 20 20 20 B4
4238:20 20 20 20 20 20 20 20 BC
4240:20 20 12 20 92 20 20 20 96
4248:20 20 20 20 12 20 20 92 CE
4250:20 96 D1 20 D1 20 D1 20 99
4258:D1 20 D1 20 D1 20 D1 20 DC

Chapter 2

4260:D1 20 D1 20 D1 20 D1 20 E4
4268:D1 20 D1 20 D1 20 D1 12 DE
4270:05 20 92 20 20 30 30 30 26
4278:20 20 12 20 20 92 96 D1 A3
4280:20 D1 20 D1 20 D1 20 D1 05
4288:20 D1 20 D1 20 D1 20 D1 0D
4290:20 D1 20 D1 20 D1 20 D1 15
4298:20 D1 20 D1 20 12 05 20 36
42A0:92 20 20 20 20 20 20 20 5E
42A8:12 20 20 92 20 96 D1 20 8A
42B0:D1 20 D1 20 D1 20 D1 20 35
42B8:D1 20 D1 20 D1 20 D1 20 3D
42C0:D1 20 D1 20 D1 20 D1 20 45
42C8:D1 20 D1 12 05 20 92 20 87
42D0:12 9B 42 4C 41 43 4B 92 92
42D8:20 12 05 20 20 92 96 D1 DE
42E0:20 12 95 A9 20 92 A9 9E 84
42E8:B7 B7 B7 B7 B7 B7 B7 B7 6D
42F0:B7 B7 B7 B7 B7 B7 B7 B7 75
42F8:B7 B7 B7 B7 95 DF 12 20 2A
4300:DF 92 96 D1 20 12 05 20 7E
4308:92 20 12 9B 53 43 4F 52 74
4310:45 92 20 12 05 20 20 92 7E
4318:20 9E CE 96 D1 12 95 20 BB
4320:92 96 20 D1 20 D1 20 D1 11
4328:20 D1 20 D1 20 D1 20 D1 AE
4330:20 D1 20 D1 20 D1 20 D1 B6
4338:95 12 20 92 96 D1 9E CD 42
4340:96 D1 12 05 20 92 20 20 C4
4348:20 20 20 20 20 12 20 20 96
4350:92 9E CE 96 D1 9E CE 12 C3
4358:95 20 92 9E B7 B7 B7 B7 B1
4360:B7 B7 B7 B7 B7 B7 B7 B7 E6
4368:B7 B7 B7 B7 B7 B7 B7 B7 EE
4370:B7 B7 95 12 20 92 9E CD EA
4378:96 D1 9E CD 12 05 20 92 E6
4380:20 20 30 30 30 20 20 12 7C
4388:20 20 92 96 D1 9E CE 1C A5
4390:D1 12 95 20 92 1C D1 20 02
4398:D1 20 D1 20 D1 20 D1 20 1F
43A0:D1 20 D1 20 D1 20 D1 20 27
43A8:D1 20 D1 20 95 12 20 92 24
43B0:1C D1 9E CD 96 D1 12 05 8F
43B8:20 92 20 20 20 20 20 DB
43C0:20 12 20 20 92 9E CE 1C AA
43C8:D1 12 9B A9 20 20 20 20 AC
43D0:20 20 20 20 20 20 20 57
43D8:20 20 20 20 20 20 20 5F
43E0:20 20 20 20 DF 92 1C D1 D8
43E8:9E CD 12 05 20 92 20 20 70
43F0:20 20 20 20 20 12 20 20 3F

Type It In

43F8:92 1C D1 12 9B A9 20 20 0F
4400:20 20 20 20 20 20 20 20 88
4408:20 20 20 20 20 20 20 20 90
4410:20 20 20 20 20 20 20 20 98
4418:DF 92 1C D1 12 05 20 92 4D
4420:20 20 20 20 20 20 20 12 9A
4428:20 20 92 12 9B A9 20 20 20
4430:20 20 20 20 20 20 20 20 B8
4438:20 20 20 20 20 20 20 20 C0
4440:20 20 20 20 20 20 20 20 C8
4448:20 20 DF 92 12 05 20 92 85
4450:20 20 20 20 20 20 20 12 CA
4458:20 20 9B 20 20 20 20 20 50
4460:20 20 20 20 20 20 20 20 E8
4468:20 20 20 20 20 20 20 20 F0
4470:20 20 20 20 20 20 20 20 F8
4478:20 05 20 92 20 20 20 20 61
4480:20 20 20 12 20 20 9B 20 1F
4488:20 20 20 20 20 20 20 20 11
4490:20 20 20 20 20 20 20 20 19
4498:20 20 20 20 20 20 20 20 21
44A0:20 20 20 20 20 05 20 92 2F
44A8:20 20 20 20 20 20 20 12 23
44B0:20 20 98 20 20 20 20 20 48
44B8:20 20 20 20 20 20 20 20 41
44C0:20 20 20 20 20 20 20 20 49
44C8:20 20 20 20 20 20 20 20 51
44D0:20 05 20 92 20 20 20 20 B9
44D8:20 20 20 12 20 20 92 20 65
44E0:97 DF 12 20 92 A9 20 20 0D
44E8:20 20 20 20 20 20 20 20 71
44F0:20 20 20 20 20 20 20 20 79
44F8:20 20 20 20 DF 12 20 92 B9
4500:A9 20 12 05 20 92 20 20 A5
4508:20 20 20 20 20 12 20 20 5A
4510:92 20 97 20 12 20 20 20 52
4518:20 20 52 49 4E 47 53 49 19
4520:44 45 20 20 4B 41 52 41 69
4528:54 45 20 20 20 20 20 92 88
4530:20 20 12 05 20 92 20 20 11
4538:20 20 20 20 20 12 20 20 8A
4540:92 20 97 20 12 20 92 20 67
4548:20 20 20 20 20 20 20 20 D2
4550:20 20 20 20 20 20 20 20 DA
4558:20 20 20 20 20 20 20 12 D4
4560:20 92 20 20 12 05 20 92 1D
4568:20 20 20 20 20 20 20 12 E4
4570:20 92 20 12 05 20 20 20 DD
4578:20 20 20 20 20 20 20 20 03
4580:20 20 20 20 20 20 20 20 0B
4588:20 20 20 20 20 20 20 20 13

Chapter 2

4590:20 20 20 20 20 20 20 20 1B
4598:20 20 20 91 91 92 00 9C CB
45A0:57 48 49 54 45 20 57 49 F9
45A8:4E 53 21 00 9C 42 4C 41 1B
45B0:43 4B 20 57 49 4E 53 21 74
45B8:00 00 00 00 00 00 00 00 43

Program 7

Eagles and Gators

You need a copy of "MLX" on tape or disk before you can type in "Eagles and Gators." When you load and run MLX, the prompts will ask you for a starting address and an ending address. For the Eagles and Gators data, they are as follows:

Starting Address: 0801
Ending Address: 1CE8

Follow the MLX instructions carefully, and be sure to save a copy of the data to disk or tape before you leave MLX.

Eagles and Gators

For mistake-proof program entry, use "MLX: Machine Language Entry Program" to type in this program.

```
0801:18 08 00 00 9E 32 30 37 74
0809:36 3A 8F 22 14 14 14 14 04
0811:45 41 47 4C 45 53 00 00 39
0819:00 00 00 A0 00 A9 56 85 0C
0821:03 A9 08 85 04 A9 00 85 C2
0829:05 A9 40 85 06 A9 E3 85 AA
0831:FB A9 1C 85 FC A5 03 E5 EF
0839:FB A5 04 E5 FC B0 13 B1 12
0841:03 91 05 E6 03 D0 02 E6 8C
0849:04 E6 05 D0 02 E6 06 4C C6
0851:36 08 4C 00 40 A9 3B 85 AC
0859:FB A9 03 85 FC A0 BF A9 1E
0861:00 91 FB 88 D0 F9 A9 FF 9F
0869:85 FB A9 D3 85 FC A0 19 28
0871:A9 00 91 FB 88 D0 F9 A9 6D
0879:93 20 D2 FF A9 00 8D 20 3E
0881:D0 8D 21 D0 18 A0 0B A2 8A
0889:05 20 F0 FF A9 00 85 FB 96
0891:A9 45 85 FC 20 AD 42 18 9C
0899:A0 0C A2 08 20 F0 FF A9 40
08A1:14 85 FB A9 45 85 FC 20 91
08A9:AD 42 20 1E 43 18 A0 0A CC
08B1:A2 0E 20 F0 FF A9 26 85 22
08B9:FB A9 45 85 FC 20 AD 42 39
08C1:20 CD 42 18 A0 0D A2 11 AE
08C9:20 F0 FF A9 3B 85 FB A9 52
08D1:45 85 FC A0 00 20 BD 42 CD
```

Chapter 2

08D9:AD 04 DC 8D 86 02 B1 FB D1
08E1:C9 7E F0 07 20 D2 FF C8 1A
08E9:4C 80 40 20 9F FF 20 E4 6C
08F1:FF C9 00 F0 CE C9 59 F0 C4
08F9:78 C9 4E F0 03 4C 6E 40 F7
0901:A9 93 20 D2 FF 20 74 42 A9
0909:20 6D 41 18 A0 11 A2 09 C7
0911:20 F0 FF A9 4C 85 FB A9 24
0919:45 85 FC 20 AD 42 18 A0 18
0921:14 A2 0B 20 F0 FF A9 5E 82
0929:85 FB A9 45 85 FC 20 AD 94
0931:42 18 A0 11 A2 0F 20 F0 12
0939:FF A9 68 85 FB A9 45 85 B1
0941:FC 20 AD 42 20 9F FF 20 53
0949:E4 FF C9 00 F0 F6 C9 51 4F
0951:D0 03 4C E2 FC C9 31 B0 66
0959:03 4C EF 40 C9 3A B0 E4 7F
0961:38 E9 30 8D 3F 03 A9 93 D5
0969:20 D2 FF 20 EA 4B 4C 0F 6E
0971:54 A9 93 20 D2 FF 20 74 D7
0979:42 18 A0 04 A2 00 20 F0 4D
0981:FF A9 7C 85 FB A9 45 85 7C
0989:FC A9 05 85 FD A9 47 85 28
0991:FE A0 00 B1 FB 20 D2 FF 6C
0999:A5 FB C9 FF 90 02 E6 FC 0E
09A1:E6 FB A5 FB C5 FD A5 FC 09
09A9:E5 FE 90 E7 20 EC 50 AD 02
09B1:67 03 C9 00 F0 09 AD 6A E2
09B9:03 C9 00 F0 02 D0 ED 4C 4A
09C1:AB 40 A9 40 85 FB A9 03 65
09C9:85 FC A9 E4 85 FD A0 08 CE
09D1:A5 FD 91 FB E6 FD E6 FD 23
09D9:88 C0 01 D0 F3 A9 EA 91 3B
09E1:FB A0 08 A9 48 85 FB A9 AF
09E9:0D 91 FB 88 D0 F9 A9 50 01
09F1:85 FB A9 F2 85 FD A0 08 97
09F9:A5 FD 91 FB E6 FD E6 FD 4B
0A01:88 C0 01 D0 F3 A9 F2 91 74
0A09:FB A9 59 85 FB A0 07 A9 23
0A11:00 91 FB 88 D0 F9 A9 5B AE
0A19:8D 6C 03 A9 4E 8D 6E 03 92
0A21:A9 03 8D 6D 03 8D 6F 03 83
0A29:A9 71 85 FB A9 32 85 FD FD
0A31:A0 08 A5 FD 91 FB 18 69 42
0A39:0A 85 FD 88 D0 F4 A9 79 23
0A41:85 FB A9 E6 85 FD A0 08 28
0A49:A5 FD 91 FB 38 E9 0A 85 A4
0A51:FD 88 D0 F4 A9 81 85 FB 4A
0A59:A0 10 A9 FF 91 FB 88 A9 2E
0A61:01 91 FB 88 D0 F4 A9 82 92

Type It In

0A69:8D 92 03 A9 8A 8D 94 03 9A
0A71:A9 72 8D 96 03 A9 7A 8D 53
0A79:98 03 A9 03 8D 93 03 8D 4E
0A81:95 03 8D 97 03 8D 99 03 D0
0A89:A9 1E 8D 00 D0 A9 46 8D F2
0A91:02 D0 A9 6E 8D 04 D0 A9 BE
0A99:96 8D 06 D0 A9 BE 8D 08 95
0AA1:D0 A9 E6 8D 0A D0 A9 0F 34
0AA9:8D 0C D0 A9 37 8D 0E D0 19
0AB1:A9 C0 8D 10 D0 A9 00 8D 38
0AB9:17 D0 8D 1D D0 A9 01 8D CD
0AC1:18 D4 EA A9 FF 8D 15 D0 40
0AC9:60 78 A9 00 8D 0E DC A9 69
0AD1:8C 8D 14 03 A9 4A 8D 15 E8
0AD9:03 A9 0F 8D 88 4A A9 96 EB
0AE1:8D 89 4A A9 00 8D 8A 4A 98
0AE9:A9 03 8D 8B 4A AD 11 D0 F9
0AF1:29 7F 8D 11 D0 AD 89 4A D7
0AF9:8D 12 D0 A9 01 8D 1A D0 51
0B01:58 60 A0 00 B1 FB C9 7E FE
0B09:F0 07 20 D2 FF C8 4C AF F5
0B11:42 60 EE 01 D0 CE 03 D0 E6
0B19:CE 0D D0 EE 0F D0 20 CA A9
0B21:44 60 A9 1E 8D 00 D0 A9 40
0B29:46 8D 02 D0 A9 C0 8D 10 8E
0B31:D0 A9 0F 8D 0C D0 A9 37 03
0B39:8D 0E D0 A9 07 8D 27 D0 DB
0B41:8D 2E D0 A9 0D 8D 28 D0 1E
0B49:8D 2D D0 A9 C3 8D 15 D0 75
0B51:A9 FF 8D F8 07 8D FF 07 F2
0B59:A9 E4 8D F9 07 A9 EA 8D 11
0B61:FE 07 60 EA EA A0 00 B1 FE
0B69:FB 91 FD C8 C0 40 F0 02 19
0B71:D0 F5 60 A9 08 85 FB A9 0C
0B79:47 85 FC A9 C0 85 FD A9 90
0B81:3A 85 FE 20 10 43 A9 80 59
0B89:85 FD 20 10 43 A9 48 85 BD
0B91:FB A9 00 85 FD A9 3B 85 FA
0B99:FE 20 10 43 A9 40 85 FD C4
0BA1:A9 3A 85 FE 20 10 43 A9 2D
0BA9:88 85 FB A9 00 85 FD 20 B1
0BB1:10 43 A9 40 85 FD A9 3B 8C
0BB9:85 FE 20 10 43 A9 C8 85 2F
0BC1:FB A9 80 85 FD 20 10 43 7C
0BC9:A9 C0 85 FD A9 39 85 FE B1
0BD1:20 10 43 A9 08 85 FB A9 F6
0BD9:48 85 FC A9 80 85 FD 20 E5
0BE1:10 43 A9 C0 85 FD A9 3B C4
0BE9:85 FE 20 10 43 A9 48 85 5E
0BF1:FB A9 40 85 FD A9 39 85 5F

Chapter 2

ØBF9:FE 2Ø 1Ø 43 A9 ØØ 85 FD 24
ØCØ1:A9 3C 85 FE 2Ø 1Ø 43 A9 ØF
ØCØ9:88 85 FB A9 4Ø 85 FD 2Ø 15
ØC11:1Ø 43 A9 ØØ 85 FD A9 39 E7
ØC19:85 FE 2Ø 1Ø 43 A9 C8 85 9Ø
ØC21:FB A9 48 85 FC A9 4Ø 85 97
ØC29:FD A9 3E 85 FE 2Ø 1Ø 43 A6
ØC31:A9 ØØ 85 FD 2Ø 1Ø 43 A9 2Ø
ØC39:Ø8 85 FB A9 49 85 FC A9 D4
ØC41:8Ø 85 FD 2Ø 1Ø 43 A9 CØ 5E
ØC49:85 FD A9 3D 85 FE 2Ø 1Ø 25
ØC51:43 A9 48 85 FB A9 8Ø 85 E3
ØC59:FD 2Ø 1Ø 43 A9 CØ 85 FD Ø8
ØC61:A9 3E 85 FE 2Ø 1Ø 43 A9 EF
ØC69:88 85 FB A9 ØØ 85 FD A9 FC
ØC71:3F 85 FE 2Ø 1Ø 43 A9 4Ø 8D
ØC79:85 FD A9 3D 85 FE 2Ø 1Ø 55
ØC81:43 A9 C8 85 FB A9 ØØ 85 23
ØC89:FD 2Ø 1Ø 43 A9 4Ø 85 FD 36
ØC91:A9 3F 85 FE 2Ø 1Ø 43 A9 6Ø
ØC99:Ø8 85 FB A9 4A 85 FC A9 3D
ØCA1:CØ 85 FD A9 3C 85 FE 2Ø EB
ØCA9:1Ø 43 A9 8Ø 85 FD A9 3F 8E
ØCB1:85 FE 2Ø 1Ø 43 A9 48 85 28
ØCB9:FB A9 CØ 85 FD 2Ø 1Ø 43 7E
ØCC1:A9 8Ø 85 FD A9 3C 85 FE A7
ØCC9:2Ø 1Ø 43 A9 ØF 8D Ø5 D4 86
ØCD1:8D ØC D4 8D 13 D4 A9 FØ 57
ØCD9:8D Ø6 D4 8D ØD D4 8D 14 98
ØCE1:D4 A9 3F 8D ØØ D4 A9 18 4E
ØCE9:8D Ø1 D4 A9 42 8D Ø7 D4 69
ØCF1:A9 Ø6 8D Ø8 D4 A9 ØC 8D 85
ØCF9:ØE D4 A9 Ø6 8D ØF D4 A9 DF
ØDØ1:15 8D Ø4 D4 8D ØB D4 8D A6
ØDØ9:12 D4 A9 3F 8D 18 D4 A9 AA
ØD11:F7 8D 17 D4 A9 22 8D 16 C1
ØD19:D4 A9 ØF 8D 15 D4 6Ø AD 2D
ØD21:3E Ø3 C9 2F FØ Ø4 EE 3E FA
ØD29:Ø3 6Ø A9 ØØ 8D 3E Ø3 EE 6C
ØD31:3F Ø3 AD 3F Ø3 8D ØE D4 94
ØD39:CE 4Ø Ø3 AD 4Ø Ø3 8D 16 45
ØD41:D4 AD 3F Ø3 C9 35 9Ø Ø6 93
ØD49:A9 18 8D Ø1 D4 6Ø AD 3F C2
ØD51:Ø3 8D Ø1 D4 6Ø 8E Ø8 9F AA
ØD59:ØØ 2Ø 45 41 47 4C 45 53 81
ØD61:2Ø 26 2Ø 47 41 54 4F 52 D9
ØD69:53 2Ø 92 7E ØØ ØØ ØØ ØØ 6F
ØD71:ØØ ØØ ØØ ØØ ØØ ØØ ØØ 8B
ØD79:ØØ ØØ ØØ 9F 57 41 4E 54 3E
ØD81:2Ø 49 4E 53 54 52 55 43 D6

Type It In

ØD89:54 49 4F 4E 53 2Ø 3F 7E Ø7
ØD91:5Ø 52 45 53 53 2Ø 12 59 DE
ØD99:92 2Ø 4F 52 2Ø 12 4E 92 8C
ØDA1:7E 9Ø 5Ø 49 43 4B 2Ø 53 98
ØDA9:4B 49 4C 4C 2Ø 4C 45 56 1D
ØDB1:45 4C 7E 28 2Ø 31 2Ø 63 3D
ØDB9:2Ø 39 2Ø 29 7E 9E 5Ø 52 2A
ØDC1:45 53 53 2Ø 12 51 92 2Ø DA
ØDC9:2Ø 54 4F 2Ø 51 55 49 54 BB
ØDD1:7E 9Ø 12 77 11 9D 2Ø 11 59
ØDD9:9D 9D ØØ ØØ 2Ø 2Ø 2Ø 92 7E
ØDE1:63 69 11 9D 7B 91 91 5Ø 99
ØDE9:4C 55 47 2Ø 4A 4F 59 53 FF
ØDF1:54 49 43 4B 53 2Ø 49 4E A1
ØDF9:2E 2E 11 11 7B 91 9D 75 BC
ØEØ1:63 63 63 63 12 2Ø 2Ø 2Ø BB
ØEØ9:91 9D 9D 2Ø 91 9D 69 8D 6E
ØE11:11 11 11 ØØ 2Ø 12 5Ø 4F 55
ØE19:52 54 2Ø 32 2Ø 46 4F 52 A5
ØE21:2Ø 45 41 47 4C 45 53 92 EB
ØE29:2Ø 2Ø 2Ø 12 5Ø 4F 52 54 3B
ØE31:2Ø 31 2Ø 46 4F 52 2Ø 47 5D
ØE39:41 54 4F 52 53 92 8D 11 2B
ØE41:ØØ 2Ø 43 4C 41 4D 5Ø 2Ø 92
ØE49:59 4F 55 52 2Ø 43 52 45 AD
ØE51:41 54 55 52 45 53 2Ø 54 FE
ØE59:4F 2Ø 54 48 45 2Ø 42 4F B2
ØE61:52 44 45 52 2Ø 42 59 8D CF
ØE69:11 2Ø 5Ø 52 45 53 53 49 AC
ØE71:4E 47 2Ø 12 46 49 52 45 EC
ØE79:2Ø 42 55 54 54 4F 4E 92 35
ØE81:2Ø 57 48 45 4E 2Ø 54 48 C4
ØE89:45 59 2Ø 48 49 54 2Ø 49 4C
ØE91:54 69 11 9D 62 11 9D 9D 5E
ØE99:6D 62 6E 11 9D 9D 2A 11 AC
ØEA1:9D 7A 8D 91 91 2Ø 42 4F D6
ØEA9:52 44 45 52 2Ø 4D 4F 56 F8
ØEB1:45 53 2Ø 57 48 45 4E 2Ø D2
ØEB9:53 49 44 45 53 2Ø 41 52 9E
ØEC1:45 2Ø 55 4E 45 51 55 41 73
ØEC9:4C 8D 9F 11 11 2Ø 2Ø 2Ø DD
ØED1:4D 4F 56 45 2Ø 52 49 47 AB
ØED9:48 54 2Ø 4F 52 2Ø 4C 45 19
ØEE1:46 54 2Ø 54 4F 2Ø 43 4F 5Ø
ØEE9:4E 54 52 4F 4C 2Ø 41 8D 74
ØEF1:11 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 86
ØEF9:2Ø 2Ø 44 49 46 46 45 52 73
ØFØ1:45 4E 54 2Ø 43 52 45 41 11
ØFØ9:54 55 52 45 8D 11 11 2Ø 38
ØF11:2Ø Ø5 57 49 4E 2Ø 42 59 DØ

Chapter 2

0F19:20 50 55 53 48 49 4E 47 86
0F21:20 4F 50 50 4F 4E 45 4E BE
0F29:54 53 20 4F 46 46 20 53 1E
0F31:43 52 45 45 4E 8D 11 11 5E
0F39:9E 20 20 20 20 20 12 50 AA
0F41:52 45 53 53 20 46 49 52 78
0F49:45 20 42 55 54 54 4F 4E 90
0F51:20 54 4F 20 42 45 47 49 7F
0F59:4E 92 00 00 00 00 03 80 C9
0F61:00 0F C0 02 1F 80 06 37 B9
0F69:80 03 7D 80 01 FF A0 00 89
0F71:FF 30 00 7F E0 00 7F 00 99
0F79:00 7F E0 00 FF 60 01 FE 16
0F81:00 03 3E 00 06 3E 00 04 55
0F89:1E 00 00 1C 00 00 1E 00 B4
0F91:00 0E 08 00 0F 30 00 03 70
0F99:E0 00 01 80 00 00 0C 00 68
0FA1:00 1E 00 00 3E 00 04 3F 80
0FA9:00 0E 6D 20 03 FF 30 00 73
0FB1:FF 60 00 FF C0 00 7F 00 EC
0FB9:00 FF C0 00 FF 60 03 FE 76
0FC1:00 06 7C 00 00 7C 00 00 E2
0FC9:3C 00 00 3C 00 00 3C 00 42
0FD1:00 1C 00 00 1E 00 00 0F F6
0FD9:E0 00 03 80 FF 00 18 00 01
0FE1:00 3C 00 00 3C 00 04 7E 77
0FE9:20 0C 5A 30 06 FF 60 03 5D
0FF1:FF C0 00 FF 00 00 FF 00 40
0FF9:00 FF 00 03 FF C0 06 FF 57
1001:60 00 7E 00 00 7C 00 00 13
1009:7C 00 00 3C 00 00 3C 00 A3
1011:00 3C 00 00 3C 00 00 1E 40
1019:00 00 07 80 00 00 18 00 52
1021:00 3C 00 00 3C 00 04 7E B8
1029:20 0C 5A 30 06 FF 60 03 9E
1031:FF C0 00 FF 00 00 FF 00 81
1039:00 FF 00 03 FF C0 06 FF 98
1041:60 00 7E 00 00 7E 00 00 5B
1049:7E 00 00 3C 00 00 3C 00 E4
1051:00 3C 00 00 3C 00 00 18 7A
1059:00 00 18 00 F7 00 18 00 6C
1061:00 3C 00 00 3C 00 04 7E F8
1069:20 0C 5A 30 06 FF 60 03 DE
1071:FF C0 00 FF 00 00 FF 00 C1
1079:00 FF 00 03 FF C0 06 FF D8
1081:60 00 7E 00 00 3E 00 00 9A
1089:3E 00 00 3C 00 00 3C 00 05
1091:00 3C 00 00 3C 00 00 78 1B
1099:00 01 E0 00 00 00 30 00 76
10A1:00 78 00 00 7C 00 00 FC C0

Type It In

10A9:20 04 B6 70 0C FF C0 06 A0
10B1:FF 00 03 FF 00 00 FE 00 30
10B9:03 FF 00 06 FF 00 00 7F 3B
10C1:C0 00 3E 60 00 3E 00 00 09
10C9:3C 00 00 3C 00 00 3C 00 44
10D1:00 38 00 00 78 00 07 F0 C2
10D9:00 01 C0 00 FF 01 C0 00 D7
10E1:03 F0 00 01 F8 40 01 EC 87
10E9:60 01 BE C0 05 FF 80 0C 93
10F1:FF 00 07 FE 00 00 FE 00 E0
10F9:07 FE 00 06 FF 00 00 7F 3D
1101:80 00 7C C0 00 7C 60 00 B1
1109:78 20 00 38 00 00 78 00 E3
1111:10 70 00 0C F0 00 07 C0 6E
1119:00 01 80 00 00 00 00 00 8B
1121:00 00 00 00 00 00 00 00 43
1129:00 0E 00 70 1F 3C F8 7F 31
1131:FF FE 7E 7E 7E FC 99 3F 25
1139:EF 18 F7 8F 3C F1 C7 E7 72
1141:E3 07 C3 E0 03 C3 C0 07 4D
1149:E7 E0 1E FF 78 0E 3C 30 FF
1151:00 00 00 00 00 00 00 00 73
1159:00 00 00 00 FF 00 00 00 7B
1161:00 00 00 00 00 00 00 00 83
1169:00 1E 00 78 7F BD FE FF 8B
1171:FF FF DE 7E 7B 8E DB 71 96
1179:0F 18 F0 03 18 C0 07 FF 49
1181:E0 07 81 E0 07 C3 E0 03 20
1189:FF C0 07 FF E0 3E 3C 7C B1
1191:1C 00 38 00 00 00 00 00 C8
1199:00 00 00 00 FF 00 00 00 BB
11A1:00 00 00 00 00 00 30 00 24
11A9:0C FE 00 7F FF 99 FF DF CF
11B1:FF FB 1E 3C 78 0E FF 70 C6
11B9:07 18 E0 07 18 E0 0F FF 54
11C1:F0 0F 18 F0 07 81 E0 07 39
11C9:FF E0 07 FF E0 1E 7E 78 F9
11D1:3E 00 7C 20 00 04 00 00 B4
11D9:00 00 00 00 00 00 00 00 FB
11E1:00 00 00 00 00 00 F8 00 F5
11E9:1F FE 00 7F 7F 81 FE 3F 92
11F1:DB FC 1F 3C F8 0E FF 70 59
11F9:07 99 E0 0F 18 F0 0F FF B5
1201:F0 0E 7E 70 07 18 E0 03 55
1209:C3 C0 0E FF E0 0E 7E 70 AD
1211:3E 00 7C 22 00 44 00 00 17
1219:00 00 00 00 FF 00 00 00 3D
1221:C0 00 03 E0 00 07 7C 00 29
1229:3E 7F 81 FE 3F 81 FC 1F 85
1231:DB F8 06 FF 60 03 3C C0 8A

Chapter 2

1239:07 99 E0 0E 18 70 0F FF E4
1241:F0 0C FF 30 07 3C E0 03 D1
1249:81 C0 03 FF C0 0F 7E F0 EE
1251:18 00 18 3E 00 7C 32 00 BE
1259:4C 00 00 00 00 C0 00 03 A9
1261:C0 00 03 70 00 0E 7E 00 82
1269:7E 3F 81 FC 1F 81 F8 1F AC
1271:C3 F8 06 FF 60 03 7E C0 43
1279:07 99 E0 0E 18 70 0F DB 01
1281:F0 0C FF 30 06 3C 60 03 09
1289:81 C0 03 FF C0 0E 7E E0 1B
1291:1C 00 38 3F 00 FC 31 00 15
1299:8C 30 00 0C FF 1C 00 38 79
12A1:30 00 0C 38 00 1C 3E 00 CF
12A9:7C 3F 81 FC 1F C3 F8 0F E4
12B1:C3 F0 06 FF 60 03 7E C0 81
12B9:07 BD E0 0E 18 70 0B DB 42
12C1:D0 0C FF 30 06 3C 60 03 39
12C9:81 C0 03 FF C0 07 7E E0 3F
12D1:1C 00 38 3E 00 7C 33 00 47
12D9:CC 18 00 18 00 0F 96 00 55
12E1:03 AC 12 D0 A2 02 CC 88 41
12E9:4A F0 07 88 CC 88 4A F0 E6
12F1:01 CA BD 89 4A 8D 21 D0 35
12F9:8D 20 D0 BD 87 4A 8D 12 75
1301:D0 AD 19 D0 8D 19 D0 E0 7E
1309:01 D0 06 20 C8 4A 4C C2 71
1311:4A 20 59 4B 20 EA FF 68 59
1319:A8 68 AA 68 40 AD 7A 03 3A
1321:8D 01 D0 AD 7B 03 8D 03 49
1329:D0 AD 7C 03 8D 05 D0 AD B2
1331:7D 03 8D 07 D0 AD 7E 03 36
1339:8D 09 D0 AD 7F 03 8D 0B 8B
1341:D0 AD 80 03 8D 0D D0 AD 6B
1349:81 03 8D 0F D0 AD 41 03 56
1351:8D F8 07 AD 42 03 8D F9 6B
1359:07 AD 43 03 8D FA 07 AD 1B
1361:44 03 8D FB 07 AD 45 03 58
1369:8D FC 07 AD 46 03 8D FD A8
1371:07 AD 47 03 8D FE 07 AD C3
1379:48 03 8D FF 07 AD 49 03 BA
1381:8D 27 D0 AD 4A 03 8D 28 CE
1389:D0 AD 4B 03 8D 29 D0 AD 7D
1391:4C 03 8D 2A D0 AD 4D 03 CD
1399:8D 2B D0 AD 4E 03 8D 2C 0C
13A1:D0 AD 4F 03 8D 2D D0 AD 26
13A9:50 03 8D 2E D0 60 AD 72 23
13B1:03 8D 01 D0 AD 73 03 8D B8
13B9:03 D0 AD 74 03 8D 05 D0 BB
13C1:AD 75 03 8D 07 D0 AD 76 A2

Type It In

13C9:03 8D 09 D0 AD 77 03 8D E1
13D1:0B D0 AD 78 03 8D 0D D0 28
13D9:AD 79 03 8D 0F D0 AD 51 D6
13E1:03 8D F8 07 AD 52 03 8D C6
13E9:F9 07 AD 53 03 8D FA 07 05
13F1:AD 54 03 8D FB 07 AD 55 E9
13F9:03 8D FC 07 AD 56 03 8D 6F
1401:FD 07 AD 57 03 8D FE 07 68
1409:AD 58 03 8D FF 07 AD 59 28
1411:03 8D 27 D0 AD 5A 03 8D 7A
1419:28 D0 AD 5B 03 8D 29 D0 66
1421:AD 5C 03 8D 2A D0 AD 5D BD
1429:03 8D 2B D0 AD 5E 03 8D 23
1431:2C D0 AD 5F 03 8D 2D D0 C8
1439:AD 60 03 8D 2E D0 60 A0 9F
1441:00 A9 00 85 FB A9 D4 85 E1
1449:FC A9 00 91 FB C0 0F F0 65
1451:04 C8 4C F4 4B A9 0F 8D 33
1459:18 D4 A9 00 8D 17 D4 A9 14
1461:60 8D 05 D4 A9 19 8D 13 EA
1469:D4 A9 0F 8D 00 D4 8D 07 96
1471:D4 8D 0E D4 A9 0F 8D 01 1C
1479:D4 A9 4F 8D 08 D4 A9 4F 6F
1481:8D 0F D4 A9 00 8D 04 D4 7C
1489:A9 00 8D 0B D4 A9 00 8D C3
1491:12 D4 A9 00 8D 06 D4 8D E8
1499:0D D4 8D 14 D4 A9 60 8D 0C
14A1:0C D4 60 A9 60 8D 05 D4 C3
14A9:AD 89 4A 4A 4A 4A 85 FB 7B
14B1:A9 24 E5 FB 8D 01 D4 A9 F7
14B9:60 8D 05 D4 A9 20 8D 04 50
14C1:D4 A9 21 8D 04 D4 60 A9 99
14C9:60 8D 0C D4 AD 89 4A 4A C6
14D1:4A 4A 8D 08 D4 A9 80 8D BF
14D9:0B D4 A9 81 8D 0B D4 60 AC
14E1:A9 2A 8D 3C 03 A9 10 8D 4B
14E9:12 D4 A9 15 8D 12 D4 60 95
14F1:A9 04 8D 3C 03 A9 10 8D D1
14F9:12 D4 A9 15 8D 12 D4 60 A5
1501:A9 09 8D 8A 4A A5 FB 85 03
1509:FD A5 FC 85 FE 98 8D 3D 46
1511:03 A9 10 8D 12 D4 A9 15 4E
1519:8D 12 D4 A2 1F A9 00 8D 80
1521:0F D4 20 97 4D 8A 8D 3E 73
1529:03 0A 0A 0A 0A 8D 0F D4 B2
1531:A0 12 20 5B 4D AD 3E 03 8A
1539:AA CA D0 E6 AD 3D 03 A8 05
1541:A5 FD 85 FB A5 FE 85 FC 5F
1549:A9 E0 8D 0F D4 A9 85 8D 09
1551:12 D4 A9 00 8D 8A 4A 60 7A

Chapter 2

1559:A9 09 8D 8B 4A A5 FB 85 6B
1561:FD A5 FC 85 FE 98 8D 3D 9E
1569:03 A9 10 8D 12 D4 A9 15 A6
1571:8D 12 D4 A2 18 A9 08 8D B0
1579:0F D4 20 75 4D 8A 8D 3E A9
1581:03 0A 0A EA EA 8D 08 D4 12
1589:A0 12 20 5B 4D AD 3E 03 E2
1591:AA CA D0 E6 AD 3D 03 A8 5D
1599:A5 FD 85 FB A5 FE 85 FC B7
15A1:A9 E0 8D 0F D4 A9 15 8D 80
15A9:12 D4 A9 03 8D 8B 4A 60 07
15B1:C0 00 F0 15 C0 01 D0 03 5A
15B9:4C 6F 4D 88 A2 C6 CA D0 AE
15C1:FD 88 D0 EC A2 C3 CA D0 80
15C9:FD 60 A9 41 85 FB A9 03 C6
15D1:85 FC A0 00 B1 FB C9 F1 15
15D9:F0 08 18 69 01 91 FB 4C AA
15E1:91 4D A9 E4 91 FB C8 C0 7A
15E9:08 D0 E9 60 A9 51 85 FB 29
15F1:A9 03 85 FC A0 00 B1 FB 96
15F9:C9 FF F0 08 18 69 01 91 A1
1601:FB 4C B3 4D A9 F2 91 FB C1
1609:C8 C0 08 D0 E9 A0 08 EE A8
1611:3C 03 AD 3C 03 8D 0F D4 D6
1619:88 D0 F4 60 AD 6C 03 18 9F
1621:69 29 85 FB AD 6D 03 85 6B
1629:FC AD 67 03 C9 00 D0 61 AD
1631:A0 00 B1 FB C9 00 F0 43 17
1639:A5 FB 38 E9 10 85 FB A0 0C
1641:00 B1 FB 18 69 19 CD 89 AF
1649:4A B0 03 4C 31 4E 20 AB 9A
1651:4C AD 89 4A 38 E9 17 91 0E
1659:FB A5 FB 18 69 10 85 FB 80
1661:A9 00 91 FB A5 FB 18 69 0B
1669:08 85 FB B1 FB C9 00 F0 8D
1671:03 4C 3C 4E A9 FF 91 FB 0B
1679:4C 3C 4E A5 FB 38 E9 10 A3
1681:85 FB A9 25 91 FB A5 FB BA
1689:18 69 10 85 FB A9 01 91 90
1691:FB 60 AD 6E 03 18 69 41 FC
1699:85 FB AD 6F 03 85 FC AD 0A
16A1:6A 03 C9 00 D0 61 A0 00 4A
16A9:B1 FB C9 00 F0 43 A5 FB C2
16B1:38 E9 10 85 FB A0 00 B1 E2
16B9:FB 38 E9 08 CD 89 4A 90 69
16C1:03 4C A7 4E AD 89 4A 18 9C
16C9:69 05 91 FB 20 03 4D A5 2B
16D1:FB 18 69 10 85 FB A9 00 9F
16D9:91 FB A5 FB 38 E9 08 85 41
16E1:FB B1 FB C9 00 F0 03 4C AA

Type It In

16E9:B2 4E A9 01 91 FB 4C B2 10
16F1:4E A5 FB 38 E9 10 85 FB 48
16F9:A9 F7 91 FB A5 FB 18 69 A1
1701:10 85 FB A9 FF 91 FB 60 51
1709:AD 6C 03 85 FB AD 6D 03 56
1711:85 FC A0 00 AD 65 03 C9 28
1719:FF F0 06 C9 01 F0 20 D0 BD
1721:3B A9 00 91 FB A5 FB C9 A8
1729:59 F0 0A C6 FB A5 FB 8D E9
1731:6C 03 4C 07 4F A9 60 85 B7
1739:FB 8D 6C 03 4C 07 4F A0 44
1741:00 A9 00 91 FB A5 FB C9 2B
1749:60 F0 0A E6 FB A5 FB 8D 8F
1751:6C 03 4C 07 4F A9 59 85 C9
1759:FB 8D 6C 03 AD 70 03 C9 A5
1761:11 D0 05 A9 0C 8D 70 03 02
1769:EE 70 03 AD 70 03 91 FB 15
1771:AD 6E 03 85 FB AD 6F 03 43
1779:85 FC AD 64 03 C9 FF F0 D5
1781:06 C9 01 F0 22 D0 3D A0 C3
1789:00 A9 0D 91 FB A5 FB C9 15
1791:49 F0 0A C6 FB A5 FB 8D 4A
1799:6E 03 4C 6F 4F A9 50 85 87
17A1:FB 8D 6E 03 4C 6F 4F A0 8E
17A9:00 A9 0D 91 FB A5 FB C9 35
17B1:50 F0 0A E6 FB A5 FB 8D EF
17B9:6E 03 4C 6F 4F A9 49 85 99
17C1:FB 8D 6E 03 AD 71 03 C9 52
17C9:09 D0 05 A9 03 8D 71 03 20
17D1:EE 71 03 AD 71 03 91 FB C5
17D9:AD A8 03 F0 04 CE A8 03 28
17E1:60 A9 03 8D A8 03 A9 82 0B
17E9:85 FB A9 03 85 FC A0 00 A0
17F1:B1 FB C9 00 D0 1A A5 FB 67
17F9:38 E9 29 85 FB A9 06 91 60
1801:FB A5 FB 18 69 29 85 FB 90
1809:C0 07 F0 19 C8 4C 9B 4F 09
1811:A5 FB 38 E9 29 85 FB A9 B9
1819:00 91 FB A5 FB 18 69 29 C3
1821:85 FB 4C B3 4F A9 8A 85 93
1829:FB A9 03 85 FC A0 00 B1 96
1831:FB C9 00 D0 1B A5 FB 38 7E
1839:E9 41 85 FB A9 01 91 FB 8F
1841:A5 FB 18 69 41 85 FB C0 B5
1849:07 F0 04 C8 4C DA 4F 60 13
1851:A5 FB 38 E9 41 85 FB A9 BA
1859:0D 91 FB A5 FB 18 69 41 A2
1861:85 FB 4C F2 4F A9 82 85 B7
1869:FB A9 03 85 FC A9 72 85 B3
1871:FD A9 03 85 FE A0 00 B1 EF

Chapter 2

1879:FB C9 FF F0 06 C9 01 F0 73
1881:26 D0 1B B1 FD 18 69 17 B1
1889:CD 89 4A B0 0A B1 FD 18 82
1891:69 01 91 FD 4C 49 50 20 11
1899:C2 50 A9 01 91 FB C0 07 89
18A1:F0 04 C8 4C 22 50 60 B1 ED
18A9:FD C9 1E 90 0A B1 FD 38 63
18B1:E9 01 91 FD 4C 49 50 A9 FA
18B9:FF 91 FB 4C 49 50 A9 8A FB
18C1:85 FB A9 03 85 FC A9 7A 07
18C9:85 FD A9 03 85 FE A0 00 0B
18D1:B1 FB C9 FF F0 06 C9 01 47
18D9:F0 1F D0 14 B1 FD C9 FB BA
18E1:B0 0A B1 FD 18 69 01 91 FC
18E9:FD 4C 9B 50 A9 01 91 FB 15
18F1:C0 07 F0 04 C8 4C 7B 50 61
18F9:60 B1 FD 38 E9 06 CD 89 96
1901:4A 90 0A B1 FD 38 E9 01 7E
1909:91 FD 4C 9B 50 20 D7 50 C9
1911:A9 FF 91 FB 4C 9B 50 A9 25
1919:10 8D 0C D4 A9 45 8D 08 0B
1921:D4 A9 80 8D 0B D4 A9 81 91
1929:8D 0B D4 60 A9 24 8D 05 83
1931:D4 A9 2F 8D 01 D4 A9 10 B5
1939:8D 04 D4 A9 11 8D 04 D4 04
1941:60 AD 00 DC A0 00 A2 00 27
1949:4A B0 01 88 4A B0 01 C8 55
1951:4A B0 01 CA 4A B0 01 E8 A1
1959:4A 8E 66 03 2A 29 01 8D D6
1961:68 03 AD 01 DC A0 00 A2 5A
1969:00 4A B0 01 88 4A B0 01 24
1971:C8 4A B0 01 CA 4A B0 01 A2
1979:E8 4A 8E 69 03 2A 29 01 2F
1981:8D 6B 03 AD 68 03 C9 00 73
1989:D0 11 AD 67 03 C9 00 D0 A4
1991:12 AD AB 03 C9 FF F0 0B 19
1999:EE AB 03 A9 01 8D 67 03 39
19A1:4C 5C 51 20 9B 4C A9 00 9E
19A9:8D AB 03 AD 68 03 8D 67 9A
19B1:03 AD 6B 03 C9 00 D0 11 6F
19B9:AD 6A 03 C9 00 D0 12 AD 6F
19C1:AC 03 C9 FF F0 0B EE AC 82
19C9:03 A9 01 8D 6A 03 4C 8A 63
19D1:51 20 8B 4C A9 00 8D AC FF
19D9:03 AD 6B 03 8D 6A 03 AD 60
19E1:66 03 C9 00 F0 11 AD 65 CD
19E9:03 C9 00 F0 12 AD AD 03 C4
19F1:C9 01 F0 0B EE AD 03 A9 F5
19F9:00 8D 65 03 4C B8 51 20 74
1A01:72 4C A9 00 8D AD 03 AD 8D

Type It In

1A09:66 03 8D 65 03 AD 69 03 DD
1A11:C9 00 F0 11 AD 64 03 C9 28
1A19:00 F0 12 AD AE 03 C9 01 BC
1A21:F0 0B EE AE 03 A9 00 8D A5
1A29:64 03 4C E6 51 20 4E 4C 3C
1A31:A9 00 8D AE 03 AD 69 03 7B
1A39:8D 64 03 60 A9 82 85 FB 12
1A41:A9 03 85 FC A9 00 8D 9A 8E
1A49:03 A0 00 B1 FB C9 00 D0 1A
1A51:03 EE 9A 03 C0 07 F0 04 4E
1A59:C8 4C F6 51 A9 8A 85 FB 77
1A61:A9 03 85 FC A9 00 8D 9B AF
1A69:03 A0 00 B1 FB C9 00 D0 3A
1A71:03 EE 9B 03 C0 07 F0 04 8E
1A79:C8 4C 16 52 AD 9B 03 CD BC
1A81:9A 03 90 6D F0 6A AD 89 C2
1A89:4A C9 3E D0 03 4C 8B 53 DD
1A91:38 E9 01 8D 89 4A A9 82 A0
1A99:85 FB A9 03 85 FC A0 00 56
1AA1:B1 FB C9 00 D0 15 A5 FB 09
1AA9:38 E9 10 85 FB B1 FB 38 A5
1AB1:E9 01 91 FB A5 FB 18 69 C3
1AB9:10 85 FB C0 07 F0 04 C8 AF
1AC1:4C 4B 52 A9 8A 85 FB A9 DF
1AC9:03 85 FC A0 00 B1 FB C9 13
1AD1:00 D0 15 A5 FB 38 E9 10 DB
1AD9:85 FB B1 FB 38 E9 01 91 C2
1AE1:FB A5 FB 18 69 10 85 FB 11
1AE9:C0 07 F0 04 C8 4C 78 52 59
1AF1:60 AD 89 4A C9 E8 D0 03 2E
1AF9:4C 07 53 18 69 01 8D 89 F5
1B01:4A A9 82 85 FB A9 03 85 81
1B09:FC A0 00 B1 FB C9 00 D0 D8
1B11:15 A5 FB 38 E9 10 85 FB D4
1B19:B1 FB 18 69 01 91 FB A5 AC
1B21:FB 18 69 10 85 FB C0 07 2E
1B29:F0 04 C8 4C B6 52 A9 8A 93
1B31:85 FB A9 03 85 FC A0 00 EF
1B39:B1 FB C9 00 D0 15 A5 FB A2
1B41:38 E9 10 85 FB B1 FB 18 1F
1B49:69 01 91 FB A5 FB 18 69 1D
1B51:10 85 FB C0 07 F0 04 C8 49
1B59:4C E3 52 60 A9 60 8D 0C F4
1B61:D4 8D 05 D4 A9 00 8D 5C 18
1B69:03 A9 20 8D 75 03 A9 FF 73
1B71:8D 85 03 A9 FA 8D 89 4A 36
1B79:A9 08 8D 15 D0 A9 08 8D 54
1B81:1D D0 8D 17 D0 A9 5C 8D 11
1B89:AF 03 A9 10 8D 04 D4 A9 5E
1B91:15 8D 04 D4 AD 3D 03 8D 79

Chapter 2

1B99:0F D4 AD 75 03 8D 01 D4 BE
1BA1:20 97 4D 20 10 50 20 10 8B
1BA9:50 A0 40 20 5B 4D AD AF 55
1BB1:03 C9 01 F0 06 CE AF 03 D8
1BB9:4C 35 53 A9 00 8D 15 D0 99
1BC1:A9 FF 8D 3D 03 A0 FF 20 0D
1BC9:5B 4D AD 3D 03 D0 06 CE C0
1BD1:3D 03 4C 70 53 A9 00 8D C6
1BD9:17 D0 8D 1D D0 4C AB 40 A2
1BE1:A9 15 8D 13 D4 8D 05 D4 D0
1BE9:A9 0D 8D 4C 03 A9 E0 8D BC
1BF1:7D 03 A9 01 8D 8D 03 A9 3F
1BF9:1C 8D 89 4A A9 08 8D 15 15
1C01:D0 A9 08 8D 1D D0 8D 17 44
1C09:D0 A9 5C 8D AF 03 A9 10 65
1C11:8D 12 D4 A9 15 8D 12 D4 A1
1C19:AD 3D 03 8D 01 D4 AD 7D E4
1C21:03 8D 0F D4 20 75 4D 20 FE
1C29:69 50 20 69 50 A0 40 20 6A
1C31:5B 4D AD AF 03 C9 01 F0 4D
1C39:06 CE AF 03 4C B9 53 A9 E7
1C41:00 8D 15 D0 A9 FF 8D 3D 32
1C49:03 A0 FF 20 5B 4D AD 3D D5
1C51:03 D0 06 CE 3D 03 4C F4 70
1C59:53 A9 00 8D 17 D0 8D 1D B2
1C61:D0 4C AB 40 AC 3F 03 A9 A0
1C69:0A 8D 3F 03 CE 3F 03 88 24
1C71:D0 FA AD A0 03 C9 3C F0 39
1C79:06 EE A0 03 4C 3B 54 20 CC
1C81:EC 50 20 C7 4D 20 3D 4E 78
1C89:20 B3 4E A9 00 8D A0 03 9D
1C91:AD A1 03 C9 0F F0 06 EE 3D
1C99:A1 03 4C 53 54 20 10 50 B5
1CA1:20 69 50 A9 00 8D A1 03 65
1CA9:AD A2 03 C9 1E F0 06 EE 0E
1CB1:A2 03 4C 6B 54 20 75 4D 97
1CB9:20 97 4D A9 00 8D A2 03 AA
1CC1:AD A3 03 C9 FF F0 06 EE 75
1CC9:A3 03 4C 80 54 20 E7 51 6A
1CD1:A9 00 8D A3 03 AE 3F 03 1F
1CD9:A0 18 88 D0 FD CA D0 F8 3C
1CE1:4C 1D 54 00 00 00 00 00 12

Chapter 3
The
Source Code



The Fast Assembler

Yves Han

Here's a truly amazing machine language assembler. "The Fast Assembler" supports multiple statement lines, labels, and macrolike INCLUDE files. It can assemble to memory or to disk. Written very compactly, it occupies only about 2600 bytes, leaving the rest of memory for your source code. It also adds to the BASIC editor several new features useful to both BASIC and machine language programmers.

Symbolic label-based assemblers are the most convenient way to write machine language (ML) programs. The instructions are entered as *source code* and later assembled into object code (the actual ML program—the numbers in memory). And rather than always having to specify memory addresses, you can name routines with meaningful labels. It's as if you could enter GOSUB JOYSTICK in BASIC.

Saving Memory by Using the BASIC Editor

You write the source code for your ML programs for "The Fast Assembler" (FA) with the BASIC editor. You save the source code to tape or disk as you would a BASIC program, and list it to a printer exactly as you would list BASIC.

The FA is an extension of the BASIC interpreter especially designed for writing programs in machine language. Writing it as a BASIC extension kept the program short (under 2600 bytes) because many subroutines of the BASIC interpreter could be used. Some modifications have been made to BASIC to make writing programs easier by copying BASIC ROM into the underlying RAM.

Even if you don't write programs in machine language, you can still use the assembler because of the new features

Chapter 3

added to BASIC and the extra BASIC commands it adds. The assembler will execute a BASIC program just as normal BASIC would.

Typing Fast Assembler

You'll need a copy of "MLX" to type in The Fast Assembler. First enter MLX and save a copy or two. When you are ready to enter the data for FA, run MLX and answer the starting and ending address prompts as follows:

Starting Address: 0801
Ending Address: 1220

After entering all the data for The Fast Assembler, save a copy on tape or disk. To start up FA, first load it as if it were a BASIC program (don't use an extra ,1 in the LOAD command). Type

LOAD "filename",8 (for disk)
LOAD"filename" (for tape)

Press RETURN; then type

RUN

Press RETURN again. The enabling SYS is built into the first line of the program. The screen will clear, and a message will appear at the top of the screen, indicating FA has been enabled. You can now start programming—in BASIC or machine language.

BASIC Modifications and Enhancements

The following changes have been made to the BASIC interpreter:

- **Structured listings.** Spaces between the line number and the first character on the line are not deleted. This makes it possible to indent lines and makes listings easier to read.
- **List pause.** You can freeze a listing by holding down the SHIFT key or pressing SHIFT-LOCK. Listing can be continued by releasing the SHIFT key.
- **ASCII translations and hexadecimal/binary numbers.** In arithmetic expressions, you can use hexadecimal and binary numbers. Hexadecimal numbers should be preceded by \$ and binary numbers by %. You can also use a character preceded by a single quote ('A is the same as ASC("A")). The single quote preceding a character can be used to find the

value of a BASIC token, as well. For example, PRINT 'END will print the value 128, which is the BASIC code for END. If you put a space between the quotation mark and the character, the ASCII value of the space will be used instead of the character value.

- **Variable and function names.** The rules for variable and function names have been changed a bit. Instead of the first *two*, the first *eight* characters are recognized. FA recognizes NUMBER1 and NUMBER2 as separate variables, while ordinary BASIC would consider them the same variable (NU). Variables may contain, but not be equal to, BASIC/ assembler commands or mnemonics: LAND is a legitimate variable name, even though it contains the keyword AND. But variable labels starting with *TI* or *ST* (reserved keywords) are not automatically set to zero the first time you use them. An exception to this recognition of eight-character names is that only the first two characters of array variables are significant.
- **Keywords.** Because variable and function names may contain keywords, FA has to be able to decide whether a keyword is a keyword or part of a variable or function name. The assembler recognizes a keyword if it's followed by a space or non-alphabetic character. For example, in PRINT "OK" the keyword PRINT will be recognized as a PRINT command, but in A\$="OK":PRINTA\$, the keyword PRINT is interpreted as part of the variable name PRINTA\$. You would have to insert a space (PRINT A\$) if you wanted to print the variable A\$. In keywords like PRINT# or GET#, you must remember to include a space between the keyword and the logical file number value. For example, you must use PRINT# 1; a statement like PRINT#1 will cause a syntax error.
- **REM and DATA.** Capital letters in REM and DATA lines are listed as capital letters and not as tokenized BASIC keywords. For example, **10 rem AB** lists as it's entered and not as **10 rem atnpeek** as normal BASIC would do.

New BASIC Commands

AUTO *step value*

This command turns on automatic line numbering and defines the step value between the line numbers. To enter AUTO mode, type **AUTO** followed by the step value, and press RETURN. Then enter a line with a line number. The next line number prints automatically. To leave auto mode, move the

Chapter 3

cursor to an empty line and press RETURN. To turn automatic line numbering off altogether, enter AUTO only.

You can also use this command to delete part of a program. Turn automatic line numbering on with a step value of 1. Type the number of the first line you want to delete and press RETURN. Keep pressing RETURN until you've reached the end of the section you want to delete. Instead of pressing RETURN again and again, you can enter POKE 650,128 and hold RETURN down until you've reached the last line to be erased.

OLD

If you accidentally type NEW, you can restore your program with this command. It can also be used if you've installed a reset button. If you've assembled a program and are testing it, sometimes your computer locks up. Use the reset button and then enter SYS 4408 to restart the assembler; type OLD to restore the source program. If your program has not destroyed the assembler or the source program, everything will be there.

Semicolon (;)

This has the same function as the REM statement. It need not be separated with a colon from the preceding command. For example,

```
10 X=0:REM SET X TO ZERO
```

is the same as

```
10 X=0;SET X TO ZERO
```

The semicolon in the commands PRINT and INPUT is not treated as a REM statement, but as a separator.

Using Labels as Variables and Addresses

Label names follow the same rules as variable names. They can be used in arithmetic expressions the way normal variables are used.

You can define a label in two ways. You can place the label name just before the command to which you want to refer. If more commands are on the same line, you must separate the label from the commands with a colon. Or you can label the current program counter: LABELNAME=*. The asterisk (*) is a special variable which gives the value of the program counter. The counter is the address where the next instruction or datum will be placed. You can only read the variable *. You cannot assign a value to it with the statement *=expr.

Here's an example of using labels to mark routines in a program (don't type this in; it's only a fragment of a program):

```
50 JSR DISPLAY1; JUMP TO LABELED SUBROUTINE (LINE 90)
60 LDA $FF: BNE SKIPIT ; CONDITIONAL BRANCH AHEAD
   TO SKIPIT
70 TYA
80 SKIPIT: LDX #4: STA $8000,X: RTS;TARGET OF BRANCH
   IN 60
90 DISPLAY1=* ; THIS LABELS THE CURRENT PROGRAM
   COUNTER
100 ;
110 LDA #65: JSR $FFD2: RTS
```

Remember that in the lines above, the semicolon marks the beginning of a comment which, like a REM, is ignored by FA. The technique in line 90 is valuable if you think you may be adding some code at the beginning of the routine. As listed, the subroutine called DISPLAY1 starts with LDA #65, but later you could go in and add some lines between 90 and 110.

Three Passes to Assemble

Three passes are required to assemble source code (what you write) into object code (an executable ML program that the computer can follow). But FA doesn't do it by itself. You have to insert a loop that repeats three times with BASIC commands:

```
10 FOR PASS=1 TO 3
   .
   . (Insert source code)
   .
90 NEXT PASS:END
```

If you use an invalid addressing mode such as *LSR (expr),y* you'll see *ILLEGAL ADDRESSING MODE ERROR*. Mnemonics can only be used in program mode—that is, in a program you execute with RUN. If you enter a mnemonic in direct mode, you'll see *ILLEGAL DIRECT ERROR*.

Also note that for immediate addressing, the argument can be an actual number or an arithmetic expression which evaluates to a value in the range 0–255. Or you can substitute a string expression, in which case the assembler takes the ASCII value of the first character as the argument. If the string length is zero, the argument becomes zero.

Chapter 3

Assembler Commands

Assembler commands which write data to the output device can only be used in program mode; otherwise you'll get *ILLEGAL DIRECT ERROR*. All assembler commands must be included in every pass.

ORG *address,mode,device,name*

This command must be used at the start of each pass. It does several things. First, it sets the origin (ORG), the memory address for the beginning of the ML program. It assigns an initial value to the program counter. It also sets the assembler mode, which should be 0 on the first two passes and 1 on the third and last. ORG also sets the output device and filename (if necessary).

Not all arguments are necessary. Also permitted are

ORG

ORG *address*

ORG *address,mode*

Default values for the arguments are

address = 49152 (\$C000)

mode = 0

device = 0 and no name

If you use a mnemonic or assembler command before you've used the command ORG, you'll see *UNDEF'D LOCATION COUNTER ERROR*.

The address assigns a value to the program counter. Usually, you use more than one pass to assemble the source program. Only during the last pass should the object code be written to memory or to the output device. *Mode* tells the assembler when the last pass is reached. Zero means it's not the last pass, so no object code should be produced, and there's no range checking for arguments and no checking for too large branches.

On the final pass, you should set the mode to 1, which signals the last pass, when object code is written to the output device.

Finally, you set the device number of the output device and a string expression which contains the filename if the object code is not written to memory. Zero means the output device is memory. Be careful not to write to memory locations where the assembler is placed (\$0801-\$121B) or where the BASIC interpreter is placed (\$A000-\$BFFF).

A device number in the range 8-11 means the output device is a disk drive. If *mode* is equal to 1, the assembler will open a PRG file with the name specified in the argument name. The logical file number will be 8.

BYTE *expression,expression,...*

This command writes numbers or characters to memory or the selected output device. It can have one or more arithmetic or string expressions separated by commas. Arithmetic expressions must give a positive value less than 256. The value will be placed in one byte. Each character of a string expression will be placed in one byte.

WORD *expression,expression,...*

This has the same function as BYTE except that values of arithmetic expressions must be positive and less than 65536. The value will be placed in two bytes in low/high format.

INCLUDE *name,device*

This command assembles a file from disk and inserts the resulting object code into memory or the output device. The file must be a normal PRG file and may not contain BASIC commands which cause a branch to another line or stop the program. Also not permitted are the BASIC commands DEF, RETURN, CLR, NEW, and the assembler commands SEND and INCLUDE.

The file is opened with a logical file number of 9. The file is closed when the end of the file is reached. The name is the filename you're including, and the device number can be 8-11 (use 8 if you have a single drive). If you have only one disk drive and you assemble to disk, the file(s) for the command INCLUDE must be on the same disk to which you assemble.

All variables and labels are *global*, which means you can pass parameters to INCLUDE files so they can work like macro-instructions. Let's say you're writing a program that needs to access several different disk files, and there are several points in the program that use the Kernal routines SETLFS, SETNAM, and OPEN. You could write the source code that performs these Kernal calls and save it to disk under the program name "OPEN" to be used later. Then, in the main program, use INCLUDE "OPEN",8. When the source code is compiled, the series of commands from the OPEN file are automatically inserted in the proper place in the object code.

Chapter 3

SEND *stringexpr*

The command **SEND** may be used only if the object program is written to disk. It's used to link object code to a BASIC program. *Stringexpr* must contain a BASIC line with line number. If you forget the line number, you'll get *MISSING LINE NUMBER ERROR*. If you want to send more than one line, you must use **SEND** for each line, and you have to send the lines in the right order. You must send the lines before the actual object code is written to disk. The address in the **ORG** command must be the start of BASIC RAM (2049).

UNSEND

If you load a program which consists of both BASIC and ML, the interpreter has to know where the BASIC part ends. **UNSEND** places a mark which the computer recognizes as the end of the BASIC part.

Example Programs

```
100 FOR PASS=1 TO 3:PRINT "PASS"PASS,
110 ORG $C000
120 IF PASS=3 THEN ORG $C000,1
130 START: LDX #0
140 LOOP: LDA TEXT,X:PRINT TEXT
150 BEQ EXIT
160 JSR $FFD2
170 INX
180 BNE LOOP
190 EXIT: RTS
200 PRINT *
210 TEXT: BYTE "EXAMPLE 1",0
220 NEXT PASS:END
```

Lines 110 and 120 show how to use the command **ORG**. In every pass, line 110 sets mode 0. But in pass 3, line 120 sets mode 1. The object code will start at 49152 (hexadecimal \$C000). Line 200 prints the current value of the location counter (*).

You can assemble the program with the command **RUN**. The program will give the following output:

```
PASS 1 0      49165
PASS 2 49165  49166
PASS 3 49166  49166
```

The first column is the pass number. The second column is the value of the label **TEXT** in the instruction **LDA TEXT,X** in line 140. The third column is the value the label should

The Source Code

have when the source code is assembled. You can see that only in pass 3 are these values equal to each other. This is because the assembler defaults to zero-page addressing. In pass 1, TEXT has a value less than 256, so zero-page addressing is assumed. This means a two-byte instruction instead of three. The value assigned to TEXT will be too low, as you can see in pass 1. In pass 2, this value, which is too low, will be used in assembling line 140. The assembler decides not to use zero-page addressing, so TEXT is assigned the correct value. In pass 3, the correct value replaces the previously incorrect values during assembly.

```
5 ;EXAMPLE PROGRAM 2
6 ;
10 PRINT CHR$(147)
11 DEF FN H(X)=INT(X/256)
12 DEF FN L(X)=X-256*FN H(X)
20 PRINT:PRINT" LOADER MAKER"
30 PRINT:PRINT" ENTER THE NAME OF THE PROGRAM T
   HAT"
40 PRINT" HAS TO BE LOADED BY THE LOADER."
50 INPUT" >";NAME$
60 PRINT:PRINT" ENTER THE NAME OF THE LOADER."
70 INPUT" >";N$
80 PRINT:PRINT" ENTER THE ADDRESS TO EXECUTE TH
   E"
90 PRINT" PROGRAM."
100 INPUT" >";ADDRESS:ADDRESS=ADDRESS-1
105 ;
110 FOR PASS=1 TO 3
115 ;
120 ORG 2049
130 IF PASS=3 THEN ORG 2049,1,8,N$
135 ;
140 SEND "10 SYS"+STR$(LOADER)
150 UNSEND
160 LOADER: LDA #8:TAX:LDY #1
170         JSR $FFBA
180         LDX #FN L(NAME)
190         LDY #FN H(NAME)
200         LDA #LEN(NAME$)
210         JSR $FFBD
220         LDA #FN H(ADDRESS)      :PHA
230         LDA #FN L(ADDRESS)     :PHA
240         LDA #0:JMP $FFD5
250 NAME:   BYTE NAME$
255 ;
260 NEXT PASS:CLOSE 8:END
```

Chapter 3

The above example program shows how to use the commands SEND and UNSEND to write a program which includes a SYS within a BASIC line.

The main routine at lines 160–250 illustrates how to load another program from an ML program. Note that the lines up to 100 are BASIC—they prepare the variables and defined functions for use in the source code. If you assemble the program with the command RUN, you'll get a program that can load another ML program from disk and execute it. The object code will be written to disk.

In line 140, the command SEND writes a BASIC line to the output device by which you can load and run the program as if it were a normal BASIC program. Line 150 marks the end of the BASIC part of the object code.

The INPUTs in lines 50, 70, and 100 permit you to enter the parameters for the object program when the source program is assembled. In this way you can make different object programs with one source program.

Another advantage of writing the assembler as a BASIC extension is that you can assemble a program to the top of memory. Use the following construction to do this:

```
100 POKE 56,PEEK(56)-4:CLR
110 TOPOFMEM=PEEK(55)+256*(PEEK(56)+4)
120 ADDRESS=0:MODE=0
130 FOR PASS=1 TO 3
140 ORG ADDRESS
150 IF PASS=3 THEN ORG ADDRESS, MODE
```

```
.
.
. Source code
```

```
.
.
900 NEXT PASS
910 IF MODE=1 THEN END
920 ADDRESS=TOPOFMEM-*
930 MODE=1:GOTO 130
```

In this program, the source code goes through six passes. During the first three passes the location counter remains at 0. Mode 0 is used so the object program will not be written to the output device. The length of the program is calculated and subtracted from TOPOFMEM. This address is used in the second three passes to assemble to the top of memory. MODE is set to 1 so the assembler will write the object code to the out-

put device during the sixth pass (actually pass 3 of the second time around). Line 100 is used to reserve 1K at the top of memory for the object program.

Large Programs

If your source program won't fit into memory, you can split your program and use the command INCLUDE—for example,

```
10 FOR PASS=1 TO 3
20 ORG ADDRESS
30 IF PASS=3 THEN ORG ADDRESS,1
```

. Part 1 of source code

```
90 INCLUDE "PART 2",8
100 INCLUDE "PART 3",8
110 NEXT PASS:END
```

The labels and variables used in the INCLUDE files will be global variables, which means you can use them in arithmetic expressions everywhere in the program.

Another possibility is chaining the programs, but then you can't use a FOR-NEXT loop for the passes. You must use another way to define the passes—for example,

```
FIRSTPROGRAM
10 PASS=PASS+1:IF PASS=4 THEN
END
20 ORG ADDRESS
30 IF PASS=3 THEN ORG ADDRESS,1
```

. Source code

```
90 LOAD"SECONDPROGRAM",8
SECONDPROGRAM
```

. Source code

```
90 LOAD"FIRSTPROGRAM",8
```

Note that these are just examples. You'd have to insert your own source code as indicated. To chain programs, you would load and execute the first program. It controls the number of passes and loads the next program. The next program loads the following program and so on until the last program, which must load the first again.

Chapter 3

The Fast Assembler

For mistake-proof program entry, use "MLX: Machine Language Entry Program" to type in this program.

```
0801:0C 08 0A 00 9E 20 34 34 6C
0809:30 38 00 00 00 A9 36 85 D7
0811:01 A6 7A CA 30 0F BD 00 80
0819:02 C9 20 F0 F6 E8 BD 00 86
0821:02 C9 20 D0 01 E8 A0 04 A6
0829:84 0F BD 00 02 10 0A C9 25
0831:FF F0 06 E9 9F D0 16 A9 F2
0839:20 C9 20 F0 0C 85 08 C9 2F
0841:22 F0 4A C9 3F D0 4F A9 0A
0849:99 C8 99 FB 01 E8 D0 DA 73
0851:A4 71 24 0B 10 06 C8 A9 19
0859:90 99 FB 01 A5 0B 09 80 93
0861:C8 99 FB 01 24 0B 30 C2 3C
0869:C9 83 F0 05 49 8F D0 BA 92
0871:2C A9 3A 85 08 BD 00 02 DA
0879:F0 1A 24 0F 70 04 C5 08 25
0881:F0 C7 C9 22 D0 07 0A 45 53
0889:0F 85 0F A9 22 C8 99 FB 62
0891:01 E8 D0 E1 F0 5E 84 71 10
0899:86 7A A0 9E 84 22 A0 A0 18
08A1:84 23 A0 00 84 0B 51 22 E5
08A9:D0 07 E8 C8 BD 00 02 D0 50
08B1:F5 C9 80 D0 26 E8 BD 00 9C
08B9:02 20 1D 09 B0 92 A5 0B 2D
08C1:30 19 84 08 A0 0B D9 2E 54
08C9:09 F0 F1 88 10 F8 A4 08 17
08D1:C9 34 90 07 BD 00 02 C9 12
08D9:28 F0 E1 88 C8 B1 22 10 60
08E1:FB E6 0B A6 7A C8 B1 22 F1
08E9:D0 C2 C8 F0 0A A4 71 BD BE
08F1:00 02 D0 11 4C C9 A5 A9 2C
08F9:3A 85 22 A9 09 85 23 A9 B5
0901:80 85 0B D0 A7 E8 C8 99 2F
0909:FB 01 20 1D 09 B0 0A BD 0C
0911:00 02 F0 E0 20 1D 09 90 E7
0919:EC 4C 2B 08 C9 30 90 0B D5
0921:C9 3A 90 08 C9 41 90 03 B0
0929:C9 5B 24 38 60 04 18 23 65
0931:26 31 32 33 2E 2D 2C 2B C5
0939:2A 42 59 54 C5 57 4F 52 DD
0941:C4 4F 52 C7 41 55 54 CF 28
0949:4F 4C C4 49 4E 43 4C 55 B0
0951:44 C5 53 45 4E C4 55 4E 34
0959:53 45 4E C4 42 43 C3 42 65
0961:43 D3 42 45 D1 42 4D C9 A2
0969:42 4E C5 42 50 CC 42 56 9D
0971:C3 42 56 D3 41 44 C3 41 E1
```

The Source Code

```
Ø979:4E C4 43 4D DØ 45 4F D2 2E
Ø981:4C 44 C1 4F 52 C1 53 42 7A
Ø989:C3 53 54 C1 41 53 CC 44 2E
Ø991:45 C3 49 4E C3 4C 53 D2 ØE
Ø999:52 4F CC 52 4F D2 53 54 28
Ø9A1:D8 43 5Ø D8 43 5Ø D9 4C E3
Ø9A9:44 D8 4C 44 D9 53 54 D9 8Ø
Ø9B1:4A 53 D2 4A 4D DØ 42 49 38
Ø9B9:D4 42 52 CB 43 4C C3 43 E3
Ø9C1:4C C4 43 4C C9 43 4C D6 23
Ø9C9:44 45 D8 44 45 D9 49 4E 21
Ø9D1:D8 49 4E D9 4E 4F DØ 5Ø AB
Ø9D9:48 C1 5Ø 48 DØ 5Ø 4C C1 31
Ø9E1:5Ø 4C DØ 52 54 C9 52 54 31
Ø9E9:D3 53 45 C3 53 45 C4 53 2C
Ø9F1:45 C9 54 41 D8 54 41 D9 2C
Ø9F9:54 53 D8 54 58 C1 54 58 36
ØAØ1:D3 54 59 C1 ØØ 48 AD 8D 65
ØAØ9:Ø2 C9 Ø1 FØ F9 68 1Ø 16 67
ØA11:C9 FF FØ 12 24 ØF 3Ø ØE 15
ØA19:5Ø ØF C5 Ø8 DØ Ø8 48 A5 2F
ØA21:ØF 49 4Ø 85 ØF 68 4C F3 16
ØA29:A6 C9 9Ø FØ 18 48 C9 83 1D
ØA31:FØ Ø5 49 8F DØ ØB 2C A9 D5
ØA39:3A 85 Ø8 A5 ØF 49 4Ø 85 CA
ØA41:ØF 68 4C 24 A7 C8 B1 5F E5
ØA49:3Ø Ø5 88 A9 9Ø DØ F3 84 96
ØA51:49 AA AØ FF CA 1Ø Ø8 C8 38
ØA59:B9 3A Ø9 1Ø FA 3Ø F5 C8 48
ØA61:B9 3A Ø9 1Ø Ø3 4C EF A6 D2
ØA69:2Ø 47 AB DØ F2 2Ø 73 ØØ EØ
ØA71:A2 ØØ 86 ØD C9 AC FØ 16 71
ØA79:86 62 86 63 C9 24 FØ 3F 7Ø
ØA81:C9 25 FØ 28 C9 27 FØ 17 48
ØA89:2Ø 79 ØØ 4C 8D AE 2Ø 73 AB
ØA91:ØØ A5 FB A4 FC 84 62 85 1D
ØA99:63 A2 9Ø 38 4C 49 BC 2Ø BE
ØAA1:AC ØB 48 2Ø 73 ØØ 68 DØ 17
ØAA9:EE 4C Ø8 AF 2Ø 73 ØØ BØ C3
ØAB1:E8 C9 32 BØ E4 4A 26 63 FD
ØAB9:26 62 9Ø FØ 4C 48 B2 2Ø A3
ØAC1:73 ØØ 9Ø ØA C9 47 BØ D1 EØ
ØAC9:C9 41 9Ø CD E9 Ø7 29 ØF CE
ØAD1:A2 Ø4 Ø6 63 26 62 BØ E4 3Ø
ØAD9:CA DØ F7 Ø5 63 85 63 9Ø 5F
ØAE1:DE 8D A8 Ø2 85 45 2Ø B2 32
ØAE9:ØB 2Ø 13 B1 BØ Ø3 4C Ø8 3B
ØAF1:AF A2 ØØ 86 ØD 86 ØE 2Ø AD
ØAF9:AC ØB 9Ø Ø5 2Ø 13 B1 9Ø CA
ØBØ1:ØA EØ Ø7 BØ F2 E8 9D A8 5F
ØBØ9:Ø2 DØ EC E8 EØ Ø8 BØ Ø4 ØD
```

Chapter 3

0B11:A9 00 F0 F2 AD A9 02 85 E6
0B19:46 20 B2 0B C9 24 D0 06 E7
0B21:A9 FF 85 0D D0 13 C9 25 19
0B29:D0 1B A5 10 D0 C0 A9 80 81
0B31:85 0E 05 45 85 45 8D A8 87
0B39:02 A5 46 09 80 85 46 8D 47
0B41:A9 02 20 AC 0B 05 10 49 51
0B49:28 D0 03 4C D1 B1 84 10 3B
0B51:A5 2D A6 2E 86 60 85 5F 5D
0B59:E4 30 90 04 C5 2F B0 1A A6
0B61:A0 00 B9 A8 02 51 5F D0 6E
0B69:08 C8 C0 08 90 F4 4C 85 C4
0B71:B1 A5 5F 69 0D 90 DF E8 9F
0B79:D0 DA 68 48 C9 2A D0 03 DB
0B81:4C 23 B1 AD AA 02 D0 03 99
0B89:4C 28 B1 A5 2F A4 30 20 EC
0B91:47 B1 A0 07 B9 A8 02 91 42
0B99:5F 88 10 F8 A0 07 4C 74 41
0BA1:B1 10 05 A0 08 4C C7 B5 F5
0BA9:4C F6 B5 E6 7A D0 02 E6 CA
0BB1:7B A0 00 B1 7A C9 3A B0 E8
0BB9:03 4C 84 00 60 A6 9D 30 FD
0BC1:29 E0 21 F0 0B E0 20 F0 E4
0BC9:1A A9 0E A0 0E 4C 27 11 24
0BD1:A6 FD F0 0D A2 08 20 18 37
0BD9:E1 20 0C E1 20 CC FF F0 AD
0BE1:02 81 FB E6 FB D0 02 E6 55
0BE9:FC 60 4C AB B3 A5 66 30 0C
0BF1:12 A5 61 C9 91 B0 0C 20 CA
0BF9:9B BC A4 64 A5 65 85 14 C9
0C01:84 15 60 A5 9D 4A 90 03 41
0C09:4C 48 B2 A9 00 A8 F0 EE BD
0C11:20 73 00 F0 04 C9 3B D0 B3
0C19:07 A9 00 85 15 85 FE 60 94
0C21:C9 23 D0 04 A2 01 D0 08 04
0C29:A2 00 C9 28 D0 05 A2 80 AE
0C31:20 73 00 86 FE 20 9E AD 02
0C39:24 0D 30 0A 20 EE 0B A5 C5
0C41:FE 30 1C F0 51 60 A5 FE CD
0C49:C9 01 F0 03 4C 08 AF 20 D6
0C51:A6 B6 A0 00 84 15 C9 00 8A
0C59:F0 02 B1 22 85 14 60 A2 A2
0C61:00 86 0B A0 00 BD 9F 0C C8
0C69:51 7A D0 04 E8 C8 D0 F5 25
0C71:0A D0 0D A6 0B BD AC 0C 83
0C79:85 FE 20 FB A8 4C 73 00 35
0C81:E8 BD 9E 0C 10 FA E6 0B 57
0C89:BD 9F 0C D0 D6 A5 FE 30 72
0C91:BB A9 02 85 FE 60 A9 03 5A
0C99:85 0B A2 08 D0 C5 2C 58 5A
0CA1:A9 29 2C D9 A9 00 2C D8 7A

The Source Code

```
ØCA9:2C D9 00 05 06 0F 03 04 15
ØCB1:20 73 00 20 C4 0C 20 79 C8
ØCB9:00 C9 3B D0 03 20 3B A9 71
ØCC1:4C AE A7 F0 05 C9 3B D0 46
ØCC9:05 60 38 4C ED A7 C9 AF 99
ØCD1:D0 F8 A2 09 8A D0 72 D0 C2
ØCD9:03 4C 2F A8 E9 80 B0 03 AC
ØCE1:4C 08 AF C9 40 B0 F9 C9 37
ØCE9:08 B0 0D 0A A8 B9 C7 0D 3D
ØCF1:48 B9 C6 0D 48 4C 73 00 A0
ØCF9:E9 08 AA C9 08 B0 39 BD 2E
ØD01:D6 0D 20 BE 0B 20 73 00 79
ØD09:F0 D6 C9 3B F0 D2 20 8A DB
ØD11:AD A5 9D 4A 90 18 20 F7 E0
ØD19:B7 18 A5 14 E5 FB AA 08 87
ØD21:A5 15 E5 FC 28 10 02 49 AE
ØD29:FF C9 00 D0 04 8A 4C BE 64
ØD31:0B A9 26 A0 0E 4C 27 11 0B
ØD39:C9 1F 90 0D C9 38 B0 A0 14
ØD41:BD D6 0D 20 BE 0B 4C 73 C1
ØD49:00 48 BD D6 0D 85 02 20 3D
ØD51:11 0C A5 15 20 9D 0D F0 7F
ØD59:0C 20 9D 0D F0 07 A9 46 43
ØD61:A0 0E 4C 27 11 A5 02 29 97
ØD69:0F A8 B9 39 0E 10 06 A5 62
ØD71:FE 09 10 85 FE 68 38 E9 9B
ØD79:08 A6 FE A8 18 B9 5D 0E 1C
ØD81:7D 74 0E 20 BE 0B A5 FE A7
ØD89:29 0F F0 38 29 08 08 A5 BC
ØD91:14 20 BE 0B 28 F0 2D A5 4B
ØD99:15 4C BE 0B F0 06 A5 FE C3
ØDA1:09 08 85 FE A5 FE 4A 4A EA
ØDA9:4A 4A A5 02 B0 04 29 0F 47
ØDB1:90 04 4A 4A 4A 4A A8 A5 75
ØDB9:FE 29 07 AA B9 39 0E 49 41
ØDC1:FF 3D 94 0E 60 D4 0F D1 E4
ØDC9:0F A7 0E 42 0F 11 10 2A 42
ØDD1:10 B8 10 B0 10 90 B0 F0 44
ØDD9:30 D0 10 50 70 05 05 05 ED
ØDE1:05 05 05 05 06 17 11 11 70
ØDE9:17 17 17 23 28 28 39 1A D8
ØDF1:2B 2C 4C 24 00 18 D8 58 E2
ØDF9:B8 CA 88 E8 C8 EA 48 08 4D
ØE01:68 28 40 60 38 F8 78 AA AA
ØE09:A8 BA 8A 9A 98 55 4E 44 1E
ØE11:45 46 27 44 20 4C 4F 43 9E
ØE19:41 54 49 4F 4E 20 43 4F D1
ØE21:55 4E 54 45 D2 42 52 41 DF
ØE29:4E 43 48 20 4F 55 54 20 E0
ØE31:4F 46 20 52 41 4E 47 C5 47
ØE39:1C 0C 04 14 84 6E 6C 0D EB
```

Chapter 3

0E41:86 96 8E 8C 80 49 4C 4C EE
0E49:45 47 41 4C 20 41 44 44 99
0E51:52 45 53 53 49 4E 47 20 B9
0E59:4D 4F 44 C5 60 20 C0 40 1A
0E61:A0 00 E0 80 01 C1 E1 41 06
0E69:21 61 81 E0 C0 A2 A0 80 FE
0E71:14 40 20 09 09 05 15 15 D7
0E79:01 11 80 09 80 0D 1D 19 86
0E81:80 80 80 80 00 04 14 14 62
0E89:80 80 80 80 80 0C 1C 1C A6
0E91:80 80 2C 01 02 04 08 10 E3
0E99:20 40 80 20 79 00 F0 04 99
0EA1:C9 3B D0 02 68 68 60 A9 FA
0EA9:00 85 FB 85 FD 85 B7 A9 1E
0EB1:C0 85 FC A5 9D 09 20 85 60
0EB9:9D 20 9C 0E 20 8A AD 20 C7
0EC1:F7 B7 84 FB 85 FC 20 9C 15
0EC9:0E 20 00 E2 E0 02 90 03 56
0ED1:4C 48 B2 8A 05 9D 85 9D 6C
0ED9:20 9C 0E 20 00 E2 86 FD 87
0EE1:20 0E E2 20 9E AD 20 A3 7F
0EE9:B6 20 BD FF A6 FD F0 B6 E6
0EF1:A9 08 A0 01 20 BA FF A5 9A
0EF9:9D 4A 90 AA A5 B7 D0 06 E7
0F01:A5 FD C9 08 B0 15 20 C1 07
0F09:E1 A2 08 20 18 E1 A5 FB 53
0F11:20 0C E1 A5 FC 20 0C E1 3B
0F19:4C CC FF 20 10 F7 4C 3B C6
0F21:A4 20 8B B0 85 49 84 4A BA
0F29:20 79 00 F0 07 C9 3B F0 8B
0F31:03 4C AC A9 A6 3A E8 D0 D4
0F39:03 4C 08 AF 20 92 0A 4C 93
0F41:D0 BB F0 10 20 6B A9 A5 7D
0F49:14 8D B1 02 A5 15 8D B2 7A
0F51:02 A9 80 2C A9 00 8D B0 C6
0F59:02 60 AD B0 02 29 BF 2C B1
0F61:A9 00 8D B0 02 A6 14 8E 72
0F69:FE 01 A6 15 8E FF 01 2C 10
0F71:B0 02 70 24 20 60 A5 86 0D
0F79:7A 84 7B 20 73 00 AA F0 49
0F81:D9 A2 FF 86 3A 90 03 4C 04
0F89:96 A4 AD B0 02 09 40 8D 1F
0F91:B0 02 20 79 00 4C 9C A4 33
0F99:10 C6 18 AD FE 01 6D B1 D7
0FA1:02 85 63 AD FF 01 6D B2 FA
0FA9:02 85 62 B0 B3 A2 90 38 03
0FB1:20 49 BC 20 DF BD 78 A2 55
0FB9:00 BD 00 01 C9 30 90 06 8D
0FC1:9D 77 02 E8 D0 F3 A9 20 25
0FC9:9D 77 02 E8 86 C6 58 D0 34
0FD1:A3 A9 08 2C A9 00 85 02 4A

The Source Code

```
0FD9:20 9E AD 24 0D 30 1F 20 2F
0FE1:EE 0B A5 02 D0 0D A6 15 2C
0FE9:F0 09 A6 9D E0 20 F0 03 DD
0FF1:4C 48 B2 20 8D 0D 20 9C 1E
0FF9:0E 20 0E E2 D0 DA 20 A6 EF
1001:B6 85 64 A0 00 C4 64 F0 41
1009:ED B1 22 20 BE 0B C8 D0 57
1011:F4 A0 01 A9 08 91 2B 20 8B
1019:33 A5 A5 22 A4 23 65 2D BC
1021:85 2D 90 01 C8 84 2E 4C 72
1029:5C A6 20 9E AD 20 A3 B6 FA
1031:20 BD FF 20 00 E2 A9 09 BA
1039:A0 00 20 BA FF 20 C1 E1 3F
1041:A2 09 20 1E E1 20 12 E1 70
1049:A5 90 29 02 F0 05 A9 04 98
1051:4C 3B A4 20 12 E1 20 CC 22
1059:FF A5 7A 48 A5 7B 48 A2 05
1061:09 20 1E E1 20 12 E1 20 1D
1069:12 E1 AA F0 34 20 12 E1 97
1071:20 12 E1 A2 FF E8 20 12 82
1079:E1 9D 00 02 A8 D0 F6 20 A8
1081:CC FF A9 FF 85 7A A9 01 A7
1089:85 7B 20 73 00 20 C4 0C 9C
1091:20 79 00 AA F0 C9 C9 3A 47
1099:F0 F0 C9 3B F0 C1 4C 08 8A
10A1:AF 20 CC FF A9 09 20 C3 B0
10A9:FF 68 85 7B 68 85 7A 60 FA
10B1:20 B4 10 A9 00 4C BE 0B 65
10B9:20 9E AD 20 A3 B6 A8 A9 3C
10C1:00 C0 58 90 03 4C 71 A5 F7
10C9:99 00 02 88 30 04 B1 22 96
10D1:90 F6 A5 7A 48 A5 7B 48 6C
10D9:A9 FF 85 7A A9 01 85 7B FE
10E1:20 73 00 90 07 A9 14 A0 9F
10E9:11 4C 27 11 20 6B A9 20 BD
10F1:7C A5 A5 14 8D FE 01 A5 BF
10F9:15 8D FF 01 8C FC 01 8C FE
1101:FD 01 84 0B A0 00 B9 FC 19
1109:01 20 BE 0B C8 C4 0B 90 3C
1111:F5 B0 96 4D 49 53 53 49 89
1119:4E 47 20 4C 49 4E 45 20 2B
1121:4E 55 4D 42 45 D2 85 22 30
1129:84 23 4C 47 A4 66 0F 0E 3F
1131:08 06 0A B1 0C 6E 0A A9 0D
1139:37 85 01 A9 00 85 58 85 5F
1141:5A 85 5F A9 C0 85 59 85 CC
1149:5B A9 A0 85 60 20 BB A3 8E
1151:A2 09 BD 2E 11 9D 02 03 A7
1159:CA 10 F7 A9 4C 8D C4 B5 56
1161:8D 92 B0 8D 43 B1 8D A5 7F
1169:A9 A9 0D 8D 4E B5 8D 55 FE
```

Chapter 3

1171:B1 A9 08 8D 0F B6 8D 89 A8
1179:B1 A9 EA 8D 10 B6 A9 60 24
1181:8D 69 B1 A9 8C A0 0B 8D 20
1189:44 B1 8C 45 B1 A9 E2 A0 BA
1191:0A 8D 93 B0 8C 94 B0 A9 5B
1199:A2 A0 0B 8D C5 B5 8C C6 54
11A1:B5 A9 D7 A0 0C 8D 2C A0 9D
11A9:8C 2D A0 A9 22 A0 0F 8D 4B
11B1:A6 A9 8C A7 A9 A9 C4 A0 BB
11B9:0C 8D 49 A9 8C 4A A9 A9 93
11C1:FB A0 F0 8D 20 D0 8C 21 7F
11C9:D0 A9 1B A0 12 8D 81 02 F7
11D1:8C 82 02 20 BF E3 A9 E4 E2
11D9:A0 11 20 1E AB 20 30 E4 99
11E1:4C 9D E3 93 08 0E 9B 11 08
11E9:20 20 2A 20 C1 D3 D3 C5 36
11F1:CD C2 CC C5 D2 20 56 33 98
11F9:20 2A 20 20 20 42 59 20 99
1201:D9 56 45 53 20 C8 41 4E 7A
1209:20 20 31 39 38 35 0D 0D BC
1211:20 20 20 20 20 20 20 20 35
1219:20 00 00 00 00 00 00 00 4D

The Source Code

The source code for the seven games in *COMPUTE!'s More Machine Language Games for the Commodore 64* is listed in *PAL* format (the *PAL* assembler, from Pro-Line Software).

PAL uses several pseudo-ops, such as *.ASC* and *.WORD*, that aren't found in most other assemblers. Most pseudo-ops, however, are in standard MOS 6502 notation: **=* updates the program counter (some assemblers use *.ORG* instead); *.BYT* or *.BYTE* assembles a list of numbers (some assemblers use *.DB* or *.DW*); *.WOR* or *.WORD* assembles a list of addresses into low-byte/high-byte format; *.ASC* is used to assemble an ASCII character string (many assemblers use *.BYTE* for this also); the pseudo-op *<* extracts the low byte of a 16-bit expression, while *>* extracts the high byte of a 16-bit expression (some assemblers reverse the use of *<* and *>*, and others use *\$255* and */256* for the same effect); and *=* is used to assign an expression to a label (some assemblers use *.EQU*).

You'll find comments in almost every game's source code, though some are more heavily commented than others. You needn't type them in (if that's what you're using the source code for), but if you do, remember to precede each comment with a semicolon (;).

Since all source code was written with or translated to *PAL*, it's not surprising that the listings which follow are also in *PAL* format. The listings aren't source code in the strictest sense, but rather source code that has been assembled by *PAL*. Your assembler may display things differently.

Here's a typical line from the listing for "Space Arena" (the boldface headings indicate each column's content). Note that pseudo-ops like *.ASC* and *.BYT* don't have a final operand, but instead list the ASCII string or a series of numbers.

Address	Opcode	Operand	Label	Mnemonic	Operand
C028	B9	36 CF	NWCHR	LDA	CHARAC,Y

Chapter 3

Notes on the Code

As mentioned above, the listings that follow show the *assembled* source code, not the raw source code you would actually enter for your assembler. Thus, you can't type the listings in the format shown. To enter the code so that you can assemble it yourself, you'll need to know the source code format your assembler expects. For example, for Commodore's *Macro Assembler Development System* you would type in just the labels, mnemonics, and operands (the three rightmost columns of the listings). *PAL*, on the other hand, requires a line number (just as BASIC does) before every line, but allows you to use the colon to combine more than one instruction per line.

You'll come across some odd-looking characters in several of the games' listings. These characters, limited to the .ASC pseudo-op, are found only within quotation marks (""). They're not meant to be typed in as shown, but instead represent the key(s) you should press at that point.

Words within braces are the most common such characters: {DOWN} means to press the cursor-down key; {5 LEFT} means to press the cursor-left key five times.

To indicate that a key should be *shifted* (pressed while you're holding down the SHIFT key), the character is underlined. For example, A means hold down the SHIFT key and press A. You may see strange characters on your screen, but that's to be expected. If you find a pair of braces enclosing a number followed by an underlined key—for instance, {8 A}—type the key as many times as indicated (in the example, enter eight SHIFTEd A's).

If a key is enclosed in special brackets, [>], hold down the Commodore key (at the lower left corner of the keyboard) and press the indicated character.

Refer to the following table.

The Source Code

When You Read:

	Press:	See:
{CLR}	SHIFT CLR/HOME	
{HOME}	CLR/HOME	
{UP}	SHIFT ↑ CRSR ↓	
{DOWN}	↑ CRSR ↓	
{LEFT}	SHIFT ← CRSR →	
{RIGHT}	← CRSR →	
{RVS}	CTRL 9	
{OFF}	CTRL 0	
{BLK}	CTRL 1	
{WHT}	CTRL 2	
{RED}	CTRL 3	
{CYN}	CTRL 4	
{PUR}	CTRL 5	
{GRN}	CTRL 6	
{BLU}	CTRL 7	
{YEL}	CTRL 8	

When You Read:

	Press:	See:
⌈ 1 ⌋	COMMODORE 1	
⌈ 2 ⌋	COMMODORE 2	
⌈ 3 ⌋	COMMODORE 3	
⌈ 4 ⌋	COMMODORE 4	
⌈ 5 ⌋	COMMODORE 5	
⌈ 6 ⌋	COMMODORE 6	
⌈ 7 ⌋	COMMODORE 7	
⌈ 8 ⌋	COMMODORE 8	
{ F1 }	f1	
{ F2 }	SHIFT f1	
{ F3 }	f3	
{ F4 }	SHIFT f3	
{ F5 }	f5	
{ F6 }	SHIFT f5	
{ F7 }	f7	
{ F8 }	SHIFT f7	
←		
↑	SHIFT	



Chapter 3

```

C0C0 99 00 D9          STA  $D900,Y
C0C3 99 00 DA          STA  $DA00,Y
C0C6 99 00 DB          STA  $DB00,Y
C0C9 88                 DEY
C0CA D0 D7            BNE  L7
C0CC 20 FF C6          JSR  DSCOR
C0CF A0 26             LDY  #38
C0D1 B9 66 CF          LDA  TPCOLR,Y
C0D4 99 28 D8          STA  $D828,Y
C0D7 88                 DEY
C0D8 D0 F7            BNE  UPCOL
C0DA A9 00             LDA  #0
C0DC 8D 20 D0          STA  $D020
C0DF 8D 21 D0          STA  $D021
C0E2 AD 36 C7          LDA  CLK
C0E5 09 30             ORA  #$30
C0E7 8D 5E 07          STA  1886
C0EA AD 37 C7          LDA  SPD
C0ED 09 30             ORA  #$30
C0EF 8D 0E 07          STA  1806
;
;choose speed and time limit
C0F2 20 E4 FF          JSR  $FFE4
C0F5 C9 85             CMP  #133
C0F7 F0 36             BEQ  START
C0F9 C9 86             CMP  #134
C0FB F0 1B             BEQ  SPEED
C0FD C9 87             CMP  #135
C0FF D0 F1            BNE  GETIN
C101 EE 36 C7          INC  CLK
C104 AD 36 C7          LDA  CLK
C107 C9 06             CMP  #6
C109 D0 05            BNE  SHTIM
C10B A9 01             LDA  #1
C10D 8D 36 C7          STA  CLK
C110 09 30             ORA  #$30
C112 8D 5E 07          STA  1886
C115 4C F2 C0          JMP  GETIN
C118 EE 37 C7          INC  SPD
C11B AD 37 C7          LDA  SPD
C11E C9 04             CMP  #4
C120 D0 05            BNE  SHSPD
C122 A9 01             LDA  #1
C124 8D 37 C7          STA  SPD
C127 09 30             ORA  #$30
C129 8D 0E 07          STA  1806
C12C 4C F2 C0          JMP  GETIN
;
C12F A0 03             START  LDY  #3
C131 A9 00             LDA  #0
C133 99 32 C7          ZSCR  STA  SCRT,Y
C136 88                 DEY
C137 10 FA            BPL  ZSCR
C139 20 FF C6          JSR  DSCOR
C13C A9 00             LDA  #0
C13E 8D 08 DC          STA  $D008
C141 8D 09 DC          STA  $D009
C144 8D 0A DC          STA  $D00A
;
;erase options
C147 A9 A0             LDA  #160
C149 85 FD            STA  TEMP1
C14B A9 04             LDA  #4
C14D 85 FE            STA  TEMP2
C14F A2 13            LDY  #19
C151 A0 26            CLRLN LDY  #38
C153 B1 FD            LDA  (TEMP1),Y
C155 30 04            BMI  SKPCH
C157 A9 20            LDA  #32
C159 91 FD            STA  (TEMP1),Y
C15B 88                 SKPCH DEY
C15C D0 F5            BNE  L8
C15E A5 FD            LDA  L8
;
C160 18                 CLC
C161 69 28            ADC  #40
C163 85 FD            STA  TEMP1
C165 90 02            BCC  B1
C167 E6 FE            INC  TEMP2
C169 CA                DEX
C16A D0 E5            BNE  CLRLN
;
C16C A0 02            LDY  #2
C16E B9 00 00 L9      LDA  $0000,Y
C171 99 00 08          STA  $0800,Y
C174 C8                INY
C175 D0 F7            BNE  L9
C177 A9 00            LDA  #0
C179 8D 0E DC          STA  $D00E
;
;set initial sprite conditions
C17C A2 40             SET  LDX  #$40
C17E BD A8 C7 L10     LDA  INITT,X
C181 95 10            STA  XCOORD,X
C183 CA                DEX
C184 10 F8            BPL  L10
C186 A9 84            LDA  #132
C188 8D F8 07          STA  2040
C18B A9 8C            LDA  #140
C18D 8D F9 07          STA  2041
C190 A9 00            LDA  #0
C192 85 47            STA  FRSCNT
C194 85 4E            STA  FRSCN2
;
C196 A0 18            LDY  #24
C198 88                 DEY
C199 99 00 D4          SOUND STA  $D400,Y
C19C D0 FA            BNE  SOUND
C19E A9 0F            LDA  #15
C1A0 8D 18 D4          STA  $D418
C1A3 A9 80            LDA  #128
C1A5 8D 14 D4          STA  $D414
C1A8 A9 81            LDA  #129
C1AA 8D 12 D4          STA  $D412
;
C1AD A0 07            LDY  #7
C1AF A9 0C            LDA  #12
C1B1 99 27 D0          COLOR STA  $D027,Y
C1B4 88                 DEY
C1B5 10 FA            BPL  COLOR
C1B7 A9 0E            LDA  #14
C1B9 8D 27 D0          STA  $D027
C1BC 8D 2D D0          STA  $D02D
C1BF A9 05            LDA  #5
C1C1 8D 28 D0          STA  $D028
C1C4 8D 2E D0          STA  $D02E
C1C7 A9 3C            LDA  #%00111100
C1C9 8D 1C D0          STA  $D01C
C1CC A9 0B            LDA  #11
C1CE 8D 25 D0          STA  $D025
C1D1 A9 0F            LDA  #15
C1D3 8D 26 D0          STA  $D026
;
;set timer by speed
C1D6 AD 37 C7          LDA  SPD
C1D9 0A                ASL  A
C1DA 0A                ASL  A
C1DB 0A                ASL  A
C1DC 0A                ASL  A
C1DD 85 02            STA  TEMP
C1DF A9 60            LDA  #96
C1E1 38                SEC
C1E2 E5 02            SBC  TEMP
C1E4 8D 07 DC          STA  $D007
C1E7 8D 06 DC          STA  $D006
C1EA A9 09            LDA  #9
C1EC 8D 0F DC          STA  $D00F

```


The Source Code

```

C1EF A9 3F          LDA # $3F
C1F1 8D 15 DO      STA $D015

C1F4 A9 01 MAIN    LDA #1
C1F6 2C OF DC TIMER BIT $DC0F
C1F9 DO FB        BNE TIMER
C1FB A9 09        LDA #9
C1FD 8D OF DC      STA $DC0F

;check for sprite collisions
C200 AD 1E DO      LDA $D01E
C203 85 45        STA CLLREG
C205 29 C3        AND #%11000011
C207 DO 03        BNE COLL
C209 4C DC C2     JMP DCOL

C20C 10 51 COLL   BPL NMIS2
C20E A5 45        LDA CLLREG
C210 29 01        AND #1
C212 FO 35        BEQ NMTS
C214 A2 00        LDY #0
C216 A0 0E        LDY #14
C218 20 5A C4     JSR CKCOL
C21B DO 2C        BNE NMTS

;missile 2 hit ship 1
C21D A9 01        LDA #1
C21F 20 EE C6     JSR SCORE

;add half of velocity to ship
C222 A2 01        LDY #1
C224 B5 3E DFLC  LDA XYSPD+14,X
C226 30 04        BMI NEGS
C228 4A           LSR A
C229 4C 2E C2     JMP INFL
C22C 38 NEGS     SEC
C22D 6A           ROR A
C22E 18 INFL    CLC
C22F 75 30        ADC XYSPD,X
C231 10 08        BPL MVF
C233 C9 C0        CMP #192
C235 B0 0A        BCS NDFLC
C237 A9 C0        LDA #192
C239 DO 06        BNE NDFLC
C23B C9 41        CMP #65
C23D 90 02        BCC NDFLC
C23F A9 40        LDA #64
C241 95 30 NDFLC STA XYSPD,X
C243 CA           DEX
C244 FO DE        BEQ DFLC
C246 4C 50 C2     JMP DMIS2

;check missile 2 to asteroid
C249 A2 0E NMTS  LDY #14
C24B 20 D7 C4     JSR ASTCOL
C24E DO 0F        BNE NMIS2
C250 AD 15 DO DMIS2 LDA $D015
C253 29 7F        AND # $7F
C255 8D 15 DO    STA $D015
C258 A0 00        LDY #0
C25A 84 1F        STY XCOOR+15
C25C 20 28 C5     JSR HTSND

C25F A5 45 NMIS2  LDA CLLREG
C261 29 40        AND #64
C263 FO 53        BEQ NMIS1
C265 A5 45        LDA CLLREG
C267 29 02        AND #2
C269 FO 35        BEQ NMTS1
C26B A2 02        LDY #2
C26D A0 0C        LDY #12
C26F 20 5A C4     JSR CKCOL
C272 DO 2C        BNE NMTS1

;missile 1 hit ship 2
C274 A9 01        LDA #1
C276 20 EE C6     JSR SCORE
C279 A2 01        LDY #1
C27B B5 3C DFLC  LDA XYSPD+12,X
C27D 30 04        BMI NEGS1
C27F 4A           LSR A
C280 4C 85 C2     JMP INFL1
C283 38 NEGS1   SEC
C284 6A           ROR A
C285 18 INFL1  CLC
C286 75 32        ADC XYSPD+2,X
C288 10 08        BPL MVF1
C28A C9 C0        CMP #192
C28C B0 0A        BCS NDFLC1
C28E A9 C0        LDA #192
C290 DO 06        BNE NDFLC1
C292 C9 41        CMP #65
C294 90 02        BCC NDFLC1
C296 A9 40        LDA #64
C298 95 32 NDFLC1 STA XYSPD+2,X
C29A CA           DEX
C29B FO DE        BEQ DFLC1
C29D 4C A7 C2     JMP DMIS1

;check missile 1 to asteroid
C2A0 A2 0C NMTS1 LDY #12
C2A2 20 D7 C4     JSR ASTCOL
C2A5 DO 11        BNE NMIS1
C2A7 AD 15 DO DMIS1 LDA $D015
C2A9 29 BF        AND # $BF
C2AB 8D 15 DO    STA $D015
C2AD C9 00        LDA #0
C2B1 85 1D        STA XCOOR+13
C2B3 A0 07        LDY #7
C2B5 20 28 C5     JSR HTSND

C2B8 A5 45 NMIS1  LDA CLLREG
C2BA 29 02        AND #2
C2BC FO 0C        BEQ NSHP2
C2BE A2 02        LDY #2
C2C0 20 D7 C4     JSR ASTCOL
C2C3 DO 05        BNE NSHP2

;ship 2 hit asteroid
C2C5 A0 01        LDY #1
C2C7 4C 5E C5     JMP KILL
C2CA A5 45 NSHP2 LDA CLLREG
C2CC 29 01        AND #1
C2CE FO 0C        BEQ DCOL
C2D0 A2 00        LDY #0
C2D2 20 D7 C4     JSR ASTCOL
C2D5 DO 05        BNE DCOL

;ship 1 hit asteroid
C2D7 A0 00        LDY #0
C2D9 4C 5E C5     JMP KILL

C2DC 20 0B C6 DCOL JSR MOVE
C2DF C6 40        DEC JOYCNT
C2E1 DO 7A        BNE NJOY
C2E3 A9 07 JOY   LDA #7
C2E5 85 40        STA JOYCNT
C2E7 A2 01        LDY #1

;ship control
C2E9 BD 00 DC OJOY LDA $DC00,X
C2EC 29 04        AND #4
C2EE DO 05        BNE NLEFT
C2F0 D6 41        DEC SHPDIR,X
C2F2 4C FE C2     JMP DIREC
C2F5 BD 00 DC NLEFT LDA $DC00,X
C2F8 29 08        AND #8
C2FA DO 00        BNE NRGHT

```

Chapter 3

```

C2FC F6 41          INC SHPDIR,X
C2FE B5 41          LDA SHPDIR,X
C300 29 0F          AND #15
C302 95 41          STA SHPDIR,X
C304 09 80          ORA #128
C306 9D F8 07      STA 2040,X
C309 BD 00 DC       LDA $DC00,X
C30C 29 01          AND #1
C30E D0 28          BNE NTHRU
C310 B4 41          LDY SHPDIR,X
C312 8A             TXA
C313 0A             ASL A
C314 AA             TAX
C315 B5 30          LDA XYSPD,X
C317 18             CLC
C318 79 75 C7      ADC THXT,Y
C31B C9 41          CMP #65
C31D 90 04          BCC NVTX
C31F C9 C0          CMP #192
C321 90 02          BCC VTX
C323 95 30          STA XYSPD,X
C325 B5 31          LDA XYSPD+1,X
C327 18             CLC
C328 79 71 C7      ADC THYT,Y
C32B C9 41          CMP #65
C32D 90 04          BCC NVTY
C32F C9 C0          CMP #192
C331 90 02          BCC VTY
C333 95 31          STA XYSPD+1,X
C335 8A             TXA
C336 4A             LSR A
C337 AA             TAX
C338 CA             NTHRU
C339 FO AE          BEQ OJOY

;thrust sound
C33B AD 00 DC          LDA $DC00
C33E 29 01          AND #1
C340 85 02          STA TEMP
C342 AD 01 DC          LDA $DC01
C345 29 01          AND #1
C347 FO 04          BEQ MKSND
C349 A4 02          LDY TEMP
C34B D0 0B          BNE NOSND
C34D 18             CLC
C34E 69 10          ADC #16
C350 E5 02          SBC TEMP
C352 8D 0F D4        STA $D40F
C355 4C 5D C3        JMP NJOY
C358 A9 00           LDA #0
C35A 8D 0F D4        STA $D40F

C35D A5 43           NJOY LDA MSL2CN
C35F D0 4E          BNE GO1
C361 AD 15 D0        LDA $D015
C364 29 BF          AND #8F
C366 8D 15 D0        STA $D015
C369 AD 00 DC          LDA $DC00
C36C 29 10          AND #16
C36E D0 41          BNE NSHOT1

;new shot 1 fired
C370 A0 00           LDY #0
C372 20 43 C5        JSR FR SND
C375 A9 80           LDA #128
C377 85 43           STA MSL2CN
C379 A5 10           LDA XCOOR
C37B 18             CLC
C37C 69 20          ADC #32
C37E 85 1C          STA XCOOR+12
C380 A5 11          LDA XCOOR+1
C382 69 01          ADC #1
C384 85 1D          STA XCOOR+13

C386 A5 20           LDA YCOOR
C388 18             CLC
C389 69 40          ADC #64
C38B 85 2C          STA YCOOR+12
C38D A5 21          LDA YCOOR+1
C38F 69 01          ADC #1
C391 85 2D          STA YCOOR+13
C393 A4 41          LDY SHPDIR
C395 B9 75 C7       LDA THXT,Y
C398 0A             ASL A
C399 0A             ASL A
C39A 0A             ASL A
C39B 0A             ASL A
C39C 85 3C          STA XYSPD+12
C39E B9 71 C7       LDA THYT,Y
C3A1 0A             ASL A
C3A2 0A             ASL A
C3A3 0A             ASL A
C3A4 0A             ASL A
C3A5 85 3D          STA XYSPD+13
C3A7 AD 15 D0        LDA $D015
C3AA 09 40          ORA #540
C3AC 8D 15 D0        STA $D015
C3AF C6 43 GOI1     DEC MSL2CN

C3B1 A5 44           NSHOT1 LDA MSL2CN
C3B3 D0 4E          BNE GO2
C3B5 AD 15 D0        LDA $D015
C3B8 29 7F          AND #7F
C3BA 8D 15 D0        STA $D015
C3BD AD 01 DC          LDA $DC01
C3C0 29 10          AND #16
C3C2 D0 41          BNE NSHOT2

;new shot 2 fired
C3C4 A0 07           LDY #7
C3C6 20 43 C5        JSR FR SND
C3C9 A9 80           LDA #128
C3CB 85 44           STA MSL2CN
C3CD A5 12           LDA XCOOR+2
C3CF 18             CLC
C3D0 69 50          ADC #80
C3D2 85 1E          STA XCOOR+14
C3D4 A5 13           LDA XCOOR+3
C3D6 69 01          ADC #1
C3D8 85 1F          STA XCOOR+15
C3DA A5 22           LDA YCOOR+2
C3DC 18             CLC
C3DD 69 40          ADC #64
C3DF 85 2E          STA XCOOR+14
C3E1 A5 23           LDA YCOOR+3
C3E3 69 01          ADC #1
C3E5 85 2F          STA YCOOR+15
C3E7 A4 42           LDY SHPDIR
C3E9 B9 75 C7       LDA THXT,Y
C3EC 0A             ASL A
C3ED 0A             ASL A
C3EE 0A             ASL A
C3EF 0A             ASL A
C3F0 85 3E          STA XYSPD+14
C3F2 B9 71 C7       LDA THYT,Y
C3F5 0A             ASL A
C3F6 0A             ASL A
C3F7 0A             ASL A
C3F8 0A             ASL A
C3F9 85 3F          STA XYSPD+15
C3FB AD 15 D0        LDA $D015
C3FE 09 80          ORA #80
C400 8D 15 D0        STA $D015
C403 C6 44 GOI2     DEC MSL2CN

;display time limit
C405 AE 0A DC NSHOT2 LDX $DC0A

```

The Source Code

```

C408 AD 09 DC          LDA  $DC09
C40B FO 09          BEQ  SIXTY
C40D E8             INX
C40E A9 60          LDA  #$60
C410 F8             SED
C411 38             SEC
C412 ED 09 DC      SBC  $DC09
C415 D8             CLD
C416 85 02   SIXTY STA  TEMP
C418 4A             LSR  A
C419 4A             LSR  A
C41A 4A             LSR  A
C41B 4A             LSR  A
C41C 09 30          ORA  #$30
C41E 8D 3C 04      STA  $043C
C421 A5 02          LDA  TEMP
C423 29 0F          AND  #$0F
C425 09 30          ORA  #$30
C427 8D 3D 04      STA  $043D
C42A 86 02          STX  TEMP
C42C AD 36 C7      LDA  CLK
C42F 38             SEC
C430 E5 02          SBC  TEMP
C432 09 30          ORA  #$30
C434 8D 3A 04      STA  $043A
C437 AD 0A DC      LDA  $DCCOA
C43A CD 36 C7      CMP  CLK
C43D FO 03          BEQ  TIMUP
C43F 4C F4 C1      JMP  MAIN

C442 A9 00   TIMUP  LDA  #0
C444 8D 15 D0      STA  $D015
C447 A0 02          LDY  #2
C449 B9 00 08 FZP  LDA  $0800,Y
C44C 99 00 00      STA  $0000,Y
C44F C8             INY
C450 D0 F7          BNE  FZP
C452 A9 01          LDA  #1
C454 8D 0E DC      STA  $DC0E
C457 4C 8C C0      JMP  TITLE

; check coord for collision
; between sprite x and sprite y

C45A B5 10   CKCOL  LDA  XCOORD,X
C45C 38             SEC
C45D F9 10 00      SBC  XCOORD,Y
C460 85 FD          STA  TEMP1
C462 B5 11          LDA  XCOORD+1,X
C464 F9 11 00      SBC  XCOORD+1,Y
C467 85 FE          STA  TEMP2
C469 90 0F          BCC  OWAY
C46B A5 FD          LDA  TEMP1
C46D F9 51 C7      SBC  COLXT,Y
C470 A5 FE          LDA  TEMP2
C472 F9 52 C7      SBC  COLXT+1,Y
C475 B0 5D          BCS  NCOL
C477 4C 95 C4      JMP  CKY
C47A B9 10 00  OWAY LDA  XCOORD,Y
C47D 38             SEC
C47E F5 10          SBC  XCOORD,X
C480 85 FD          STA  TEMP1
C482 B9 11 00      LDA  XCOORD+1,Y
C485 F5 11          SBC  XCOORD+1,X
C487 85 FE          STA  TEMP2
C489 A5 FD          LDA  TEMP1
C48B FD 51 C7      SBC  COLXT,X
C48E A5 FE          LDA  TEMP2
C490 FD 52 C7      SBC  COLXT+1,X
C493 B0 3F          BCS  NCOL

C495 B5 20   CKY   LDA  YCOORD,X
C497 38             SEC
C498 F9 20 00      LDA  YCOORD,Y
C49B 85 FD          STA  TEMP1
C49D B5 21          LDA  YCOORD+1,X
C49F 38             SEC
C4A0 F9 21 00      SBC  YCOORD+1,Y
C4A3 85 FE          STA  TEMP2
C4A5 90 0F          BCC  OWAY
C4A7 A5 FD          LDA  TEMP1
C4A9 F9 61 C7      SBC  COLTY,Y
C4AC A5 FE          LDA  TEMP2
C4AE F9 62 C7      SBC  COLTY+1,Y
C4B1 B0 21          BCS  NCOL
C4B3 4C D1 C4      JMP  COL
C4B6 B9 20 00  OWAY LDA  YCOORD,Y
C4B9 38             SEC
C4BA F5 20          SBC  YCOORD,X
C4BC 85 FD          STA  TEMP1
C4BE B9 21 00      LDA  YCOORD+1,Y
C4C1 F5 21          SBC  YCOORD+1,X
C4C3 85 FE          STA  TEMP2
C4C5 A5 FD          LDA  TEMP1
C4C7 FD 61 C7      SBC  COLTY,X
C4CA A5 FE          LDA  TEMP2
C4CC FD 62 C7      SBC  COLTY+1,X
C4CF B0 03          BCS  NCOL

C4D1 A9 00   COL   LDA  #0
C4D3 60           RTS
C4D4 A9 01   NCOL  LDA  #1
C4D6 60           RTS

; check collision of sprite x
; and each asteroid

C4D7 A5 45   ASTCOL LDA  CLLREG
C4D9 0A          ASL  A
C4DA 0A          ASL  A
C4DB 85 02      STA  TEMP
C4DD A0 0A      LDY  #10
C4DF 06 02      ASL  TEMP
C4E1 90 05      BCC  NHIT
C4E3 20 5A C4   JSR  NHITCOL
C4E6 F0 09      BEQ  HIT
C4E8 88          DEY
C4E9 88          DEY
C4EA C0 02      CPY  #2
C4EC D0 F1      BNE  HITL
C4EE A9 01      LDA  #1
C4F0 60          RTS

; push asteroid
C4F1 A9 01   HIT   LDA  #1
C4F3 85 02      STA  TEMP
C4F5 C8          INY
C4F6 E8          INX
C4F7 B5 30   DFLCA LDA  XYSPD,X
C4F9 30 07      BMI  NEGA
C4FB 4A          LSR  A
C4FC 4A          LSR  A
C4FD 4A          LSR  A
C4FE 4A          LSR  A
C4FF 4C 08 C5   JMP  INFLA
C502 4A          LSR  A
C503 4A          LSR  A
C504 4A          LSR  A
C505 4A          LSR  A
C506 09 F0      ORA  #$F0
C508 18          CLC
C509 79 30 00   ADC  XYSPD,Y
C50C 10 08      BPL  MVFA
C50E C9 C0      CMP  #192
C510 B0 0A      BCS  NDFLCA
C512 A9 C0      LDA  #192

```

Chapter 3

```

C514 D0 06          BNE  NDFLCA
C516 C9 41          CMP   #65
C518 90 02          BCC  NDFLCA
C51A A9 40          LDA  #64
C51C 99 30 00      NDFLCA STA  XYSPD,Y
C51F CA            DEY
C520 88            DEY
C521 C6 02          DEC  TEMP
C523 F0 D2          BEQ  DFLCA
C525 A9 00          LDA  #0
C527 60            RTS

;make hit sound -register y
;
C528 A9 00          HTSND   LDA  #0
C52A 99 06 D4      STA  $D406,Y
C52D 99 47 00      STA  FRSCNT,Y
C530 99 04 D4      STA  $D404,Y
C533 A9 0A          LDA  #10
C535 99 01 D4      STA  $D401,Y
C538 A9 35          LDA  #$35
C53A 99 05 D4      STA  $D405,Y
C53D A9 21          LDA  #33
C53F 99 04 D4      STA  $D404,Y
C542 60            RTS

;make fire sound -register y
;
C543 A9 00          FRSND   LDA  #0
C545 99 06 D4      STA  $D406,Y
C548 99 04 D4      STA  $D404,Y
C54B A9 28          LDA  #40
C54D 99 47 00      STA  FRSCNT,Y
C550 99 01 D4      STA  $D401,Y
C553 A9 09          LDA  #9
C555 99 05 D4      STA  $D405,Y
C558 A9 11          LDA  #17
C55A 99 04 D4      STA  $D404,Y
C55D 60            RTS

;
C55E A9 90          KILL    LDA  #144
C560 99 F8 07      STA  2040,Y
C563 B9 85 C7      LDA  CRWDT,Y
C566 8D 05 C6      STA  WCRWD+1

;set which crowd cheers
C569 E8            INX
C56A A9 01          LDA  #1
C56C 85 48          STA  CRWDCN
C56E A9 20          LDA  #$20
C570 20 EE C6      JSR  SCORE

;
;explosion
C573 A0 07          LDY  #7
C575 A9 00          LDA  #0
C577 88            DEY
C578 99 0E D4      CTSND   STA  $D40E,Y
C57B D0 FA          BNE  CTSND
C57D A9 0A          LDA  #10
C57F 8D 13 D4      STA  $D413
C582 A9 08          LDA  #8
C584 8D 0F D4      STA  $D40F
C587 A9 81          LDA  #129
C589 8D 12 D4      STA  $D412
C58C A9 1E          LDA  #30
C58E 85 46          STA  TIMCNT
C590 20 0B C6      JSR  MOVE
C593 A9 01          LDA  #1
C595 2C 0F DC      BIT  $DC0F
C598 D0 FB          BNE  TMR
C59A A9 09          LDA  #9
C59C 8D 0F DC      STA  $DC0F
C59F C6 46          DEC  TIMCNT

C5A1 D0 ED          BNE  EXPLO
;
;cheering sound
C5A3 A9 80          LDA  #128
C5A5 8D 12 D4      STA  $D412
C5A8 A9 8C          LDA  #$8C
C5AA 8D 13 D4      STA  $D413
C5AD A9 11          LDA  #17
C5AF 8D 0F D4      STA  $D40F
C5B2 A9 A0          LDA  #160
C5B4 8D 16 D4      STA  $D416
C5B7 A9 84          LDA  #$84
C5B9 8D 17 D4      STA  $D417
C5BC A9 1F          LDA  #31
C5BE 8D 18 D4      STA  $D418
C5C1 A9 81          LDA  #129
C5C3 8D 12 D4      STA  $D412
C5C6 A9 3C          LDA  #%00111100
C5C8 8D 15 D0      STA  $D015
C5CB A9 46          LDA  #70
C5CD 85 46          STA  TIMCNT
C5CF 20 0B C6      JSR  MOVE
C5D2 20 F2 C5      JSR  CROWD
C5D5 A9 01          LDA  #1
C5D7 2C 0F DC      BIT  $DC0F
C5DA D0 FB          BNE  TMR2
C5DC A9 09          LDA  #9
C5DE 8D 0F DC      STA  $DC0F
C5E1 C6 46          DEC  TIMCNT
C5E3 D0 EA          BNE  CHEER
C5E5 A9 00          LDA  #0
C5E7 8D 17 D4      STA  $D417
C5EA A9 0F          LDA  #15
C5EC 8D 18 D4      STA  $D418
C5EF 4C 7C C1      JMP  SET

;switch characters on crowd
C5F2 C6 48          CROWD   DEC  CRWDCN
C5F4 D0 14          BNE  NMCW
C5F6 A9 08          LDA  #8
C5F8 85 48          STA  CRWDCN
C5FA A0 0A          LDY  #10
C5FC AD 87 C7      LDA  CRWDT+2
C5FF 49 01          EOR  #1
C601 8D 87 C7      STA  CRWDT+2
C604 99 00 04      WCRWD   STA  $D400,Y
C607 88            DEY
C608 10 FA          BPL  WCRWD
C60A 60            RTS

;move sprites
C60B A2 0E          MOVE    LDX  #14

;add x speed to coord
C60D B5 30          POSI    LDA  XYSPD,X
C60F 10 1D          BPL  ADD
C611 C9 FE          CMP  #254
C613 B0 3E          BCS  B2
C615 75 10          ADC  XCOORD,X
C617 95 10          STA  XCOORD,X
C619 B5 11          LDA  XCOORD+1,X
C61B E9 00          SBC  #0
C61D 95 11          STA  XCOORD+1,X
C61F C9 05          CMP  #5
C621 90 28          BCC  REV
C623 D0 2E          BNE  B2
C625 B5 10          LDA  XCOORD,X
C627 C9 40          CMP  #64
C629 B0 28          BCS  B2
C62B 4C 4B C6      JMP  REV

```

The Source Code

```

C62E C9 03 ADD      CMP #3
C630 90 21 BCC B2
C632 18 CLC
C633 75 10 ADC XCOOR,X
C635 95 10 STA XCOOR,X
C637 B5 11 LDA XCOOR+1,X
C639 69 00 ADC #0
C63B 95 11 STA XCOOR+1,X
C63D DD 89 C7 CMP BNCXT+1,X
C640 90 11 BCC B2
C642 D0 07 BNE REV
C644 B5 10 LDA XCOOR,X
C646 DD 88 C7 CMP BNCXT,X
C649 90 08 BCC B2

;hit boundary - reverse x direction
C64B B5 30 REV LDA XYSPD,X
C64D 49 FF EOR #$FF
C64F A8 TAY
C650 C8 INY
C651 94 30 STY XYSPD,X

;divide x coord for position
C653 B5 10 B2 LDA XCOOR,X
C655 85 02 STA TEMP
C657 A0 08 LDY #8
C659 B5 11 LDA XCOOR+1,X
C65B C9 30 CMP #48
C65D 90 02 BCC HIBIT
C65F E9 30 SBC #48
C661 26 49 HIBIT ROL POSMSB
C663 06 02 L11 ASL TEMP
C665 2A ROL A
C666 C9 30 CMP #48
C668 90 04 BCC DVCT
C66A E9 30 SBC #48
C66C E6 02 INC TEMP
C66E 88 DVCT DEY
C66F D0 F2 BNE L11
C671 A5 02 LDA TEMP
C673 9D 00 D0 STA $D000,X

;add y speed to coord
C676 B5 31 LDA XYSPD+1,X
C678 10 1D BPL ADDY
C67A C9 FE CMP #254
C67C B0 3E BCS B3
C67E 75 20 ADC YCOOR,X
C680 95 20 STA YCOOR,X
C682 B5 21 LDA YCOOR+1,X
C684 E9 00 SBC #0
C686 95 21 STA YCOOR+1,X
C688 C9 11 CMP #17
C68A 90 28 BCC REVY
C68C D0 2E BNE B3
C68E B5 20 LDA YCOOR,X
C690 C9 80 CMP #128
C692 B0 28 BCS B3
C694 4C B4 C6 JMP REVY

C697 C9 03 ADDY CMP #3
C699 90 21 BCC B3
C69B 18 CLC
C69C 75 20 ADC YCOOR,X
C69E 95 20 STA YCOOR,X
C6A0 B5 21 LDA YCOOR+1,X
C6A2 69 00 ADC #0
C6A4 95 21 STA YCOOR+1,X
C6A6 DD 99 C7 CMP BNCYT+1,X
C6A9 90 11 BCC B3
C6AB D0 07 BNE REVY
C6AD B5 20 LDA YCOOR,X
C6AF DD 98 C7 CMP BNCYT,X

C6B2 90 08 BCC B3

;reverse y direction
C6B4 B5 31 REVY LDA XYSPD+1,X
C6B6 49 FF EOR #$FF
C6B8 A8 TAY
C6B9 C8 INY
C6BA 94 31 STY XYSPD+1,X

;divide y coord for position
C6BC B5 21 B3 LDA YCOOR+1,X
C6BE 85 02 STA TEMP
C6C0 B5 20 LDA YCOOR,X
C6C2 A0 06 LDY #6
C6C4 46 02 L12 LSR TEMP
C6C6 6A ROR A
C6C7 88 DEY
C6C8 D0 FA BNE L12
C6CA 9D 01 D0 STA $D001,X

C6CD CA DEX
C6CE CA DEX
C6CF 30 03 BMI NPOSI
C6D1 4C 0D C6 JMP POSI
C6D4 A5 49 NPOSI LDA POSMSB
C6D6 8D 10 D0 STA $D010

;update fire sound
C6D9 A4 47 LDY FRSCNT
C6DB F0 06 BEQ NFRS
C6DD 88 DEY
C6DE 84 47 STY FRSCNT
C6E0 8C 01 D4 STY $D401
C6E3 A4 4E NFRS LDY FRSCN2
C6E5 F0 06 BEQ NFRS2
C6E7 88 DEY
C6E8 84 4E STY FRSCN2
C6EA 8C 08 D4 STY $D408
C6ED 60 NFRS2 RTS

;add to score -player x
C6EE F8 SCORE SED
C6EF 18 CLC
C6F0 7D 32 C7 ADC SCRT,X
C6F3 9D 32 C7 STA SCRT,X
C6F6 BD 33 C7 LDA SCRT+1,X
C6F9 69 00 ADC #0
C6FB 9D 33 C7 STA SCRT+1,X
C6FE D8 CLD

;display score
C6FF A0 01 DSCOR LDY #1
C701 AD 34 C7 LDA SCRT+2
C704 85 FD STA TEMP1
C706 AD 35 C7 LDA SCRT+3
C709 85 FE STA TEMP2
C70B A2 04 NXDGT LDY #4
C70D A9 00 LDA #0
C70F 06 FD SFT ASL TEMP1
C711 26 FE ROL TEMP2
C713 2A ROL A
C714 CA DEX
C715 D0 F8 BNE SFT
C717 09 30 ORA #$30
C719 99 28 04 STA $0428,Y
C71C C8 INY
C71D C0 05 CPY #5
C71F D0 0C BNE SCOR2
C721 A0 23 LDY #35
C723 AD 32 C7 LDA SCRT
C726 85 FD STA TEMP1

```

Chapter 3

C728	AD 33 C7	LDA	SCRT+1	C7FD	40 40 72	.BYT	64,64,114,64,64,64,64,73
C72B	85 FE	STA	TEMP2				
C72D	C0 27 SCOR2	CPY	#39	C805	5D 30 30	.BYT	93,48,48,48,48,93,132,132
C72F	D0 DA	BNE	NXDGT				
C731	60	RTS		C80D	84 84 84	.BYT	132,132,132,132,132,132,132,132
C732	00 00 00 SCRT	.BYT	0,0,0,0	C815	84 5D 30	.BYT	132,93,48,58,48,48,93,132
C736	01 CLK	.BYT	1				
C737	01 SPD	.BYT	1	C81D	84 84 84	.BYT	132,132,132,132,132,132,132,132
	hum sound			C825	84 84 5D	.BYT	132,132,93,48,48,48,48,93
C738	00 06 00 SNDT	.BYT	0,6,0,0,17,15,0				
C73F	04 03 00	.BYT	4,3,0,0,21,15,0	C82D	6B 40 40	.BYT	107,64,64,64,64,113,64,64
C746	04 0C 00	.BYT	4,12,0,0,21,15,0				
C74D	00 00 00	.BYT	0,0,0,15	C835	40 40 40	.BYT	64,64,64,64,64,64,64,64
	width of sprites			C83D	40 71 40	.BYT	64,113,64,64,64,64,113,64
C751	30 03 30 COLXT	.WOR	816,816				
C755	80 04 80	.WOR	1152,1152	C845	40 40 40	.BYT	64,64,64,64,64,64,64,64
C759	80 04 80	.WOR	1152,1152				
C75D	FO 00 FO	.WOR	240,240	C84D	40 40 71	.BYT	64,64,113,64,64,64,64,115
	height of sprites			C855	5D 20 20	.BYT	93,32,32,32,32,32,32,32
C761	00 03 00 COLYT	.WOR	768,768				
C765	C0 04 C0	.WOR	1216,1216	C85D	20 20 20	.BYT	32,32,32,128,32,32,32,32
C769	C0 04 C0	.WOR	1216,1216				
C76D	00 01 00	.WOR	256,256	C865	20 20 20	.BYT	32,32,32,32,32,32,32,32
	accelerations for direction			C86D	20 20 20	.BYT	32,32,32,32,32,32,32,32
C771	FA FB FC THYT	.BYT	250,251,252,254				
C775	00 02 04 THXT	.BYT	0,2,4,5	C875	20 20 20	.BYT	32,32,32,32,32,32,32,32
C779	06 05 04	.BYT	6,5,4,2				
C77D	00 FE FC	.BYT	0,254,252,251	C87D	5D 20 20	.BYT	93,32,32,32,32,32,32,32
C781	FA FB FC	.BYT	250,251,252,254				
				C885	20 20 20	.BYT	32,32,32,32,32,32,32,32
	crowd position			C88D	20 20 20	.BYT	32,32,32,32,32,32,32,32
C785	3F 2E CRWDT	.BYT	63,46				
				C895	20 20 20	.BYT	32,32,32,32,32,32,32,32
	crowd character			C89D	20 20 20	.BYT	32,32,32,32,32,32,32,32
C787	85	.BYT	133				
	max x coord for sprite			C8A5	5D 20 20	.BYT	93,32,32,32,128,32,32,32
C788	FO 3C FO BNCXT	.WOR	15600,15600				
C78C	70 3B 70	.WOR	15216,15216	C8AD	20 13 20	.BYT	32,19,32,16,32,1,32,3
C790	70 3B 70	.WOR	15216,15216				
C794	30 3F 30	.WOR	16176,16176	C8B5	20 05 20	.BYT	32,5,32,32,32,1,32,18
	max y coord			C8BD	20 05 20	.BYT	32,5,32,14,32,1,32,3,2
C798	00 3B 00 BNCYT	.WOR	15104,15104				
C79C	00 39 00	.WOR	14592,14592	C8C5	20 20 20	.BYT	32,32,32,32,32,32,32,32
C7A0	00 39 00	.WOR	14592,14592				
C7A4	00 3D 00	.WOR	15616,15616	C8CD	5D 20 20	.BYT	93,32,32,32,32,32,32,32
	initial variable values			C8D5	80 63 63	.BYT	128,99,99,99,99,99,99,99
C7A8	90 06 D0 INITT	.WOR	1680,15312,8448,8448				
				C8DD	63 63 63	.BYT	99,99,99,99,99,99,99,99
C7B0	00 21 00	.WOR	8448,8448,0,0				
C7B8	C0 25 C0	.WOR	9664,9664,5696,8192	C8E5	63 63 63	.BYT	99,99,99,99,99,99,32,32
C7C0	C0 29 80	.WOR	10688,13184,0,0	C8ED	20 20 20	.BYT	32,32,32,129,32,32,32,93
C7C8	00 00 00	.BYT	0,0,0,0,0,0,0,0				
C7D0	00 00 00	.BYT	0,0,0,0,0,0,0,0	C8F5	5D 20 20	.BYT	93,32,32,32,32,32,32,32
C7D8	01 04 0C	.BYT	1,4,12,0,0				
				C8FD	20 20 20	.BYT	32,32,32,32,2,25,58,32
	title screen			C905	02 12 19	.BYT	2,18,25,1,14,32,6,9
C7DD	55 40 40 SCREEN	.BYT	85,64,64,64,64,114,64,64	C90D	0C 05 13	.BYT	12,5,19,32,32,32,32,32
C7E5	40 40 40	.BYT	64,114,64,64,64,64,64,64				
				C915	20 20 20	.BYT	32,32,32,32,32,32,32,32
C7ED	40 72 40	.BYT	64,114,64,64,64,64,114,64				
				C91D	5D 20 20	.BYT	93,32,32,32,32,32,129,32
C7F5	40 40 40	.BYT	64,64,64,64,64,64,64,64				

The Source Code

C925	20 20 20	.BYT	32,32,32,32,32,32,3	CA5D	5D 20 20	.BYT	93,32,32,32,32,32,3
			2,32				2,32
C92D	20 20 20	.BYT	32,32,32,32,32,32,3	CA65	20 20 20	.BYT	32,32,32,32,32,128,
			2,32				32,32
C935	20 20 20	.BYT	32,32,32,131,32,32,	CA6D	20 20 20	.BYT	32,32,32,32,32,32,3
			32,32				2,32
C93D	20 20 20	.BYT	32,32,32,32,32,32,3	CA75	20 20 20	.BYT	32,32,32,32,32,32,3
			2,93				2,32
C945	5D 20 20	.BYT	93,32,32,32,32,32,3	CA7D	20 20 20	.BYT	32,32,32,32,32,32,3
			2,32				2,93
C94D	20 20 20	.BYT	32,32,32,32,32,32,3	CA85	5D 20 20	.BYT	93,32,32,32,32,32,3
			2,32				2,32
C955	20 82 20	.BYT	32,130,32,32,32,32,	CA8D	20 20 20	.BYT	32,32,32,32,32,32,3
			32,32				2,32
C95D	20 20 20	.BYT	32,32,32,32,32,32,1	CA95	20 20 20	.BYT	32,32,32,32,32,32,1
			28,32				28,32
C965	80 20 60	.BYT	128,32,96,32,32,32,	CA9D	20 20 20	.BYT	32,32,32,32,32,32,3
			32,93				2,32
C96D	5D 20 20	.BYT	93,32,32,32,32,32,3	CAA5	20 20 20	.BYT	32,32,32,32,32,128,
			2,32				32,93
C975	20 20 20	.BYT	32,32,32,32,32,32,3	CAAD	5D 20 20	.BYT	93,32,32,129,32,32,
			2,32				32,32
C97D	20 20 20	.BYT	32,32,32,32,32,32,3	CAB5	20 20 20	.BYT	32,32,32,32,32,32,3
			2,32				2,32
C985	20 20 20	.BYT	32,32,32,128,32,32,	CABD	20 20 20	.BYT	32,32,32,32,32,129,
			32,32				32,32
C98D	20 20 20	.BYT	32,32,32,32,32,32,3	CAC5	20 20 20	.BYT	32,32,32,130,32,32,
			2,93				32,32
C995	5D 20 20	.BYT	93,32,32,32,128,32,	CACD	20 20 20	.BYT	32,32,32,32,32,32,1
			32,32				28,93
C99D	20 06 31	.BYT	32,6,49,32,45,32,19,	CAD5	5D 20 20	.BYT	93,32,32,32,32,32,3
			20				2,32
C9A5	01 12 14	.BYT	1,18,20,32,7,1,13,5	CADD	20 20 20	.BYT	32,32,32,32,32,32,1
C9AD	20 20 20	.BYT	32,32,32,32,32,32,3				9,16
			2,32				.BYT
C9B5	20 20 20	.BYT	32,32,32,32,32,32,3	CAE5	05 05 04	.BYT	5,5,4,32,61,32,53,32
			2,93	CAED	81 20 20	.BYT	129,32,32,32,32,128
							,32,32
C9BD	5D 20 20	.BYT	93,32,32,32,32,32,3	CAF5	81 20 20	.BYT	129,32,32,32,32,32,
			2,32				32,93
C9C5	20 20 81	.BYT	32,32,129,32,32,32,	CAF6D	5D 20 20	.BYT	93,32,32,32,32,32,3
			32,32				2,32
C9CD	20 20 20	.BYT	32,32,32,32,32,32,3	CB05	20 20 20	.BYT	32,32,32,32,32,32,3
			2,32				2,32
C9D5	20 20 20	.BYT	32,32,32,32,32,32,3	CB0D	20 20 20	.BYT	32,32,32,32,32,32,3
			2,32				2,32
C9DD	20 20 20	.BYT	32,32,32,32,32,32,3	CB15	20 20 20	.BYT	32,32,32,32,32,32,3
			2,93				2,32
C9E5	5D 20 20	.BYT	93,32,32,32,32,32,3	CB1D	20 20 20	.BYT	32,32,32,32,32,32,3
			2,32				2,93
C9ED	20 06 33	.BYT	32,6,51,32,45,32,3,8	CB25	5D 20 20	.BYT	93,32,32,32,32,32,3
C9F5	01 0E 07	.BYT	1,14,7,5,32,19,16,5				2,32
C9FD	05 04 20	.BYT	5,4,32,32,32,32,32,	CB2D	20 20 80	.BYT	32,32,128,32,32,32,
			32				32,20
CA05	80 83 20	.BYT	128,131,32,32,32,32	CB35	09 0D 05	.BYT	9,13,5,32,61,32,50,
			,32,93				58
CA0D	5D 20 83	.BYT	93,32,131,32,32,32,	CB3D	30 30 20	.BYT	48,48,32,32,128,32,
			32,32				32,32
CA15	20 20 80	.BYT	32,32,128,32,32,129	CB45	20 20 20	.BYT	32,32,32,32,32,32,3
			,32,32				2,93
CA1D	20 20 20	.BYT	32,32,32,32,32,32,3	CB4D	5D 20 83	.BYT	93,32,131,32,32,32,
			2,32				32,32
CA25	20 20 20	.BYT	32,32,32,32,32,128,	CB55	20 20 20	.BYT	32,32,32,32,32,32,3
			32,32				2,129
CA2D	20 20 20	.BYT	32,32,32,32,32,32,3	CB5D	20 20 20	.BYT	32,32,32,128,32,32,
			2,93				32,32
CA35	5D 20 81	.BYT	93,32,129,32,32,32,	CB65	20 20 20	.BYT	32,32,32,32,32,32,3
			32,32				2,32
CA3D	20 06 35	.BYT	32,6,53,32,45,32,3,8	CB6D	20 20 20	.BYT	32,32,32,32,32,32,3
CA45	01 0E 07	.BYT	1,14,7,5,32,20,9,13				2,93
CA4D	05 20 0C	.BYT	5,32,12,9,13,9,20,32	CB75	5D 20 20	.BYT	93,32,32,32,32,32,3
CA55	20 20 20	.BYT	32,32,32,32,32,32,3				2,32
			2,93	CB7D	20 20 20	.BYT	32,32,32,32,32,32,3
							2,32

Chapter 3

CB85	20 20 20	.BYT	32,32,32,32,32,32,32,32	CD05	00 01 1F	.BYT	0,1,31,248,0,1,15,240
CB8D	20 20 20	.BYT	32,32,32,32,32,32,28,32	CD0D	00 01 0F	.BYT	0,1,15,240,0,1,7,224
CB95	20 20 20	.BYT	32,32,32,32,32,32,2,93	CD15	00 01 0F	.BYT	0,1,7,224,0,1,3,192
CB9D	4A 40 40	.BYT	74,64,64,64,64,64,64,4,64	CD1D	00 01 03	.BYT	0,1,3,192,0,1,1,128
CBA5	40 40 40	.BYT	64,64,64,64,64,64,64,4,64	CD25	00 20 0C	.BYT	0,32,12,0,2,13,0,2
CBAD	40 40 40	.BYT	64,64,64,64,64,64,64,4,64	CD2D	0F EC 00	.BYT	15,236,0,1,15,248,0,1
CBB5	40 40 40	.BYT	64,64,64,64,64,64,64,4,64	CD35	0F FO 00	.BYT	15,240,0,1,15,224,0,1
CBBD	40 40 40	.BYT	64,64,64,64,64,64,64,4,64	CD3D	0F CO 00	.BYT	15,192,0,1,15,128,0,1
CBC5	00 00 00	.BYT	0,0,0,0,0,0,0,0	CD45	0F 00 02	.BYT	15,0,2,14,0,2,12,0
CBCD	00 00 00	.BYT	0,0,0,0,0,0,0,0	CD4D	1E 03 00	.BYT	30,3,0,2,3,0,2,7
CBD5	80 80 92	.BYT	128,128,146,147,148,149,145,145	CD55	80 00 01	.BYT	128,0,1,7,192,0,1,7
				CD5D	FO 00 01	.BYT	240,0,1,15,254,0,1,15
				CD65	F8 00 01	.BYT	248,0,1,15,224,0,1,31
				CD6D	80 00 01	.BYT	128,0,1,30,0,2,24,0
				CD75	25 CO 00	.BYT	37,192,0,1,1,128,0,1
				CD7D	03 CO 00	.BYT	3,192,0,1,7,224,0,1
				CD85	0F FO 00	.BYT	15,240,0,1,31,240,0,1
				CD8D	3F F8 00	.BYT	63,248,0,1,127,254,0,45
				CD95	1C 00 02	.BYT	28,0,2,120,0,1,3,248
				CD9D	00 01 0F	.BYT	0,1,15,248,0,1,127,248
				CDA5	00 01 0F	.BYT	0,1,15,248,0,1,3,248
				CDAD	00 02 78	.BYT	0,2,120,0,2,28,0,44
				CDB5	7F FE 00	.BYT	127,254,0,1,63,248,0,1
				CDBD	1F FO 00	.BYT	31,240,0,1,15,240,0,1
				CDC5	07 EO 00	.BYT	7,224,0,1,3,192,0,1
				CDCD	01 80 00	.BYT	1,128,0,2,192,0,32,48
				CDD5	00 02 3C	.BYT	0,2,60,0,2,63,0,2
				CDDD	1F CO 00	.BYT	31,192,0,1,31,240,0,1
				CDE5	1F FC 00	.BYT	31,252,0,1,15,224,0,1
				CDED	0F 80 00	.BYT	15,128,0,1,15,0,2,6
				CDF5	00 02 06	.BYT	0,2,6,0,30,12,0,2
				CDFD	0E 00 02	.BYT	14,0,2,15,0,2,15,128
				CE05	00 01 0F	.BYT	0,1,15,192,0,1,15,24
				CE0D	00 01 0F	.BYT	0,1,15,240,0,1,15,248
				CE15	00 01 0F	.BYT	0,1,15,236,0,1,13,0
				CE1D	02 OC 00	.BYT	2,12,0,33,8,16,0,1
				CE25	04 32 00	.BYT	4,50,0,1,6,20,0,1
				CE2D	14 4A 00	.BYT	20,164,0,1,24,204,0,1
				CE35	03 00 02	.BYT	3,0,2,109,28,0,1,8
				CE3D	4A 00 02	.BYT	74,0,2,192,0,1,9,48
				CE45	00 01 01	.BYT	0,1,1,0,3,128,0,29
				CE4D	60 00 02	.BYT	96,0,2,240,0,2,96,0
				CE55	3A FF 00	.BYT	58,255,0,1,15,190,240,14
				CE5D	FF FO 3F	.BYT	255,240,63,249,124,63,249,188
				CE65	3D 6A B8	.BYT	61,106,184,245,123,218,181,186
				CE6D	6A F5 B6	.BYT	106,245,182,166,246,182,166,251
				CE75	DA 56 FA	.BYT	218,86,250,106,90,58,170,88
				CE7D	3A A5 A8	.BYT	58,165,168,22,150,88,11,86
				CE85	60 0B 59	.BYT	96,11,89,160,0,1,170,0
; compacted sprite data							
CBDD	01 80 00 SPRITE	.BYT	1,128,0,1,3,192,0,1				
CBE5	03 CO 00	.BYT	3,192,0,1,7,224,0,1				
CBED	07 EO 00	.BYT	7,224,0,1,15,240,0,1				
CBF5	0F FO 00	.BYT	15,240,0,1,31,248,0,1				
CBFD	1F F8 00	.BYT	31,248,0,1,48,12,0,36				
CC05	30 00 02	.BYT	48,0,2,112,0,2,240,0				
CC0D	01 01 FO	.BYT	1,1,240,0,1,3,240,0				
CC15	01 07 FO	.BYT	1,7,240,0,1,15,240,0				
CC1D	01 1F FO	.BYT	1,31,240,0,1,55,240,0				
CC25	02 80 00	.BYT	2,176,0,2,48,0,36,12				
CC2D	00 02 3C	.BYT	0,2,60,0,2,252,0,1				
CC35	03 F8 00	.BYT	3,248,0,1,15,248,0,1				
CC3D	3F F8 00	.BYT	63,248,0,1,7,240,0,1				
CC45	01 FO 00	.BYT	1,240,0,2,240,0,2,96				
CC4D	00 02 60	.BYT	0,2,96,0,41,127,254,0				
CC55	01 1F FC	.BYT	1,31,252,0,1,15,248,0				
CC5D	01 0F FO	.BYT	1,15,240,0,1,7,224,0				
CC65	01 03 CO	.BYT	1,3,192,0,1,1,128,0				
CC6D	01 03 00	.BYT	1,3,0,36,56,0,2,30				
CC75	00 02 1F	.BYT	0,2,31,192,0,1,31,240				
CC7D	00 01 1F	.BYT	0,1,31,254,0,1,31,240				
CC85	00 01 1F	.BYT	0,1,31,192,0,1,30,0				
CC8D	02 38 00	.BYT	2,56,0,36,3,0,2,1				
CC95	80 00 01	.BYT	128,0,1,3,192,0,1,7				
CC9D	EO 00 01	.BYT	224,0,1,15,240,0,1,15				
CCA5	F8 00 01	.BYT	248,0,1,31,252,0,1,127				
CCAD	FE 00 27	.BYT	254,0,39,192,0,2,192,0				
CCB5	01 01 EO	.BYT	1,1,224,0,1,3,224,0				
CCBD	01 0F EO	.BYT	1,15,224,0,1,127,240,0				
CCC5	01 1F FO	.BYT	1,31,240,0,1,7,240,0				
CCCD	01 01 F8	.BYT	1,1,248,0,2,120,0,2				
CCD5	18 00 24	.BYT	24,0,36,48,0,2,176,0				
CDD5	01 37 FO	.BYT	1,55,240,0,1,31,240,0				
CCES	01 0F FO	.BYT	1,15,240,0,1,7,240,0				
CCED	01 03 FO	.BYT	1,3,240,0,1,1,240,0				
CCF5	02 FO 00	.BYT	2,240,0,2,112,0,2,48				
CCFD	00 23 30	.BYT	0,35,48,12,0,1,31,248				

The Source Code

```

CE8D 0C FF 00 .BYT 12,255,0,1,15,239,1
76,11
CE95 FF B0 3D .BYT 255,176,61,127,252,
61,181,172
CE9D 2F E5 B8 .BYT 47,229,184,255,166,
250,255,171
CEA5 DA E5 AA .BYT 218,229,170,105,24
6,234,170,250
CEAD EF 96 FB .BYT 239,150,251,117,15
0,58,181,152
CEB5 3A 65 98 .BYT 58,101,152,37,154,8
8,9,85
CEBD 60 0A 55 .BYT 96,10,85,160,0,1,17
0,0
CEC5 0C FF 00 .BYT 12,255,0,1,3,255,24
0,14
CECD D7 F0 0E .BYT 215,240,14,219,252,
59,255,172
CED5 3B EA 68 .BYT 59,234,104,239,174,
186,239,181
CEDD AA FE D5 .BYT 170,254,213,166,25
4,213,230,245
CEE5 96 D6 F6 .BYT 150,214,246,235,23
0,59,239,88
CEED 3A AA 58 .BYT 58,170,88,42,153,15
2,9,153
CEF5 60 0A 55 .BYT 96,10,85,160,0,1,17
0,0
CFED 0C FF 00 .BYT 12,255,0,1,15,251,2
40,14
CF05 FF F0 3B .BYT 255,240,59,255,252,
59,90,172
CF0D 3F 6E 98 .BYT 63,110,152,111,109,
174,191,182
CF15 AA FE 6A .BYT 170,254,106,102,24
6,186,182,219
CF1D AE 96 FE .BYT 174,150,254,181,86,
58,181,152
CF25 39 A5 98 .BYT 57,165,152,42,229,1
52,9,90
CF2D 60 0A 55 .BYT 96,10,85,160,0,1,17
0,0
CF35 0B .BYT 11
;
;redefined characters
CF36 00 00 00 CHARAC .BYT 0,0,0,16,0,0,0,0
CF3E 00 00 00 .BYT 0,0,0,24,0,0,0,0
CF46 20 04 00 .BYT 32,4,0,58,92,0,32,4
CF4E 00 10 49 .BYT 0,16,73,24,24,146,8,
0
CF56 33 CC 33 .BYT 51,204,51,204,51,20
4,51,204
CF5E 66 99 66 .BYT 102,153,102,153,10
2,153,102,153
;
;color for top line
CF66 0F 0E 0E TPCOLR .BYT 15,14,14,14,14,15,1
4,14
CF6E 0E 0E 0E .BYT 14,14,14,14,14,14,1
4,14
CF76 0E 0F 04 .BYT 14,15,4,4,4,4,15,5
CF7E 05 05 05 .BYT 5,5,5,5,5,5,5,5
CF86 05 05 0F .BYT 5,5,15,5,5,5,5,15

```

Saloon Shootout

Source code by David Hensley, Jr. and Kevin Mykytyn

```

2800          * = 10240
; and 3
2800          TEMPO = 2
; and 5
2800          FREEO = 4
; and 253
2800          NEWO  = 252
2800          CARDS2 = 828
2800          GOODCARD = 838
2800          POSITION = 839
2800          SURPRISE = 842
2800          HCOUNT = 843
2800          OCOUNT = 844
2800          WINDOWL = 845
2800          WINDOWH = 846
2800          HORIZ  = 847
2800          INDEX  = 848
; and 851
2800          SCORE = 850
; and 853
2800          TIME  = 852
2800          COMPUTER = 854
2800          JIFF  = 855
2800          BULLETS = 856
2800          FLAG  = 857
; and 859
2800          SCREEN = 858
; and 861
2800          HIGH  = 860
2800          GLASSCOUN = 862
2800          MOUSEPOS = 863
2800          MOUSEDIR = 864
2800          MOUSEL = 865
2800          MOUSEH = 866
2800          BONFLAG = 867
2800          SONGNUM = 868
2800          NOTENUM = 869
2800          MB     = 10
2800          COUNTER = 870
2800          4C 1A 28 JMP MAIN
2803 00 TEMP .BYT 0
2804 00 TEMP2 .BYT 0
2805 00 TEMP3 .BYT 0
2806 00 TEMP4 .BYT 0
2807 00 SEED .BYT 0
2808 00 SYNC1 .BYT 0
2809 00 FALLL .BYT 0
280A 00 FALLH .BYT 0
280B 00 GLASSL .BYT 0
280C 00 GLASSH .BYT 0
280D 00 PIANOL .BYT 0
280E 00 PIANOH .BYT 0
280F 00 JOYL .BYT 0
2810 00 JOYH .BYT 0
2811 00 LENGTH .BYT 0
2812 00 00 COLOR .BYT 0,0
2814 00 00 COLDIF .BYT 0,0
2816 00 NUMHIT .BYT 0
2817 00 FIRED .BYT 0
2818 00 CARDSL .BYT 0
2819 00 CARDSH .BYT 0
281A          MAIN = *

281A A9 00 LDA #0
281C 8D 5C 03 STA HIGH
281F 8D 5D 03 STA HIGH+1
2822          START =
2822 A9 93 LDA #"(CLR)"
2824 20 D2 FF JSR $FFD2
2827 20 B2 33 JSR HUH
282A 20 4C 2F JSR CHARINIT
282D 20 5D 34 JSR DRAWSCREEN
2830 20 AD 2E JSR INIT
2833 20 70 38 JSR FIXCARDS
2836 20 18 2E JSR DGUN
2839 20 FC 29 JSR HOUT
283C CE 0B 28 LOOP DEC GLASSL
283F D0 0C BNE NOGLASS
2841 AD 0C 28 LDA GLASSH
2844 8D 0B 28 STA GLASSL
2847 20 08 30 JSR GLASS
284A 20 5F 2A JSR LOOK
284D CE 09 28 NOGLASS DEC FALLL
2850 D0 0C BNE NOFALL
2852 AD 0A 28 LDA FALLH
2855 8D 09 28 STA FALLL
2858 20 8D 30 JSR FALL
285B 20 2D 2A JSR BULLOUT
285E CE 61 03 NOFALL DEC MOUSEL
2861 D0 09 BNE NOMOUSE
2863 AD 62 03 LDA MOUSEH
2866 8D 61 03 STA MOUSEL
2869 20 F7 2A JSR MOUSE
286C CE 0D 28 NOMOUSE DEC PIANOL
286F D0 09 BNE NOPIANO
2871 AD 0E 28 LDA PIANOH
2874 8D 0D 28 STA PIANOL
2877 20 F7 30 JSR PIANO
287A AD 0D 28 NOPIANO LDA PIANOL
287D C9 0A CMP #10
287F B0 0F BCS NP
2881 AD 66 03 LDA COUNTER
2884 C9 01 CMP #1
2886 D0 08 BNE NP
2888 AD 11 FF LDA 65297
288B 29 4F AND #79
288D 8D 11 FF STA 65297
2890 CE 0F 28 NP DEC JOYL
2893 D0 09 BNE NOJOY
2895 AD 10 28 LDA JOYH
2898 8D 0F 28 STA JOYL
289B 20 40 2D JSR JOYSTICK
289E CE 18 28 NOJOY DEC CARDSL
28A1 D0 09 BNE NOCARDS
28A3 AD 19 28 LDA CARDSH
28A6 8D 18 28 STA CARDSL
28A9 20 DD 37 JSR CARDS
28AC 20 EE 28 NOCARDS JSR DELAY
28AF CE 57 03 DEC JIFF
28B2 D0 88 BNE LOOP
28B4 CE 54 03 DEC TIME
28B7 20 6E 29 JSR SCROUT
28BA AD 54 03 LDA TIME
28BD F0 08 BEQ FINISH
28BF AD 58 03 LDA BULLETS

```


Chapter 3

```

29DA 20 08 34      JSR  STROUT
29DD AC 04 28      LDY  TEMP2
29E0 C8            INY
29E1 98           TYA
29E2 20 D2 FF     JSR  $FFD2
29E5 C8           INY
29E6 C0 60       CPY  #96
29E8 D0 04       BNE  ROT
29EA 38          SEC
29EB E9 3E       SBC  #62
29ED A8          TAY
29EE 98          TYA
29EF 20 D2 FF     JSR  $FFD2
29F2 A9 92       LDA  #"(OFF)"
29F4 20 D2 FF     JSR  $FFD2
29F7 68          PLA
29F8 A8          TAY
29F9 68          PLA
29FA AA          TAX
29FB 60          RTS
29FC AD 5C 03     LDY  HIGH
29FF 38          SEC
2A00 ED 52 03     SBC  SCORE
2A03 8D 03 28     STA  TEMP
2A06 AD 5D 03     LDA  HIGH+1
2A09 ED 53 03     SBC  SCORE+1
2A0C 0D 03 28     ORA  TEMP
2A0F 80 0B       BCS  LEAVE
2A11 A0 01       LDY  #1
2A13 B9 52 03     TR   SCORE,Y
2A16 99 5C 03     LDA  HIGH,Y
;
2A19 88          DEY
2A1A 10 F7       BPL  TR
2A1C A2 0E       LEAVE
2A1E A0 00       LDY  #0
2A20 18          CLC
2A21 20 F0 FF     JSR  $FFF0
2A24 AE 5C 03     LDX  HIGH
2A27 AD 5D 03     LDA  HIGH+1
2A2A 4C 13 34     JMP  NUMOUT
;
; update bullets
;
2A2D            = *
2A2D AD 5A 03     LDA  SCREEN
2A30 18          CLC
2A31 69 C4       ADC  #<964
2A33 85 04       STA  FREE0
2A35 AD 5B 03     LDA  SCREEN+1
2A38 69 03       ADC  #>964
2A3A 85 05       STA  FREE0+1
2A3C AE 58 03     LDX  BULLETS
2A3F E8          INX
2A40 A0 00       LDY  #0
2A42 A9 BC       LDA  #188
2A44 CA          DEX
2A45 F0 02       BEQ  PN
2A47 10 02       BPL  PUTIT
2A49 A9 20       PN   PUTIT
2A4B 91 04       LDA  #32
2A4D 20 30 2D     STA  (FREE0),Y
2A50 AD 56 03     JSR  ADDCOLOR
2A53 D0 02       LDA  COMPUTER
2A55 A9 71       BNE  PLS3
2A57 91 02     PLS3 LDA  #113
2A59 C8          LDA  (TEMPO),Y
2A5A C0 23       INY
2A5C D0 E4       CPY  #35
2A5E 60          BNE  BULLOOP
                RTS
;
; who's that looking in the window

```

The Source Code

2AF7	AD 5E 03	LDA	GLASSCOUNT	2B8A	85 04	STA	FREE0
2AFA	C9 0B	CMP	#MB+1	2B8C	A5 05	LDA	FREE0+1
2AFC	F0 17	BEQ	MOVING	2B8E	E9 00	SBC	#0
2AFE	C9 0A	CMP	#MB	2B90	85 05	STA	FREE0+1
2B00	D0 12	BNE	NOMOU	2B92	A2 00	LDX	#0
2B02	A9 00	LDA	#0	2B94	20 B0 2C	JSR	PUT2
2B04	8D 5F 03	STA	MOUSEPOS	2B97	A9 0A	LDA	#10
2B07	A9 01	LDA	#1	2B99	20 F4 28	JSR	ADDScore
2B09	8D 60 03	STA	MOUSEDIR	2B9C	AD 5E 03	LDA	GLASSCOUNT
2B0C	EE 5E 03	INC	GLASSCOUNT	2B9F	C9 0A	CMP	#MB
2B0F	A9 65	LDA	#101	2BA1	80 03	BCS	NORAISE
2B11	8D 62 03	STA	MOUSEH	2BA3	EE 5E 03	INC	GLASSCOUNT
2B14	60	RTS		2BA6	60	RTS	
2B15	AD 5A 03	NOMOU	MOVING	2BA7	C9 84	NORAISE	
2B18	18	CLC		2BA9	F0 08	NOTMUG	
2B19	69 D8	ADC	#<728	2BAB	C9 89	CMP	#137
2B1B	85 04	STA	FREE0	2BAD	90 05	BCC	GO
2B1D	AD 5B 03	LDA	SCREEN+1	2BAF	C9 8D	CMP	#141
2B20	69 02	ADC	#>728	2BB1	80 01	BCS	GO
2B22	85 05	STA	FREE0+1	2BB3	60	RTS	
2B24	AC 5F 03	LDY	MOUSEPOS	2BB4	C9 A0	CMP	#160
2B27	A9 20	LDA	#32	2BB6	F0 4C	BEQ	FLO
2B29	91 04	STA	(FREE0),Y	2BB8	C9 A2	CMP	#162
2B2B	C8	INY		2BBA	80 48	BCS	FLO
2B2C	91 04	STA	(FREE0),Y	2BBC	C9 8D	CMP	#141
2B2E	AD 5F 03	LDA	MOUSEPOS	2BBE	90 44	BCC	FLO
2B31	18	CLC		2BC0	AD 47 03	LDA	POSITION
2B32	6D 60 03	ADC	MOUSEDIR	2BC3	38	SEC	
2B35	8D 5F 03	STA	MOUSEPOS	2BC4	E9 13	SBC	#19
2B38	F0 29	BEQ	BYEMOUSE	2BC6	AA	TAX	
2B3A	C9 0E	CMP	#14	2BC7	BD D2 2C	LDA	NEWT,X
2B3C	F0 25	BEQ	BYEMOUSE	2BCA	18	CLC	
2B3E	A2 00	LDX	#0	2BCB	6D 50 03	ADC	INDEX
2B40	A8	TAY		2BCE	C9 05	CMP	#5
2B41	AD 60 03	LDA	MOUSEDIR	2BD0	90 03	BCC	ORAY
2B44	10 02	BPL	RIGHT	2BD2	38	SEC	
2B46	A2 02	LDX	#2	2BD3	E9 05	SBC	#5
2B48	BD 69 2B	LDA	MOUSHP,X	2BD5	AA	TAX	
2B4B	91 04	STA	(FREE0),Y	2BD6	BD 3C 03	LDA	CARDS2,X
2B4D	C8	INY		2BD9	29 7F	AND	#127
2B4E	BD 6A 2B	LDA	MOUSHP+1,X	2BD8	4A	LSR	
2B51	91 04	STA	(FREE0),Y	2BDC	4A	LSR	
2B53	20 30 2D	JSR	ADDCOLOR	2BD0	CD 46 03	CMP	GOODCARD
2B56	AD 56 03	LDA	COMPUTER	2BE0	D0 0A	BNE	RESET
2B59	D0 02	BNE	PS5	2BE2	CE 46 03	DEC	GOODCARD
2B5B	A9 71	LDA	#113	2BE5	D0 10	BNE	NOR
2B5D	91 02	STA	(TEMPO),Y	2BE7	A9 23	LDA	#35
2B5F	88	DEY		2BE9	20 4E 29	JSR	ADDBULL
2B60	91 02	STA	(TEMPO),Y	2BEC	A9 05	RESET	
2B62	60	RTS		2BEE	8D 46 03	STA	GOODCARD
2B63	A9 00	LDA	#0	2BF1	20 70 38	JSR	FIXCARDS
2B65	8D 5E 03	STA	GLASSCOUNT	2BF4	4C 75 2C	JMP	DUN
2B68	60	RTS		2BF7	A9 0A	LDA	#10
2B69	1B 1C 1D	MOUSHP	.BYT 27,28,29,30	2BF9	20 F4 28	JSR	ADDScore
				2BFC	A9 1C	LDA	#28
				2BFE	9D 3C 03	STA	CARDS2,X
				2C01	4C 75 2C	JMP	DUN
				2C04	C9 66	FLO	
				2C06	D0 08	BNE	NOTBACK
				2C08	A9 0A	LDA	#10
				2C0A	20 2C 29	JSR	SUBSCORE
				2C0D	4C 75 2C	JMP	DUN
				2C10	C9 A2	CMP	#162
				2C12	90 1A	BCC	NTG
				2C14	C9 B4	CMP	#180
				2C16	80 16	BCS	NTG
				2C18	C9 AB	CMP	#171
				2C1A	80 08	BCS	BADG
				2C1C	A9 64	LDA	#100
				2C1E	20 2C 29	JSR	SUBSCORE
				2C21	4C 75 2C	JMP	DUN
				2C24	A9 00	BADG	#0
2B6D		HIT					
2B6D	AC 47 03	LDY	POSITION				
2B70	B1 04	LDA	(FREE0),Y				
2B72	C9 85	CMP	#133				
2B74	90 31	BCC	NOTMUG				
2B76	C9 89	CMP	#137				
2B78	B0 2D	BCS	NOTMUG				
2B7A	38	SEC					
2B7B	E9 85	SBC	#133				
2B7D	AA	TAX					
2B7E	BD FB 2C	LDA	SUB,X				
2B81	8D 03 28	STA	TEMP				
2B84	A5 04	LDA	FREE0				
2B86	38	SEC					
2B87	ED 03 28	SBC	TEMP				

Chapter 3

2C26	8D 59 03	STA	FLAG	2CC8	91 04	STA	(FREE0),Y
2C29	A9 19	LDA	#25	2CCA	E8	INX	
2C2B	20 F4 28	JSR	ADDScore	2CCB	C8	INY	
2C2E	C9 18	CMP	#27	2CCC	8D DB 2C	LDA	TABLE,X
2C30	90 26	BCC	NMH	2CCF	91 04	STA	(FREE0),Y
2C32	C9 1F	CMP	#31	2CD1	60	RTS	
2C34	B0 22	BCS	NMH	2CD2	00 00 00	NEWT	.BYT 0,0,0,1,1,2,2,2
2C36	AD 60 03	LDA	MOUSEDIR	2CDB	89 8A 8B	TABLE	.BYT 137,138,139,140
2C39	49 FF	EOR	#255	2CDF	8D 8E 8F	.BYT	141,142,143,144,14
2C3B	18	CLC					5,146,147,148,149,
2C3C	69 01	ADC	#1				150
2C3E	8D 60 03	STA	MOUSEDIR	2CE9	97 98 99	.BYT	151,152,153,154,15
2C41	AD 62 03	LDA	MOUSEH				5,156,157,158,159,
2C44	38	SEC					161
2C45	E9 14	SBC	#20	2CF3	66 66 66	.BYT	102,102,102,102,32,
2C47	30 03	BMI	NOG				32,32,32
2C49	8D 62 03	STA	MOUSEH	2CFB	00 01 28	SUB	.BYT 0,1,40,41
2C4C	A9 65	LDA	#101	2CFF	AE 16 28	FIX	LDX NUMHIT
2C4E	38	SEC		2D02	F0 23	BEQ	FG
2C4F	ED 62 03	SBC	MOUSEH	2D04	CA	DEX	
2C52	20 F4 28	JSR	ADDScore	2D05	BD C3 3B	CLP	LDA STACK1,X
2C55	4C 75 2C	JMP	DUN	2D08	85 04	STA	FREE0
2C58	C9 B6	CMP	#182	2D0A	BD D7 3B	LDA	STACK2,X
2C5A	D0 19	BNE	DUN	2D0D	85 05	STA	FREE0+1
2C5C	A9 00	LDA	#0	2D0F	BC EB 3B	LDY	STACK3,X
2C5E	8D 65 03	STA	NOTENUM	2D12	BD FF 3B	LDA	STACK4,X
2D61	A9 01	LDA	#1	2D15	91 04	STA	(FREE0),Y
2D63	8D 66 03	STA	COUNTER	2D17	20 30 2D	JSR	ADDCOLOR
2D66	EE 64 03	INC	SONGNUM	2D1A	BD 13 3C	LDA	STACK5,X
2D69	AD 64 03	LDA	SONGNUM	2D1D	91 02	STA	(TEMPO),Y
2D6C	C9 03	CMP	#3	2D1F	CA	DEX	
2D6E	90 05	BCC	DUN	2D20	10 E3	BPL	CLP
2D70	A9 00	LDA	#0	2D22	A9 00	LDA	#0
2D72	8D 64 03	STA	SONGNUM	2D24	8D 16 28	STA	NUMHIT
2D75	AE 16 28	LDX	NUMHIT	2D27	AD 11 FF	FG	LDA 65297
2D78	EE 16 28	INC	NUMHIT	2D2A	29 1F	AND	#31
2D7B	A5 04	LDA	FREE0	2D2C	8D 11 FF	STA	65297
2D7D	9D C3 3B	STA	STACK1,X	2D2F	60	RTS	
2D80	A5 05	LDA	FREE0+1	2D30	A5 04	ADDCOLOR	LDA FREE0
2D82	9D D7 3B	STA	STACK2,X	2D32	18	CLC	
2D85	AD 47 03	LDA	POSITION	2D33	6D 14 28	ADC	COLDIF
2D88	9D EB 3B	STA	STACK3,X	2D36	85 02	STA	TEMPO
2D8B	B1 04	LDA	(FREE0),Y	2D38	A5 05	LDA	FREE0+1
2D8D	C9 1B	CMP	#27	2D3A	6D 15 28	ADC	COLDIF+1
2D8F	90 06	BCC	DT	2D3D	85 03	STA	TEMPO+1
2C91	C9 1F	CMP	#31	2D3F	60	RTS	
2C93	B0 02	BCS	DT				
2C95	A9 20	LDA	#32				
2C97	9D FF 3B	DT	STACK4,X				
2C9A	20 30 2D	JSR	ADDCOLOR				
2C9D	B1 02	LDA	(TEMPO),Y	2D40	20 4C 2E	JOYSTICK	= *
2C9F	9D 13 3C	STA	STACK5,X	2D43	4A	JSR	LSR
2CA2	AD 56 03	LDA	COMPUTER	2D44	80 0D	BCS	NOTUP
2CA5	D0 02	BNE	PS4	2D46	AE 11 28	LDX	LENGTH
2CA7	A9 71	LDA	#113	2D49	E0 10	CPX	#16
2CA9	91 02	STA	(TEMPO),Y	2D4B	F0 06	BEQ	NOTUP
2CAB	A9 84	LDA	#132	2D4D	EE 11 28	INC	LENGTH
2CAD	91 04	STA	(FREE0),Y	2D50	20 18 2E	JSR	DGUN
2CAF	60	RTS		2D53	4A	LSR	
2CB0	AC 47 03	LDY	POSITION	2D54	80 0D	NOTUP	
2CB3	BD DB 2C	LDA	TABLE,X	2D56	AE 11 28	BCS	NOTDOWN
2CB6	91 04	STA	(FREE0),Y	2D59	E0 02	LDX	LENGTH
2CB8	C8	INX		2D5B	F0 06	CPX	#2
2CB9	E8	INX		2D5D	CE 11 28	BEQ	NOTDOWN
2CBA	BD DB 2C	LDA	TABLE,X	2D60	20 18 2E	DEC	LENGTH
2CBD	91 04	STA	(FREE0),Y	2D63	4A	JSR	DGUN
2CBF	E8	INX		2D64	80 14	LSR	
2CC0	98	TYA		2D66	48	BCS	NOTLEFT
2CC1	18	CLC		2D67	AD 47 03	PHA	
2CC2	69 27	ADC	#39	2D6A	F0 0D	LDA	POSITION
2CC4	A8	TAY		2D6C	A2 00	BEQ	NOG1
2CC5	BD DB 2C	LDA	TABLE,X	2D6E	20 DB 2D	LDX	#0
						JSR	PUT

The Source Code

2D71 CE 47 03 2D74 A2 04 2D76 20 DB 2D 2D79 68 2D7A 4A 2D7B 80 16 2D7D 48 2D7E AD 47 03 2D81 C9 1F 2D83 F0 0D 2D85 A2 00 2D87 20 DB 2D 2D8A EE 47 03 2D8D A2 04 2D8F 20 DB 2D 2D92 68 2D93 4A 2D94 B0 3F 2D96 AD 17 28 2D99 D0 34 2D9B AD 58 03 2D9E F0 2F 2DA0 A9 01 2DA2 20 5C 29 2DA5 20 33 2E 2DA8 AD 5A 03 2DAB 18 2DAC 69 57 2DAE 85 04 2DB0 AD 5B 03 2DB3 69 00 2DB5 85 05 2DB7 AE 11 28 2DBA F0 10 2DBC A5 04 2DBE 18 2DBF 69 28 2DC1 85 04 2DC3 A5 05 2DC5 69 00 2DC7 85 05 2DC9 CA 2DCA D0 F0 2DCC 20 6D 2B 2DCF A9 01 2DD1 8D 17 28 2DD4 60 2DD5 A9 00 2DD7 8D 17 28 2DDA 60 2DDB AD 5A 03 2DDE 18 2DDF 69 FF 2DE1 85 04 2DE3 AD 5B 03 2DE6 69 02 2DE8 85 05 2DEA AC 47 03 2DED BD 10 2E 2DF0 91 04 2DF2 E8 2DF3 98 2DF4 18 2DF5 69 28 2DF7 A8 2DF8 BD 10 2E 2DFB 91 04 2DFD E8 2DFE 98 2DFF 18 2E00 69 28 2E02 A8	DEC POSITION LDX #4 JSR PUT PLA LSR BCS NOTRIGHT PHA LDA POSITION CMP #31 BEQ NOG2 LDX #0 JSR PUT INC POSITION LDX #4 JSR PUT PLA LSR BCS NOTFIRE LDA FIRED BNE FORGT LDA BULLETS BEQ FORGT LDA #1 JSR SUBBULL JSR FIRESOUND LDA SCREEN CLC ADC #87 STA FREE0 LDA SCREEN+1 ADC #0 STA FREE0+1 LDX LENGTH BEQ NADD LDA FREE0 CLC ADC #40 STA FREE0 LDA FREE0+1 ADC #0 STA FREE0+1 DEX BNE ADDLOOP JSR HIT LDA #1 STA FIRED RTS LDA #0 STA FIRED RTS LDA SCREEN CLC ADC #<767 STA FREE0 LDA SCREEN+1 ADC #>767 STA FREE0+1 LDY POSITION LDA GUN,X STA (FREE0),Y INX TYA CLC ADC #40 TAY LDA GUN,X STA (FREE0),Y INX TYA CLC ADC #40 TAY	2E03 BD 10 2E 2E06 #4 2E08 E8 2E09 C8 2E0A BD 10 2E 2E0D 91 04 2E0F 60 2E10 A0 A0 A0 GUN 2E18 48 2E19 A9 FF 2E1B A0 FF 2E1D 99 FF 23 2E20 C8 2E21 C0 10 2E23 F0 09 2E25 CC 11 28 2E28 D0 F3 2E2A A9 96 2E2C D0 EF 2E2E 20 6E 2E HOME 2E31 68 2E32 60 2E33 2E33 AD 56 03 2E36 D0 09 2E38 AD 11 FF PLUS4 2E3B 09 40 2E3D 8D 11 FF 2E40 60 2E41 A9 80 2E43 8D 04 D4 2E46 A9 81 2E48 8D 04 D4 2E4B 60 ; get joystick value ; 2E4C 2E4C AD 56 03 2E4F F0 04 2E51 AD 00 DC 2E54 60 2E55 A9 FD 2E57 8D 08 FF 2E5A EA 2E5B EA 2E5C AD 08 FF 2E5F 8D 03 28 2E62 29 80 2E64 08 2E65 AD 03 28 2E68 28 2E69 D0 02 2E6B 29 EF 2E6D 60 2E6E AD 5A 03 2E71 18 2E72 69 77 2E74 85 02 2E76 AD 5B 03 2E79 69 00 2E7B 85 03 2E7D A0 00 2E7F A2 00 2E81 A9 20 2E83 EC 11 28 2E86 D0 02 2E88 A9 1F 2E8A 91 02 2E8C A5 02 2E8E 18	LDA GUN,X STA (FREE0),Y INX INY LDA GUN,X STA (FREE0),Y RTS .BYT 160,160,160,12 8,129,130,131 PHA LDA #255 LDY #255 STA 9215,Y INY CPY #16 BEQ HOME CPY LENGTH BNE DGLLOOP1 LDA #150 BNE DGLLOOP1 JSR GUAGE PLA RTS = * LDA COMPUTER BNE CM64 LDA 65297 ORA #64 STA 65297 RTS CM64 LDA #128 STA 54276 LDA #129 STA 54276 RTS = * LDA COMPUTER BEQ LONG2 LDA \$DC00 RTS LDA #5FD STA \$FF08 NOP NOP LDA \$FF08 STA TEMP AND #128 PHP LDA TEMP PLP BNE NOJOY2 AND #239 RTS LDA SCREEN CLC ADC #119 STA TEMPO LDA SCREEN+1 ADC #0 STA TEMPO+1 LDY #0 LDX #0 LDA #32 CPX LENGTH BNE NOTG LDA #31 STA (TEMPO),Y LDA TEMPO CLC
--	--	--	---

Chapter 3

2E8F	6D 14 28	ADC	COLDIF	2F35	8D 61 03	STA	MOUSEL	
2E92	85 FC	STA	NEW0	2F38	8D 62 03	STA	MOUSEH	
2E94	A5 03	LDA	TEMPO+1	2F3B	A9 C8	LDA	#200	
2E96	6D 15 28	ADC	COLDIF+1	2F3D	8D 54 03	STA	TIME	
2E99	85 FD	STA	NEW0+1	2F40	A9 23	LDA	#35	
2E9B	AD 56 03	LDA	COMPUTER	2F42	8D 58 03	STA	BULLETS	
2E9E	D0 02	BNE	PLS4	2F45	20 EC 2A	JSR	WINDOW	
2EA0	A9 71	LDA	#113	2F48	20 6E 29	JSR	SCROUT	
2EA2	91 FC	STA	(NEW0).Y	2F4B	60	RTS		
2EA4	20 E9 30	JSR	ADD40					
2EA7	E8	INX						
2EA8	E0 11	CPX	#17					
2EAA	D0 D5	BNE	GL					
2EAC	60	RTS						
	initialization				initialize characters			
2EAD		=	*	2F4C		CHARINIT	=	
2EAD	AD 56 03	LDA	COMPUTER	2F4C	AD 56 03		LDA	COMPUTER
2EB0	F0 1E	BEQ	NOCLSID	2F4F	D0 03		BNE	DOCUST
2EB2	A9 00	LDA	#0	2F51	4C D7 2F		JMP	CUST16
2EB4	A0 17	LDY	#23	2F54	78		SEI	
2EB6	99 00 D4	STA	54272.Y	2F55	A9 33		LDA	#51
2EB9	88	DEY		2F57	85 01		STA	1
2EBA	10 FA	BPL	CLSID	2F59	20 76 2F		JSR	TRANSFER
2EBC	A9 0F	LDA	#15	2F5C	A9 37		LDA	#55
2EBE	8D 18 D4	STA	54296	2F5E	85 01		STA	1
2EC1	A9 19	LDA	#25	2F60	58		CLI	
2EC3	8D 13 D4	STA	54291	2F61	A9 18		LDA	#24
2EC6	A9 15	LDA	#21	2F63	8D 18 D0		STA	53272
2EC8	8D 05 D4	STA	54277	2F66	A9 D8		LDA	#216
2ECB	A9 28	LDA	#40	2F68	8D 16 D0		STA	53270
2ECD	8D 01 D4	STA	54273	2F6B	A9 0F		LDA	#15
2ED0		=	*	2F6D	8D 22 D0		STA	53282
2ED0	A9 00	LDA	#0	2F70	A9 02		LDA	#2
2ED2	8D 08 28	STA	SYNC1	2F72	8D 23 D0		STA	53283
2ED5	8D 16 28	STA	NUMHIT	2F75	60		RTS	
2ED8	8D 5E 03	STA	GLASSCOUNT	2F76	A2 07	TRANSFER	LDX	#7
2EDB	A9 7D	LDA	#125	2F78	A0 00		LDY	#0
2EDD	8D 0B 28	STA	GLASSL	2F7A	A9 00		LDA	#<53248
2EE0	8D 0C 28	STA	GLASSH	2F7C	85 02		STA	TEMPO
2EE3	A9 05	LDA	#5	2F7E	85 04		STA	FREE0
2EE5	8D 0A 28	STA	FALLH	2F80	A9 D0		LDA	#>53248
2EE8	8D 09 28	STA	FALLL	2F82	85 03		STA	TEMPO+1
2EEB	8D 46 03	STA	GOODCARD	2F84	A9 20		LDA	#>8192
2EEE	8D 66 03	STA	COUNTER	2F86	85 05		STA	FREE0+1
2EF1	A9 41	LDA	#65	2F88	B1 02	CHLOOP1	LDA	(TEMPO).Y
2EF3	8D 0D 28	STA	PIANOL	2F8A	8E 03 28		STX	TEMP
2EF6	8D 0E 28	STA	PIANOH	2F8D	E0 04		CPX	#4
2EF9	A9 0F	LDA	#15	2F8F	B0 07		BCS	STR
2EFB	8D 0F 28	STA	JOYL	2F91	AE 56 03		LDX	COMPUTER
2EFE	8D 10 28	STA	JOYH	2F94	D0 02		BNE	STR
2F01	A9 FA	LDA	#250	2F96	49 FF		EOR	#255
2F03	8D 18 28	STA	CARDSL	2F98	AE 03 28	STR	LDX	TEMP
2F06	8D 19 28	STA	CARDSH	2F9B	91 04		STA	(FREE0).Y
2F09	A9 0A	LDA	#10	2F9D	88		DEY	
2F0B	8D 47 03	STA	POSITION	2F9E	D0 E8		BNE	CHLOOP1
2F0E	8D 11 28	STA	LENGTH	2FA0	E6 03		INC	TEMPO+1
2F11	A2 04	LDX	#4	2FA2	E6 05		INC	FREE0+1
2F13	20 DB 2D	JSR	PUT	2FA4	E0 04		CPX	#4
2F16	A9 FF	LDA	#255	2FA6	D0 09		BNE	STR2
2F18	8D 50 03	STA	INDEX	2FA8	AD 56 03		LDA	COMPUTER
2F1B	A9 00	LDA	#0	2FAB	D0 04		BNE	STR2
2F1D	8D 4A 03	STA	SURPRISE	2FAD	A9 D0		LDA	#>53248
2F20	8D 52 03	STA	SCORE	2FAF	85 03		STA	TEMPO+1
2F23	8D 53 03	STA	SCORE+1	2FB1	CA	STR2	DEX	
2F26	8D 63 03	STA	BONFLAG	2FB2	10 D4		BPL	CHLOOP1
2F29	A9 00	LDA	#0	2FB4	A0 00		LDY	#0
2F2B	8D 64 03	STA	SONGNUM	2FB6	B9 98 38	DEFLOOP	LDA	CHARDEF.Y
2F2E	A9 FF	LDA	#255	2FB9	99 00 24		STA	9216.Y
2F30	8D 65 03	STA	NOTENUM	2FBC	B9 93 39		LDA	CHARDEF+248.Y
2F33	A9 65	LDA	#101	2FBF	99 F8 24		STA	9216+248.Y
				2FC2	B9 B3 3A		LDA	NEWDEF.Y
				2FC5	99 D0 21		STA	8656.Y
				2FC8	88		DEY	
				2FC9	D0 EB		BNE	DEFLOOP
				2FCB	A0 1F		LDY	#31

The Source Code

```

2FCD B9 93 3A MD          LDA MOUS,Y          3056 20 FB 2F          JSR RAND
2FD0 99 D8 20          STA 8408,Y         3059 30 2E          BMI FORGET
2FD3 88              DEY              305B 20 FB 2F AG   JSR RAND
2FD4 10 F7            BPL MD           305E 29 03          AND #3
2FD6 60              RTS             3060 C9 03          CMP #3
2FD7 AD 12 FF CUST16   LDA 65298        3062 F0 25          BEQ FORGET
2FDA 29 FB          AND #251        3064 AA            TAX
2FDC 8D 12 FF        STA 65298        3065 A5 04          LDA FREE0
2FDF AD 13 FF        LDA 65299        3067 18            CLC
2FE2 29 03          AND #3          3068 7D 8A 30       ADC NUMBER,X
2FE4 09 20          ORA #32         306B 85 04          STA FREE0
2FE6 8D 13 FF        STA 65299        306D A5 05          LDA FREE0+1
2FE9 A9 D8          LDA #216        306F 69 00          ADC #0
2FEB 8D 07 FF        STA 65287        3071 85 05          STA FREE0+1
2FEE A9 62          LDA #98         3073 A0 00          LDY #0
2FF0 8D 17 FF        STA 65303        3075 A9 85          LDA #133
2FF3 A9 41          LDA #65         3077 91 04          STA (FREE0),Y
2FF5 8D 16 FF        STA 65302        3079 C8            INY
2FF8 4C 76 2F        JMP TRANSFER    307A A9 86          LDA #134
; random number      307C 91 04          (FREE0),Y
;                   307E A0 28          STA #40
;                   3080 A9 87          LDA #135
2FFB AD 07 28 RAND    = *            3082 91 04          STA (FREE0),Y
2FFB AD 07 28        LDA SEED        3084 C8            INY
2FFE 0A              ASL             3085 A9 88          LDA #136
2FFF 0A              ASL             3087 91 04          STA (FREE0),Y
3000 38              SEC            3089 60            RTS
3001 6D 07 28        ADC SEED        308A 00 F0 78       .BYT 0,240,120
3004 8D 07 28        STA SEED        308D AD 08 28       FALL
3007 60              RTS            3090 4A            LSR
;                   3091 B0 55          BCS NO
; scroll glasses     3093 A2 09          LDX #9
;                   3095 AD 5A 03       LDA SCREEN
3008 20 FF 2C        JSR FIX         3098 18            CLC
3008 EE 08 28        INC SYNC1      3099 69 E6          ADC #<742
300E AD 5A 03        LDA SCREEN     309B 85 04          STA FREE0
3011 18              CLC            309D AD 5B 03       LDA SCREEN+1
3012 69 70          ADC #<368      30A0 69 02          ADC #>742
3014 85 02          STA TEMPO      30A2 85 05          STA FREE0+1
3016 85 04          STA FREE0      30A4 A5 04          LDA FREE0
3018 AD 5B 03        LDA SCREEN+1   30A6 38            SEC
301B 69 01          ADC #>368      30A7 E9 28          SBC #40
301D 85 03          STA TEMPO+1   30A9 85 02          STA TEMPO
301F 85 05          STA FREE0+1   30AB A5 05          LDA FREE0+1
3021 A9 03          LDA #3         30AD E9 00          SBC #0
3023 8D 03 28        STA TEMP       30AF 85 03          STA TEMPO+1
3026 A2 02 GLOOP3    LDX #2         30B1 A0 01          LDY #1
3028 A0 0E GLOOP1    LDY #14        30B3 B1 02          LDA (TEMPO),Y
302A B1 02 GLOOP2    LDA (TEMPO),Y  30B5 E0 09          CPX #9
302C C9 85          CMP #133       30B7 D0 0B          BNE NCH
302E 90 04          BCC BAD        30B9 C9 85          CMP #133
3030 C9 89          CMP #137       30BB D0 07          BNE NCH
3032 90 02          BCC FOG        30BD 48            PHA
3034 A9 20 BAD       LDA #32        30BE A9 05          LDA #5
3036 C8 FOG          INY            30C0 20 2C 29       JSR SUBSCORE
3037 91 02          STA (TEMPO),Y  30C3 68            PLA
3039 88              DEY            30C4 C9 84          CMP #132
303A 88              DEY            30C6 D0 02          BNE DOT
303B 10 ED          BPL GLOOP2     30C8 A9 20          LDA #32
303D C8              INY            30CA 91 04          STA (FREE0),Y
303E A9 20          LDA #32        30CC 88            DEY
3040 91 02          STA (TEMPO),Y  30CD 10 E4          BPL FLOOP1
3042 20 E9 30        JSR ADD40      30CF A5 04          LDA FREE0
3045 CA              DEX            30D1 38            SEC
3046 D0 E0          BNE GLOOP1     30D2 E9 28          SBC #40
3048 20 E9 30        JSR ADD40      30D4 85 04          STA FREE0
304B CE 03 28        DEC TEMP       30D6 A5 05          LDA FREE0+1
304E D0 D6          BNE GLOOP3     30D8 E9 00          SBC #0
3050 AD 08 28        LDA SYNC1      30DA 85 05          STA FREE0+1
3053 4A              LSR            30DC CA            DEX
3054 B0 33          BCS FORGET     30DD D0 C5          BNE FLOOP2
;                   30DF A0 01          LDY #1

```


The Source Code

```

3208 02 03 10 .WORD 770,784,798,810,82 33B7 F0 2B BEQ COMM64
      2,834,844,854,864 33B9 A9 00 LDA #<2048
321A 69 03 71 .WORD 873,881,889,897,90 33BB 8D 12 28 STA COLOR
      4,911,917,923,929 33BE A9 08 LDA #>2048
322C A6 03 AB .WORD 934,939,944,948,95 33C0 8D 13 28 STA COLOR+1
      3,957 33C3 A9 00 LDA #<-1024
3238 87 21 86 NOTE64 .WORD 8583,9094,9634,10 33C5 8D 14 28 STA COLDIF
      207,10814,11457,1 33C8 A9 FC LDA #>-1024
      2139 33CA 8D 15 28 STA COLDIF+1
3246 3C 32 39 .WORD 12860,13625,14435 33CD A9 03 LDA #3
      ,15294,16203,1716 33CF 8D 10 FF STA 65296
      7,18188 33D2 A9 C8 LDA #200
3254 45 4B BF .WORD 19269,20415,21629 33D4 8D 0F FF STA 65295
      ,22915,24278,2572 33D7 A9 0F LDA #15
      1,27251 33D9 8D 11 FF STA 65297
3262 C7 70 7C .WORD 28871,30588,32407 33DC A9 00 LDA #0
      ,34334 33DE A2 00 LDX #<-3072
326A 18 8E 8B .WORD 36376,38539,40830 33E0 A0 0C LDY #>-3072
      ,43258,45830,4855 33E2 D0 1A BNE AR4
      6 33E4 A9 00 LDA #<55296
3276 F3 C8 E6 .WORD 51443,54502 33E6 8D 12 28 STA COLOR
327A 80 32 02 SONGS .WORD SONG1,SONG2,SON 33E9 A9 D8 LDA #>55296
      G3 33EB 8D 13 28 STA COLOR+1
3280 19 01 1B SONG1 .BYT 25,1,27,2,25,1,27,2, 33EE A9 00 LDA #<54272
      25,2,22,1,23,2,22,1 33F0 8D 14 28 STA COLDIF
3290 17 02 16 .BYT 23,2,22,2,20,1,22,2, 33F3 A9 D4 LDA #>54272
      20,1,22,2,20,2,15,8 33F5 8D 15 28 STA COLDIF+1
32A0 16 01 14 .BYT 22,1,20,2,19,1,22,2, 33F8 A9 01 LDA #1
      20,2,25,8,18,1,22,2 33FA A2 00 LDX #<-1024
32B0 19 01 1D .BYT 25,1,29,2,27,2,25,8 33FC A0 04 LDY #>1024
32B8 19 01 1B .BYT 25,1,27,2,25,1,27,2, 33FE 8D 56 03 AR4 STA COMPUTER
      25,2,22,1,23,2,22,1 3401 8E 5A 03 STX SCREEN
32C8 17 02 16 .BYT 23,2,22,2,20,1,22,2, 3404 8C 5B 03 STY SCREEN+1
      20,1,22,2,20,2,15,8 3407 60 RTS
32D8 14 01 16 .BYT 20,1,22,2,20,1,25,2,  ; string out routine
      22,2,20,1,22,2,20,1  ;
      25,2,22,2,20,1,22,2,  ;
      20,1,25,2,20,2,18,2 3408 AE 56 03 STROUT LDX COMPUTER
32F8 0D 01 0F .BYT 13,1,15,2,13,1,10,2, 340B F0 03 BEQ PLUS2
      6,4 340D 4C 1E AB JMP $AB1E
3302 09 01 10 SONG2 .BYT 9,1,16,1,9,1,13,1,16, 3410 4C 88 90 PLUS2 JMP $9088
      2,8,1,16,1,8,1,11,1  ;
      16,5  ; number output routine
3314 10 05 .BYT 16,5  ;
3316 09 01 10 .BYT 9,1,16,1,9,1,13,1,16,  ;
      2,8,1,16,1,8,1,11,1 3413 8E 03 28 NUMOUT STX TEMP
3328 10 03 04 .BYT 16,3,4,2,5,4,4,2,4,2, 3416 8D 04 28 STA TEMP+1
      5,4,4,4,9,1,9,1 3419 A0 FE LDY #254
333A 0C 01 15 .BYT 12,1,21,1,9,1,9,1,12, 341B C8 FLOOP INY
      1,21,1,21,1,21,1,24, 341C C8 INY
      1 341D AD 03 28 LDA TEMP
334C 21 01 15 .BYT 33,1,21,1,21,1,24,1, 3420 38 SEC
      33,1 3421 F9 53 34 SBC DTAB,Y
3356 21 02 21 .BYT 33,2,33,2,33,2,33,1, 3424 AD 04 28 LDA TEMP+1
      33,2,28,1,30,1,25,1 3427 F9 54 34 SBC DTAB+1,Y
3366 1C 01 1E .BYT 28,1,30,2,21,2,23,1, 342A B0 EF BCS FLOOP
      25,1,21,1,23,1,25,1 342C 98 TYA
3376 15 01 17 .BYT 21,1,23,1,25,1,21,1, 342D 4A LSR
      23,2,21,4 342E 8D 05 28 STA TEMP+2
3382 15 02 15 .BYT 21,2,21,2,21,2,21,1, 3431 A9 04 STA #4
      21,2,16,1,18,1,13,1 3433 38 SEC
3392 10 01 12 .BYT 16,1,18,2,9,2,11,1,1 3434 ED 05 28 SBC TEMP+2
      3,1,9,1,11,1,13,1 3437 AA TAX
33A2 09 01 0B .BYT 9,1,11,1,13,1,9,1,11, 3438 F0 08 BEQ Q
      2,9,5 343A A9 20 LDA #32
33AE 00 SONG3 .BYT 0 343C 20 D2 FF JSR $FFD2
33AF 3F 54 00 SONGLEN .BYT 63,84,0 343F CA DEX
      ; which computerprint 3440 D0 F8 BNE SPOUT
      ; 3442 AE 03 28 Q LDX TEMP
33B2 HUH = * LDA $328 STA TEMP+1
33B2 AD 28 03 LDA #328 LDY COMPUTER
33B5 C9 ED CMP #$ED 344B F0 03 BEQ PLUS3
      344D 4C CD BD JMP $BDCD

```

Chapter 3

```

3450 4C 5F A4 PLUS3      JMP  $A45F
3453 0A 00 64 DTAB       .WORD 10,100,1000,10000,
                        65535
:
: draw screen
:
345D          DRAWSCREE = *
345D A9 93      LDA  #"(CLR)"
345F 20 D2 FF   JSR  $FFD2
3462 AD 12 28  LDA  COLOR
3465 85 04      STA  FREE0
3467 AD 13 28  LDA  COLOR+1
346A 85 05      STA  FREE0+1
346C A9 0F      LDA  #15
346E AE 56 03  LDX  COMPUTER
3471 D0 02      BNE  PLS1
3473 A9 6F      LDA  #111
3475 A2 04      LDX  #4
3477 A0 00      LDY  #0
3479 91 04      STA  (FREE0),Y
347B 88         DEY
347C D0 FB      BNE  COLOOP
347E E6 05      INC  FREE0+1
3480 CA         DEX
3481 D0 F6      BNE  COLOOP
3483 A9 00      LDA  #0
3485 AE 56 03  LDX  COMPUTER
3488 F0 08      BEQ  PLUS1
348A 8D 20 D0  STA  53280
348D 8D 21 D0  STA  53281
3490 F0 06      BEQ  AR1
3492 8D 15 FF  PLUS1    STA  65301
3495 8D 19 FF  AR1      STA  65305
3498 A9 13      LDA  #19
349A 8D 03 28  STA  TEMP
349D AE 03 28  DLOOP1   LDX  TEMP
34A0 A0 07      LDY  #7
34A2 18        CLC
34A3 20 F0 FF  JSR  $FFF0
34A6 A9 CB      LDA  #<STRING2
34A8 A0 35      LDY  #>STRING2
34AA 20 08 34  JSR  STROUT
34AD EE 03 28  INC  TEMP
34B0 AD 03 28  LDA  TEMP
34B3 C9 17      CMP  #23
34B5 D0 E6      BNE  DLOOP1
34B7 A2 17      LDX  #23
34B9 A0 03      LDY  #3
34BB 18        CLC
34BC 20 F0 FF  JSR  $FFF0
34BF A9 EF      LDA  #<STRING3
34C1 A0 35      LDY  #>STRING3
34C3 20 08 34  JSR  STROUT
34C6 A2 03      LDX  #3
34C8 A0 07      LDY  #7
34CA 18        CLC
34CB 20 F0 FF  JSR  $FFF0
34CE A9 10      LDA  #16
34D0 8D 03 28  STA  TEMP
34D3 A9 16      LDA  #<STRING4
34D5 A0 36      LDY  #>STRING4
34D7 20 08 34  JSR  STROUT
34DA CE 03 28  DEC  TEMP
34DD D0 F4      BNE  DLOOP2
34DF A2 00      LDX  #0
34E1 A0 05      LDY  #5
34E3 18        CLC
34E4 20 F0 FF  JSR  $FFF0
34E7 A9 2A      LDA  #<STRING7
34E9 A0 36      LDY  #>STRING7
34EB 20 08 34  JSR  STROUT
34EE A9 11      LDA  #17

34F0 8D 03 28  STA  TEMP
34F3 AE 03 28  DLOOP3   LDA  TEMP
34F6 A0 08      LDY  #8
34F8 18        CLC
34F9 20 F0 FF  JSR  $FFF0
34FC A9 1B      LDA  #<STRING5
34FE A0 36      LDY  #>STRING5
3500 20 08 34  JSR  STROUT
3503 AD 03 28  LDA  TEMP
3506 38        SEC
3507 E9 03      SBC  #3
3509 8D 03 28  STA  TEMP
350C C9 0A      CMP  #10
350E B0 E3      BCS  DLOOP3
3510 A2 0B      LDX  #11
3512 A0 19      LDY  #25
3514 18        CLC
3515 20 F0 FF  JSR  $FFF0
3518 A9 5C      LDA  #<PIANO1
351A A0 36      LDY  #>PIANO1
351C 20 08 34  JSR  STROUT
351F A2 0B      LDX  #11
3521 A0 1A      LDY  #26
3523 18        CLC
3524 20 F0 FF  JSR  $FFF0
3527 A9 78      LDA  #<PIANO2
3529 A0 36      LDY  #>PIANO2
352B 20 08 34  JSR  STROUT
352E A2 0C      LDX  #12
3530 A0 1D      LDY  #29
3532 18        CLC
3533 20 F0 FF  JSR  $FFF0
3536 A9 56      LDA  #<PIANO3
3538 A0 36      LDY  #>PIANO3
353A 20 08 34  JSR  STROUT
353D A9 0D      LDA  #13
353F 8D 03 28  STA  TEMP
3542 A2 04      DLOOP4   LDX  #4
3544 AC 03 28  LDY  TEMP
3547 20 A7 35  JSR  DRAWINDOW
354A AD 03 28  LDA  TEMP
354D 18        CLC
354E 69 0A      ADC  #10
3550 8D 03 28  STA  TEMP
3553 C9 28      CMP  #40
3555 90 EB      BCC  DLOOP4
3557 AD 5A 03  LDA  SCREEN
355A 85 04      STA  FREE0
355C AD 5B 03  LDA  SCREEN+1
355F 85 05      STA  FREE0+1
3561 A0 09      LDY  #9
3563 A2 3A      LDX  #58
3565 8A         TXA
3566 91 04      STA  (FREE0),Y
3568 C8         INY
3569 E8         INX
356A E0 56      CPX  #86
356C D0 F7      BNE  PP
356E A0 31      LDY  #49
3570 A2 56      LDX  #86
3572 8A         TXA
3573 91 04      STA  (FREE0),Y
3575 C8         INY
3576 E8         INX
3577 E0 58      CPX  #88
3579 D0 F7      BNE  PP2
357B A0 4B      LDY  #75
357D A2 59      LDX  #89
357F 8A         TXA
3580 91 04      STA  (FREE0),Y
3582 C8         INY
3583 CA         DEX

```

The Source Code

3584	E0 57	CPX	#87				3 SPACE}{DOWN}
3586	D0 F7	BNE	PP3				{3 LEFT}{5{2 SPAC
3588	A2 04	LDX	#4				E}{2 DOWN}{6 LE
358A	A0 00	LDY	#0				FT}{666666667"
358C	18	CLC		371B	00	.BYT	0
358D	20 F0 FF	JSR	\$FFFF	371C	9E 12 20 WINDOW	.ASC	"{YEL}{RVS} B {D
3590	A9 92	LDA	#<MESS1				OWN}{3 LEFT}{C
3592	A0 37	LDY	#>MESS1				+C{DOWN}{3 LEF
3594	20 08 34	JSR	STROUT				T} B {DOWN}{3 LE
3597	A2 13	LDX	#19				FT}{OFF}{3 SPAC
3599	A0 06	LDY	#6				E}"
359B	18	CLC		3737	00	.BYT	0
359C	20 F0 FF	JSR	\$FFFF	3738	9E 12 DF OWINDOW	.ASC	"{YEL}{RVS}{<->
359F	A9 B2	LDA	#<STRING1				{OFF}{RVS}{DO
35A1	A0 35	LDY	#>STRING1				WN}{3 LEFT}{OF
35A3	20 08 34	JSR	STROUT				F}{RVS}{DOWN}
35A6	60	RTS					{3 LEFT}{M{OFF
35A7	18	CLC					{RVS}{N{DOWN}
35A8	20 F0 FF	JSR	\$FFFF				{3 LEFT}{OFF}{<
35AB	A9 1C	LDA	#<WINDOW				> E"
35AD	A0 37	LDY	#>WINDOW	3759	00	.BYT	0
35AF	4C 08 34	JMP	STROUT	375A	05 12 23 GOODGUY	.ASC	"{WHT}{RVS}#\$
35B2	96 12 A9 STRING1	.ASC	"{<3>}{RVS}{DO				%{DOWN}{3 LEFT
			WN}{2 LEFT}{(SP				&{(DOWN){3 LE
			ACE}{DOWN}{3 LE				FT)}+{DOWN}{3
			FT){2 SPACE}{D				3 LEFT}{OFF}{3 SP
			OWN}{4 LEFT}{(ACE)"
			3 SPACE}"	3775	00	.BYT	0
35CA	00	.BYT	0	3776	05 12 2C BADGUY	.ASC	"{WHT}{RVS}-.{D
35CB	96 12 20 STRING2	.ASC	"{<3>}{RVS}{33 SP				OWN}{3 LEFT}/01
			ACE)"				{DOWN}{3 LEFT}2
35EE	00	.BYT	0				34{DOWN}{3 LEFT
35EF	12 B7 B7 STRING3	.ASC	"{RVS} {<37 Y>}"				{OFF}{3 SPACE}
3615	00	.BYT	0	3791	00	.BYT	0
3616	92 A5 11 STRING4	.ASC	"{OFF}{<G>}{DOW	3792	1E 53 43 MESS1	.ASC	"{GRN}SCORE{
			N}{LEFT}"				4 DOWN}{4 LEFT}
361A	00	.BYT	0				TIME{4 DOWN}{
361B	B8 B8 B8 STRING5	.ASC	"{<14 U>}"				4 LEFT}{HIGH{4 DO
3629	00	.BYT	0				WN}{4 LEFT}CARD
362A	96 12 3D STRING7	.ASC	"{<3>}{RVS}={D	378C	00	.BYT	0
			OWN}{2 LEFT}{O	378D	05 50 52 MESS2	.ASC	"{WHT}PRESS RET
			FF}{RVS}={DOW				URN TO PLAY AGAI
			N}{2 LEFT}{>{OFF				N"
			}{<20 Y>}"	37D8	00	.BYT	0
			"{<11 Y>}"	37D9	11 9D 9D MESS3	.ASC	"{DOWN}{2 LEFT}
3650	B7 B7 B7	.ASC	"{WHT}{RVS}{D				0
365B	00	.BYT	0	37DC	00	.BYT	0
365C	05 12 A9 PIANO1	.ASC	"{WHT}{RVS}{D				
			OWN}{LEFT}{DO				
			WN}{LEFT}{DOWN}				
			N}{LEFT}{DOWN}{L				
			{LEFT}{DOWN}{L				
			EFT}{DOWN}{LEF				
			T}{2 SPACE}{DOW				
			N}{2 LEFT}{2 SPA				
			CE}"				
3677	00	.BYT	0	37DD			CARDS
3678	1F 12 20 PIANO2	.ASC	"{BLU}{RVS}{9 SP	37DD	EE 50 03	INC	INDEX
			ACE}{DOWN}{9 LE	37E0	AD 50 03	LDA	INDEX
			FT){9 SPACE}{DO	37E3	C9 05	CMP	#5
			WN}{9 LEFT}"	37E5	D0 05	BNE	NOMAX2
			"{9 SPACE}{DOWN	37E7	A9 00	LDA	#0
			}9 LEFT}{9 SPAC	37E9	8D 50 03	STA	INDEX
			E}{DOWN}{9 LEFT	37EC	AD 5A 03 NOMAX2	LDA	SCREEN
			"	37EF	18	CLC	
			"{9 SPACE}{DOWN	37F0	69 82	ADC	#<386
			"	37F2	85 04	STA	FREED
			"	37F4	AD 58 03	LDA	SCREEN+1
			"	37F7	69 01	ADC	#>386
			"	37F9	85 05	STA	FREED+1
			"	37FB	20 30 2D	.JSR	ADDCOLOR
			"	37FE	A0 09	LDY	#9
			"	3800	AD 56 03	LDA	COMPUTER
			"	3803	D0 02	BNE	SY
36EC	20 20 20	.ASC	"{9 SPACE}"	3805	A9 71	LDA	#113
36F5	00	.BYT	0	3807	91 02	STA	(TEMPO),Y
36F6	05 12 20 PIANO3	.ASC	"{WHT}{RVS} 5 {D	3809	88	DEY	
			OWN}{3 LEFT}{				

Chapter 3

380A	10	FB	BPL	SY	38AB	95 55 95	.BYT	149,85,149,149,149,85,101,169
380C	A0	31	LDY	#49				
380E	91 02	SZ	STA	(TEMPO),Y	38B3	6A 9A 6A	.BYT	106,154,106,170,10
3810	88		DEY					6,86,85,86
3811	C0 27		CPY	#39	38B8	6C FF 7F	.BYT	108,255,127,126,25
3813	D0 F9		BNE	SZ				4,255,127,102
3815	A9 00		LDA	#0	38C3	0C 33 C0	.BYT	12,51,192,76,115,64
3817	8D 05 28		STA	TEMP3				,127,127
381A	A9 03		LDA	#3	38CB	C0 30 C0	.BYT	192,48,192,212,17,1
381C	8D 04 28		STA	TEMP2				7,209,209
381F	AD 50 03		LDA	INDEX	38D3	7F 7F 7F	.BYT	127,127,127,127,12
3822	8D 03 28		STA	TEMP				7,127,127,21
3825	AE 03 28	CD1	LDA	TEMP	38DB	D1 D1 D5	.BYT	209,209,213,212,20
3828	BD 3C 03		LDA	CARDS2,X				8,208,208,64
382B	10 07		BPL	OV1	38E3	0C 00 C0	.BYT	12,0,192,12,67,0,11
382D	29 7F		AND	#127				5,67
382F	9D 3C 03		STA	CARDS2,X	38EB	00 00 C0	.BYT	0,0,192,196,1,17,20
3832	A9 18		LDA	#24				9,0
3834	AA	OV1	TAX		38F3	30 40 43	.BYT	48,64,67,51,76,15,6
3835	AC 05 28		LDY	TEMP3				0,21
3838	20 B3 2C		JSR	PUT3	38FB	D1 11 04	.BYT	209,17,4,192,208,16
383B	AD 05 28		LDA	TEMP3				,0,0
383E	18		CLC		3903	7F E7 C3	.BYT	127,231,195,153,15
383F	69 03		ADC	#3				3,129,153,153
3841	8D 05 28		STA	TEMP3	390B	FE FB F1	.BYT	254,251,241,251,23
3844	EE 03 28		INC	TEMP				9,223,191,127
3847	AD 03 28		LDA	TEMP	3913	FE FD FB	.BYT	254,253,251,247,22
384A	C9 05		CMP	#5				3,143,223,127
384C	D0 05		BNE	NOMAX	391B	E7 C3 99	.BYT	231,195,153,153,12
384E	A9 00		LDA	#0				9,153,153,254
3850	8D 03 28		STA	TEMP	3923	7F 99 93	.BYT	127,153,147,135,13
3853	CE 04 28	NOMAX	DEC	TEMP2				5,147,153,255
3856	D0 CD		BNE	CD1	392B	FE FB F1	.BYT	254,251,241,251,23
3858	20 FB 2F		JSR	RAND				9,223,191,127
3858	29 03		AND	#3	3933	FE FD FB	.BYT	254,253,251,247,22
385D	AA		TAX					3,143,223,127
385E	BD 3C 03		LDA	CARDS2,X	393B	FF 99 93	.BYT	255,153,147,135,13
3861	C9 1C		CMP	#28				5,147,153,254
3863	FO 05		BEQ	NP2	3943	7F 81 9D	.BYT	127,129,157,157,15
3865	09 80		ORA	#128				7,149,129,251
3867	9D 3C 03		STA	CARDS2,X	394B	FE FB F1	.BYT	254,251,241,251,23
386A	60	NP2	RTS					9,223,191,127
386B	04 08 0C	CARDS3	.BYT	4,8,12,16,20	3953	FE FD FB	.BYT	254,253,251,247,22
3870	98	FIXCARDS	TYA					3,143,223,127
3871	48		PHA		395B	7F 81 9D	.BYT	127,129,157,157,15
3872	8A		TXA					7,149,129,251
3873	48		PHA		3963	7F E1 F3	.BYT	127,225,243,243,24
3874	A0 04		LDY	#4				3,243,179,131
3876	A9 00		LDA	#0	396B	FE FB F1	.BYT	254,251,241,251,23
3878	99 3C 03	PUF	STA	CARDS2,Y				9,223,191,127
387B	88		DEY		3973	FE FD FB	.BYT	254,253,251,247,22
387C	10 FA		BPL	PUF				3,143,223,127
387E	A0 04		LDY	#4	397B	7F E1 F3	.BYT	127,225,243,243,24
3880	20 FB 2F	PUTCARD	JSR	RAND				3,243,179,131
3883	C9 05		CMP	#5	3983	FF 73 6D	.BYT	255,115,109,109,10
3885	B0 F9		BCS	PUTCARD				9,109,051,251
3887	AA		TAX		398B	FE FB F1	.BYT	254,251,241,251,23
3888	BD 3C 03		LDA	CARDS2,X				9,223,191,127
3888	D0 F3		BNE	PUTCARD	3993	FE FD FB	.BYT	254,253,251,247,22
388D	B9 6B 38		LDA	CARDS3,Y				3,143,223,127
3890	9D 3C 03		STA	CARDS2,X	399B	FF FF FF	.BYT	255,255,255,255,25
3893	88		DEY					5,255,255,255
3894	10 EA		BPL	PUTCARD	39A3	FF 73 6D	.BYT	255,115,109,109,10
3896	68		PLA					9,109,51,254
3897	AA		TAX		39AB	FF 80 80	.BYT	255,128,128,128,12
3898	68		PLA					8,128,128,128
3899	A8		TAY		39B3	FF 80 80	.BYT	255,128,128,128,12
389A	60		RTS					8,128,128,128
389B		CHARDEF	=	*	39BB	FF 00 63	.BYT	255,0,99,247,255,25
389B	96 96 96		.BYT	150,150,150,150,15				5,255,255
				0,150,150,150	39C3	FF 01 01	.BYT	255,1,1,129,129,129
38A3	96 96 96		.BYT	150,150,150,150,15				,129,129
				0,150,150,150				

The Source Code

39CB	80 84 83	.BYT	128,132,131,129,12 9,129,129,129	3B03	55 40 44	.BYT	85,64,68,68,68,64,8 5,0
39D3	FF FF FF	.BYT	255,255,255,0,34,34 ,8,8	3B0B	55 45 41	.BYT	85,69,65,64,68,69,8 5,0
39DB	81 91 E1	.BYT	129,145,225,65,65,6 5,65,65	3B13	50 14 15	.BYT	80,20,21,21,21,20,8 0,0
39E3	81 81 80	.BYT	129,129,128,159,19 1,255,255,255	3B1B	15 50 51	.BYT	21,80,81,84,85,80,2 1,0
39EB	41 3E 80	.BYT	65,62,128,127,128,2 55,255,255	3B23	55 11 51	.BYT	85,17,81,80,17,17,8 5,0
39F3	41 41 81	.BYT	65,65,129,121,125,2 55,255,255	3B2B	55 10 11	.BYT	85,16,17,17,17,16,8 5,0
39FB	FF 80 80	.BYT	255,128,128,129,12 9,129,129,129	3B33	55 10 11	.BYT	85,16,17,17,17,16,8 5,0
3A03	FF 63 F7	.BYT	255,99,247,136,128, 128,128,128	3B3B	55 10 14	.BYT	85,16,20,20,20,20,8 5,0
3A0B	FF 01 81	.BYT	255,1,129,193,193,1 93,193,193	3B43	55 10 51	.BYT	85,16,81,81,81,80,8 5,0
3A13	91 8F 81	.BYT	145,143,129,129,12 9,129,131,129	3B4B	55 11 11	.BYT	85,17,17,17,17,16,8 5,0
3A1B	FF FF 00	.BYT	255,255,0,99,66,8,2 55,255	3B53	55 10 14	.BYT	85,16,20,20,20,20,8 5,0
3A23	C5 F9 41	.BYT	197,249,65,65,65,81 ,249,193	3B5B	51 15 55	.BYT	81,21,85,85,81,81,6 4,0
3A2B	80 9F A0	.BYT	128,159,160,192,19 2,192,192,255	3B63	40 50 50	.BYT	64,80,80,80,86,86,2 2,0
3A33	FF 7F BE	.BYT	255,127,190,28,12,4 ,0,255	3B6B	00 00 00	.BYT	0,0,0,0,170,255,170, 0
3A3B	81 79 85	.BYT	129,121,133,3,3,3,3, 255	3B73	00 00 00	.BYT	0,0,0,0,170,255,170, 0
3A43	FF FF FF	.BYT	255,255,255,231,23 1,255,255,255	3B7B	00 00 00	.BYT	0,0,0,0,170,255,170, 0
3A4B	63 B1 D8	.BYT	99,177,216,239,247, 251,253,254	3B83	00 00 00	.BYT	0,0,128,170,255,1 91,186
3A53	00 80 C0	.BYT	0,128,192,224,240,2 48,252,254	3B8B	00 00 08	.BYT	0,0,8,32,128,224,24 8,254
3A5B	1C 8E C7	.BYT	28,142,199,224,240, 248,252,254	3B93	BE BE BE	.BYT	190,190,190,190,19 0,190,190,170
3A63	E3 71 38	.BYT	227,113,56,31,15,7, 3,1	3B9B	B8 20 00	.BYT	184,32,0,0,0,0,0,0 0,190,190,170
3A6B	00 00 00	.BYT	0,0,0,0,0,0,0,128	3BA3	BE BE BE	.BYT	190,190,190,190,19 0,190,190,170
3A73	01 00 00	.BYT	1,0,0,0,0,0,0,0	3BAB	2E 08 00	.BYT	46,8,0,0,0,0,0,0
3A7B	00 00 18	.BYT	0,0,24,60,60,60,60,1 26	3BB3	B8 20 00	.BYT	184,32,0,0,0,0,0,0
3A83	C0 E0 70	.BYT	192,224,112,56,28,1 4,7,3	3BBB	BE BE BE	.BYT	190,190,190,190,19 0,190,190,170
3A8B	FF FF C0	.BYT	255,255,192,192,19 2,192,192,192	3BC3	STACK1	=	*
3A93	00 00 00 MOUS	.BYT	0,0,0,15,31,255,8,0	3BC3	STACK2	=	*+20
3A9B	00 00 00	.BYT	0,0,0,144,248,252,3 2,0	3BC3	STACK3	=	*+40
3AA3	00 00 00	.BYT	0,0,0,9,31,63,4,0	3BC3	STACK4	=	*+60
3AAB	00 00 00	.BYT	0,0,0,240,248,255,1 6,0	3BC3	STACK5	=	*+80
3AB3	00 00 08 NEWDEF	.BYT	0,0,8,2,2,11,47,191				
3ABB	00 00 00	.BYT	0,0,0,2,170,255,254, 174				
3AC3	00 00 00	.BYT	0,0,0,0,170,255,170, 0				
3ACB	00 00 00	.BYT	0,0,0,0,170,255,170, 0				
3AD3	00 00 00	.BYT	0,0,0,0,170,254,170, 0				
3ADB	14 15 15	.BYT	20,21,21,21,85,84,8 0,0				
3AE3	55 40 45	.BYT	85,64,69,81,84,64,2 1,0				
3AEB	55 51 44	.BYT	85,81,68,64,68,68,8 5,0				
3AF3	55 45 45	.BYT	85,69,69,69,69,64,8 5,0				
3AFB	55 40 44	.BYT	85,64,68,68,68,64,8 5,0				

Prisonball

Source code by Kevin Mykytyn

```

0801          * = 2049
:
: prison ball
:
0801 DELX      = 703
0801 DELY      = 711
0801 XPOS      = 719
0801 XPOSH     = 727
0801 YPOS      = 735
0801 YOSL      = 743
0801 XOSL      = 751
0801 HIGHX1    = 760
0801 HIGHX2    = 761
0801 HIGHY     = 762
0801 BALLL     = 763
0801 BALLH     = 764
0801 JOYL      = 765
0801 JOYH      = 766
0801 MSB       = 2
0801 TEMP      = 6
0801 TOPLIMIT  = 59
0801 BOTLIMIT  = 220
0801 TOP       = 49
0801 BOT       = 229
0801 LEFT      = 20
0801 RIGHT     = 68
0801 TOPEDEGE = 87
0801 BOTEDEGE = 190
0801 BORDER    = 0
0801 BACKGROUN = 11
0801 WALLCOLOR = 15
0801 FREEO     = 251
0801 COLORO    = 253
0801 COLOR     = 249
0801 PFLAG     = 247
0801 LEFTPAD   = 27
0801 RIGHTPAD  = 61
0801 FF FF OA  .BYTE 255,255,10,0,158
0806 32 30 36  .ASC "2061"
080A 00 00 00  .BYTE 0,0,0
080D A9 00     LDA #0
080F 8D BA OF  STA FLAG
0812          =
0812 20 E4 OD  JSR SETSPRITE
0815 20 BB OA  JSR INIT
0818 20 CA OD  JSR SETRAST
081B 20 16 OE  JSR SETPOS
081E 20 39 OE  JSR SETSCREEN
0821 20 3F O9  JSR SCROUT
0824 AD BA OF  LDA FLAG
0827 D0 08     BNE MAIN
0829 A9 01     LDA #1
082B 8D BA OF  STA FLAG
082E 4C 65 08  JMP GAMEOVER
0831 CE FD 02  MAIN DEC
0834 D0 0C     BNE NOJOY
0836 AD FE 02  LDA
0839 8D FD 02  STA
083C 20 E9 0C  JSR
083F 20 BE 08  JSR
0842 CE FB 02  NOJOY DEC
0845 D0 0C     BNE

0847 AD FC 02  LDA
084A 8D FB 02  STA
084D 20 C3 0B  JSR
0850 20 9B 09  JSR
0853 20 E1 FF  NOBALL JSR
0856 D0 D9     BNE
0858 AD 00 DC  WTBT  LDA
085B 2D 01 DC  AND
085E 29 10     AND
0860 D0 F6     BNE
0862 4C 31 08  JMP

:
: the game is over
:
0865          GAMEOVER = *
0865 AD 00 DC  DEBOUNCE LDA 56320
0868 2D 01 DC  AND 56321
086B 29 1F     AND #31
086D C9 1F     CMP #31
086F D0 F4     BNE DEBOUNCE
0871 A9 B0     LDA #176
0873 8D 14 04  STA 1044
0876 8D 16 04  STA 1046
0879 8D 17 04  STA 1047
087C A9 03     LDA #3
087E 8D 15 D0  STA 53269
0881 A2 18     LDY #24
0883 A0 0C     LDY #12
0885 18        CLC
0886 20 F0 FF  JSR $FFF0
0889 A9 7F     LDA #-<MESS3
088B A0 0F     LDY #->MESS3
088D 20 1E AB  JSR $AB1E
0890 78        SEI
0891 A9 00     LDA #0
0893 85 C6     STA 198
0895 AD 00 DC  WTBT  LDA 56320
0898 2D 01 DC  AND 56321
089B 29 10     AND #16
089D F0 0B     BEQ JOYPLAY
089F AD 00 DC  LDA 56320
08A2 C9 7F     CMP #127
08A4 F0 EF     BEQ WTBT
08A6 A9 00     LDA #0
08A8 F0 02     BEQ STORIT
08AA A9 01     LDA #1
08AC 85 F7     STA PFLAG
08AE 0A        ASL
08AF 0A        ASL
08B0 0A        ASL
08B1 18        CLC
08B2 69 08     ADC #8
08B4 8D FB 02  STA BALLL
08B7 8D FC 02  STA BALLH
08BA 58        CLI
08BB 4C 12 08  JMP START

:
: decrement the timer
:
08BE          TICK = *
08BE A5 A2     LDA 162
08C0 C9 3C     CMP #60

```


The Source Code

```

08C2 D0 37      BNE  N1      ; output a number
08C4 A9 00      LDA  #0
08C6 B5 A2      STA  162
08C8 AD 17 04   LDA  1047
08CB C9 B0      CMP  #176
08CD F0 06      BEQ  ZU1
08CF CE 17 04   DEC  1047
08D2 4C FB 08   JMP  N1
08D5 A9 B9      LDA  #185
08D7 8D 17 04   STA  1047
08DA AD 16 04   LDA  1046
08DD C9 B0      CMP  #176
08DF F0 06      BEQ  Z2
08E1 CE 16 04   DEC  1046
08E4 4C FB 08   JMP  N1
08E7 A9 B5      LDA  #181
08E9 8D 16 04   STA  1046
08EC AD 14 04   LDA  1044
08EF C9 B0      CMP  #176
08F1 D0 05      BNE  NDONE
08F3 68        PLA
08F4 68        PLA
08F5 4C 65 08   JMP  GAMEOVER
08F8 CE 14 04   DEC  1044
08FB AD 14 04   LDA  1044
08FE C9 B0      CMP  #176
0900 D0 12      BNE  REGCOL
0902 AD 16 04   LDA  1046
0905 C9 B2      CMP  #178
0907 D0 0B      BNE  REGCOL
0909 AD 17 04   LDA  1047
090C C9 B0      CMP  #176
090E D0 04      BNE  REGCOL
0910 EE 20 D0   INC  53280
0913 60        RTS
0914 A9 00      LDA  #0
0916 8D 20 D0   STA  53280
0919 60        RTS

; add to the score
091A          ADDSCORE = *
091A A6 06      LDX  TEMP
091C BD 27 D0   LDA  53287,X
091F 29 0F      AND  #15
0921 C9 03      CMP  #3
0923 F0 19      BEQ  NSC
0925 AA        TAX
0926 AC B8 0F   LDY  SCORED
0929 B9 4E 0F   LDA  SCORES,Y
092C 18        CLC
092D 7D B4 0F   ADC  SCORE1L,X
0930 9D B4 0F   STA  SCORE1L,X
0933 BD B6 0F   LDA  SCORE1H,X
0936 69 00      ADC  #0
0938 9D B6 0F   STA  SCORE1H,X
093B 20 3F 09   JSR  SCROUT
093E 60        NSC

; output score
093F          SCROUT = *
093F AO 07      LDY  #7
0941 AE B5 0F   LDX  SCORE2L
0944 AD B7 0F   LDA  SCORE2H
0947 20 56 09   JSR  NUMOUT
094A AO 22      LDY  #34
094C AE B4 0F   LDX  SCORE1L
094F AD B6 0F   LDA  SCORE1H
0952 20 56 09   JSR  NUMOUT
0955 60        RTS

; collision
0956          NUMOUT = *
0956 8C B9 0F   STY  INDEX
0959 86 FB      STX  FREE0
095B 85 FC      STA  FREE0+1
095D A2 08      LDX  #8
095F A0 FF      LDY  #255
0961 C8        NUMLOOP1 SUBAGAIN INVY
0962 A5 FB      LDA  FREE0
0964 48        PHA
0965 38        SEC
0966 FD 91 09   SBC  DIGITS,X
0969 85 FB      STA  FREE0
096B A5 FC      LDA  FREE0+1
096D 48        PHA
096E FD 92 09   SBC  DIGITS+1,X
0971 85 FC      STA  FREE0+1
0973 90 05      BCC  DONESUB
0975 68        PLA
0976 68        PLA
0977 4C 61 09   JMP  SUBAGAIN
097A 68        DONESUB PLA
097B 85 FC      STA  FREE0+1
097D 68        PLA
097E 85 FB      STA  FREE0
0980 98        TYA
0981 09 B0      ORA  #176
0983 AC B9 0F   LDY  INDEX
0986 99 00 04   STA  1024,Y
0989 EE B9 0F   INC  INDEX
098C CA        DEX
098D CA        DEX
098E 10 CF      BPL  NUMLOOP1
0990 60        RTS
0991 01 00 0A   DIGITS .WORD 1,10,100,1000,10000

; collision
099B          COLLISION = *
099B A2 04      LDX  #4
099D 86 06      STX  TEMP
099F BD DF 02   LDA  YPOS,X
09A2 38        SEC
09A3 E9 2A      SBC  #42
09A5 29 F8      AND  #248
09A7 48        PHA
09A8 85 FB      STA  FREE0
09AA A9 00      LDA  #0
09AC 85 FC      STA  FREE0+1
09AE 06 FB      ASL  FREE0
09B0 26 FC      ROL  FREE0+1
09B2 06 FB      ASL  FREE0
09B4 26 FC      ROL  FREE0+1
09B6 68        PLA
09B7 18        CLC
09B8 65 FB      ADC  FREE0
09BA 85 FB      STA  FREE0
09BC A5 FC      LDA  FREE0+1
09BE 69 00      ADC  #0
09C0 85 FC      STA  FREE0+1
09C2 BD D7 02   LDA  XPOSH,X
09C5 85 FE      STA  COLOR0+1
09C7 BD CF 02   LDA  XPOS,X
09CA 85 FD      STA  COLOR0
09CC AO 03      LDY  #3
09CE 46 FE      LSR  COLOR0+1
09D0 66 FD      ROR  COLOR0
09D2 88        DEY
09D3 D0 F9      BNE  DIV
09D5 A5 FD      LDA  COLOR0

```

Chapter 3

09D7	38	SEC	0A63	20 65 0C	JSR	REVX
09D8	E9 02	SBC	0A66	4C 6B 0A	JMP	NEXT
09DA	85 FD	STA	0A69	A6 06	LDX	TEMP
09DC	A5 FE	LDA	0A6B	CA	NOG	DEX
09DE	E9 00	SBC	0A6C	E0 01	NEXT	CPX
09E0	85 FE	STA	0A6E	F0 03		BEQ
09E2	A5 FB	LDA	0A70	4C 9D 09		JMP
09E4	18	CLC	0A73	60	FINCOLL	RTS
09E5	65 FD	ADC				
09E7	85 FB	STA				
09E9	A5 FC	LDA				
09EB	65 FE	ADC	0A74		CLICK	=
09ED	85 FC	STA	0A74	48		PHA
09EF	A5 FC	LDA	0A75	8A		TXA
09F1	18	CLC	0A76	38		SEC
09F2	69 D8	ADC	0A77	E9 02		SBC
09F4	85 FC	STA	0A79	A8		TAY
09F6	A0 00	LDY	0A7A	B9 B8 0A		LDA
09F8	B1 FB	LDA	0A7D	A8		TAY
09FA	29 0F	AND	0A7E	A9 1E		LDA
09FC	A2 03	LDX	0A80	99 01 D4		STA
09FE	CA	DEX	0A83	A9 11		LDA
09FF	30 68	BMI	0A85	99 05 D4		STA
0A01	DD 40 0F	CMP	0A88	A9 80		LDA
0A04	DD F8	BNE	0A8A	99 04 D4		STA
0A06	8E B8 0F	STX	0A8D	A9 81		LDA
0A09	A5 FC	LDA	0A8F	99 04 D4		STA
0A0B	38	SEC	0A92	68		PLA
0A0C	E9 D4	SBC	0A93	60		RTS
0A0E	85 FC	STA				
0A10	B1 FB	LDA				
0A12	A2 03	LDX				
0A14	CA	DEX	0A94		BONG	=
0A15	30 52	BMI	0A94	48		PHA
0A17	DD 45 0F	CMP	0A95	8A		TXA
0A1A	DD F8	BNE	0A96	38		SEC
0A1C	8A	TXA	0A97	E9 02		SBC
0A1D	0A	ASL	0A99	A8		TAY
0A1E	A8	TAY	0A9A	B9 B8 0A		LDA
0A1F	A6 06	LDX	0A9D	A8		TAY
0A21	A5 FD	LDA	0A9E	68		PLA
0A23	38	SEC	0A9F	0A		ASL
0A24	E9 0B	SBC	0AA0	0A		ASL
0A26	4A	LSR	0AA1	0A		ASL
0A27	4A	LSR	0AA2	18		CLC
0A28	DD 96 0F	CMP	0AA3	69 0A		ADC
0A2B	F0 3C	BEQ	0AA5	99 01 D4		STA
0A2D	9D 96 0F	STA	0AA8	A9 13		LDA
0A30	BD BF 02 GOON	LDA	0AAA	99 05 D4		STA
0A33	10 01	BPL	0AAD	A9 20		LDA
0A35	C8	INY	0AAF	99 04 D4		STA
0A36	B9 48 0F FNE	LDA	0AB2	A9 21		LDA
0A39	A0 00	LDY	0AB4	99 04 D4		STA
0A3B	91 FB	STA	0AB7	60		RTS
0A3D	20 74 0A	JSR	0AB8	00 07 0E SEVENS		.BYTE
0A40	C9 20	CMP				
0A42	DD 1A	BNE				
0A44	A5 FD	LDA				
0A46	38	SEC				
0A47	E9 0B	SBC	0ABB		INIT	=
0A49	4A	LSR	0ABB	A9 00		LDA
0A4A	4A	LSR	0ABD	A0 17		LDY
0A4B	AA	LSR	0ABF	99 00 D4 CLSID		STA
0A4C	FE AA 0F	TAX	0AC2	88		DEY
0A4F	BD AA 0F	INC	0AC3	10 FA		BPL
0A52	C9 2E	LDA	0AC5	A9 0F		LDA
0A54	DD 08	CMP	0AC7	8D 18 D4		STA
0A56	A9 00	BNE	0ACA	A9 FF		LDA
0A58	9D AA 0F	LDA	0ACC	8D 0F D4		STA
0A5B	20 9C 0C	STA	0ACF	A9 80		LDA
0A5E	20 1A 09 NOCOUNT	JSR	0AD1	8D 12 D4		STA
0A61	A6 06	JSR	0AD4	A9 14		LDA
		LDX	0AD6	8D FD 02		STA
		TEMP				JOYL

The Source Code

```

OAD9 8D FE 02          STA JOYH
OADC A9 00             LDA #0
OADE 85 A2             STA 162
OAE0 A0 04             LDA #4
OAE2 99 AA 0F Z1      STA COUNTERS,Y
OAE5 99 B4 0F         STA SCORE1L,Y
OAE8 88                DEY
OAE9 10 F7            BPL Z1
OAEB A0 02            LDY #2
OAEF A2 4B WP         LDY #75
OAF2 10 02            LDA $D41B
OAF4 A2 B4             BPL NOD
OAF6 8A                LDY #180
OAF7 99 C1 02         TXA
OAF8 A9 06             STA DELX+2,Y
OAF9 99 98 0F         LDA #6
OAFB 88                STA SAVL+2,Y
OAFD 88                DEY
OAE0 10 EB            BPL WP
OAE2 60                RTS

; see if ball should wrap

OB03                WRAP = *
OB03 BD D7 02         LDA XPOSH,X
OB06 D0 17            BNE R13
OB08 BD CF 02         LDA XPOS,X
OB0B C9 0A            CMP #10
OB0D D0 26            BNE NOWRAP
OB0F A9 14            LDA #20
OB11 9D 96 0F         STA SAVL,X
OB14 A9 4D            LDA #77
OB16 9D CF 02         STA XPOS,X
OB19 A9 01            LDA #1
OB1B 9D D7 02         STA XPOSH,X
OB1E 60                RTS
OB1F BD CF 02 R13     LDA XPOS,X
OB22 C9 4E            CMP #78
OB24 D0 0F            BNE NOWRAP
OB26 A9 14            LDA #20
OB28 9D 96 0F         STA SAVL,X
OB2B A9 0B            LDA #11
OB2D 9D CF 02         STA XPOS,X
OB30 A9 00            LDA #0
OB32 9D D7 02         STA XPOSH,X
OB35 60                RTS

; see if hit paddles

OB36                HITPADDLE = *
OB36 BD D7 02         LDA XPOSH,X
OB39 D0 2E            BNE RIPAD
OB3B BD BF 02         LDA DELX,X
OB3E 10 28            BPL NOTLEFT
OB40 BD CF 02         LDA XPOS,X
OB43 C9 1B            CMP #LEFTPAD
OB45 F0 04            BEQ MAYBE1
OB47 C9 45            CMP #LEFTPAD+42
OB49 D0 1D            BNE NOTLEFT
OB4B AD DF 02 MAYBE1 LDA YPOS
OB4E 38                SEC
OB4F FD DF 02         SBC YPOS,X
OB52 C9 F5            CMP #245
OB54 B0 04            BCS HITPAD1
OB56 C9 0F            CMP #15
OB58 B0 0E            BCS NOTLEFT
OB5A 20 9C 0B HITPAD1 JSR NEWY
OB5D A9 01            LDA #1
OB5F 9D 27 D0         STA 53287,X
OB62 20 94 0A         JSR BONG
OB65 4C 93 0B         JMP NOY
OB68 60                NOTLEFT

OB69 BD BF 02 RIPAD  LDA DELX,X
OB6C 30 FA            BMI NOTLEFT
OB6E BD CF 02         LDA XPOS,X
OB71 C9 3D            CMP #RIGHTPAD
OB73 F0 04            BEQ MAYBE2
OB75 C9 13            CMP #RIGHTPAD-42
OB77 D0 EF            BNE NOTLEFT
OB79 AD E0 02 MAYBE2 LDA YPOS+1
OB7C 38                SEC
OB7D FD DF 02         SBC YPOS,X
OB80 C9 F5            CMP #245
OB82 B0 04            BCS HITPAD2
OB84 C9 0F            CMP #15
OB86 B0 E0            BCS NOTLEFT
OB88 20 9C 0B HITPAD2 JSR NEWY
OB8B A9 00            LDA #0
OB8D 9D 27 D0         STA 53287,X
OB90 20 94 0A         JSR BONG
OB93 A9 14            LDA #20
OB95 9D 96 0F         STA SAVL,X
OB98 20 65 0C         JSR REVX
OB9B 60                RTS
OB9C C9 00            CMP #0
OB9E D0 0B            BNE HANDLE
OBA0 AD 04 DC         LDA 56324
OBA3 10 04            BPL PLUS
OBA5 A9 FF            LDA #255
OBA7 D0 02            BNE HANDLE
OBA9 A9 01            LDA #1
OBAB 20 AB 0E HANDLE JSR NEGATE
OBAE 08                PHP
OBAF 10 03            BPL FE1
OBB1 20 AB 0E         JSR NEGATE
OBB4 0A                ASL
OBB5 0A                ASL
OBB6 18                CLC
OBB7 69 0A            ADC #10
OBB9 28                PLP
OBBA 10 03            BPL FE2
OBBC 20 AB 0E         JSR NEGATE
OBBF 9D C7 02 FE2    STA DELY,X
OBC2 60                RTS

; move the balls

OBC3                MOVBALLS = *
OBC3 A2 04            LDY #4
OBC5 20 03 0B MOV1   JSR WRAP
OBC8 20 17 0C         JSR ADDX
OBCB 20 45 0C         JSR ADDY
OBCE C9 31            CMP #TOP
OBD0 F0 04            BEQ DOREVY
OBD2 C9 E5            CMP #BOT
OBD4 D0 03            BNE DONTREVV
OBD6 20 83 0C DOREVY JSR REVY
OBD9 BD BF 02 DONTREVV LDA DELX,X
OBDC 10 0E            BPL RIGHTSIDE
OBDE BD CF 02         LDA XPOS,X
OBE1 C9 14            CMP #LEFT
OBE3 D0 26            BNE DONTREVV
OBE5 BD D7 02         LDA XPOSH,X
OBE8 D0 21            BNE DONTREVV
OBEA F0 0C            BEQ DOREVX
OBEF BD CF 02 RIGHTSIDE LDA XPOS,X
OBF1 D0 18            CMP #RIGHT
OBF3 BD D7 02         BNE DONTREVV
OBF6 F0 13            LDA XPOSH,X
OBF8 BD DF 02 DOREVX  LDA YPOS,X
OBFB C9 57            CMP #TOPEDGE
OBFD 90 04            BCC DOREVX2
OBF8 C9 BE            CMP #BOTEDGE

```


The Source Code

<p>OD12 60</p> <p>... paddle read routine</p> <p>OD13 PADDLES = *</p> <p>OD13 78 SEI</p> <p>OD14 AD 02 DC LDA \$DC02</p> <p>OD17 48 PHA</p> <p>OD18 A9 C0 LDA #\$C0</p> <p>OD1A 8D 02 DC STA \$DC02</p> <p>OD1D A9 80 LDA #\$80</p> <p>OD1F 8D 00 DC STA \$DC00</p> <p>OD22 A0 80 LDY #\$80</p> <p>OD24 EA DELK NOP</p> <p>OD25 88 DEY</p> <p>OD26 D0 FC BNE DELK</p> <p>OD28 A2 01 LDX #1</p> <p>OD2A BD 19 D4 PLOP LDA \$D419,X</p> <p>OD2D C9 3B CMP #TOPLIMIT</p> <p>OD2F B0 04 BCS NO1</p> <p>OD31 A9 3B LDA #TOPLIMIT</p> <p>OD33 D0 06 BNE NO2</p> <p>OD35 C9 DC NO1 CMP #BOTLIMIT</p> <p>OD37 90 02 BCC NO2</p> <p>OD39 A9 DC LDA #BOTLIMIT</p> <p>OD3B 48 NO2 PHA</p> <p>OD3C 8A TXA</p> <p>OD3D 0A ASL</p> <p>OD3E 0A ASL</p> <p>OD3F 0A ASL</p> <p>OD40 18 CLC</p> <p>OD41 69 BD ADC #<STACK</p> <p>OD43 85 FB STA FREE0</p> <p>OD45 A9 0F LDA #>STACK</p> <p>OD47 69 00 ADC #0</p> <p>OD49 85 FC STA FREE0+1</p> <p>OD4B BD BB 0F LDA STACKPT,X</p> <p>OD4E A8 TAY</p> <p>OD4F 68 PLA</p> <p>OD50 91 FB STA (FREE0),Y</p> <p>OD52 C0 07 CPY #7</p> <p>OD54 D0 07 BNE NOTOP</p> <p>OD56 A9 00 LDA #0</p> <p>OD58 9D BB 0F STA STACKPT,X</p> <p>OD5B F0 03 BEQ SKIPINC</p> <p>OD5D FE BB 0F NOTOP INC STACKPT,X</p> <p>OD60 A9 00 SKIPINC LDA #0</p> <p>OD62 85 FD STA COLOR0</p> <p>OD64 85 FE STA COLOR0+1</p> <p>OD66 A0 07 LDY #7</p> <p>OD68 A5 FD ADDLOOP LDA COLOR0</p> <p>OD6A 18 CLC</p> <p>OD6B 71 FB ADC (FREE0),Y</p> <p>OD6D 85 FD STA COLOR0</p> <p>OD6F A5 FE LDA COLOR0+1</p> <p>OD71 69 00 ADC #0</p> <p>OD73 85 FE STA COLOR0+1</p> <p>OD75 88 DEY</p> <p>OD76 10 F0 BPL ADDLOOP</p> <p>OD78 A5 FD LDA COLOR0</p> <p>OD7A 46 FE LSR COLOR0+1</p> <p>OD7C 6A ROR</p> <p>OD7D 46 FE LSR COLOR0+1</p> <p>OD7F 6A ROR</p> <p>OD80 46 FE LSR COLOR0+1</p> <p>OD82 6A ROR</p> <p>OD83 9D DF 02 STA YPOS,X</p> <p>OD86 CA DEX</p> <p>OD87 10 A1 BPL PLOP</p> <p>OD89 68 PLA</p> <p>OD8A 8D 02 DC STA \$DC02</p> <p>OD8D 58 CLI</p>		<p>OD8E 60</p> <p>... sprite movement routine</p> <p>OD8F SPRMOV = *</p> <p>OD8F A9 01 LDA #1</p> <p>OD91 8D 19 D0 STA \$D019</p> <p>OD94 A2 0E LDX #14</p> <p>OD96 A0 07 LDY #7</p> <p>OD98 A9 00 LDA #0</p> <p>OD9A 85 02 STA MSB</p> <p>OD9C B9 CF 02 SPRLOOP LDA XPOS,Y</p> <p>OD9F 9D 00 D0 STA 53248,X</p> <p>ODA2 B9 DF 02 LDA YPOS,Y</p> <p>ODA5 9D 01 D0 STA 53249,X</p> <p>ODA8 B9 D7 02 LDA XPOSH,Y</p> <p>ODAB 4A LSR</p> <p>ODAC 26 02 ROL MSB</p> <p>ODAE CA DEX</p> <p>ODAF CA DEX</p> <p>ODB0 88 DEY</p> <p>ODB1 10 E9 BPL SPRLOOP</p> <p>ODB3 A5 02 LDA MSB</p> <p>ODB5 8D 10 D0 STA 53248,X</p> <p>ODB8 A9 FA LDA #250</p> <p>ODBA 8D 12 D0 STA \$D012</p> <p>ODBD AD 0D DC LDA \$DC0D</p> <p>ODC0 29 01 AND #1</p> <p>ODC2 F0 03 BEQ SKIP</p> <p>ODC4 4C 31 EA JMP \$EA31</p> <p>ODC7 4C BC FE SKIP JMP \$FEBC</p> <p>... set up raster interrupt</p> <p>ODCA SETRAST = *</p> <p>ODCA A9 7F LDA #127</p> <p>ODCC 8D 0D DC STA 56333</p> <p>ODCF A9 8F LDA #<SPRMV</p> <p>ODD1 8D 14 03 STA \$0314</p> <p>ODD4 A9 0D LDA #>SPRMV</p> <p>ODD6 8D 15 03 STA \$0315</p> <p>ODD9 A9 18 LDA #27</p> <p>ODDB 8D 11 D0 STA 53265</p> <p>ODDE A9 81 LDA #129</p> <p>ODE0 8D 1A D0 STA 53274</p> <p>ODE3 60 RTS</p> <p>... set up sprites</p> <p>ODE4 SETSPRITE = *</p> <p>ODE4 A0 7F LDA #127</p> <p>ODE6 B9 B1 0E SLOOP1 LDA SPRDAT,Y</p> <p>ODE9 99 40 03 STA 832,Y</p> <p>ODEC 88 DEY</p> <p>ODED 10 F7 BPL SLOOP1</p> <p>ODEF A9 0E LDA #14</p> <p>ODF1 8D F8 07 STA 2040</p> <p>ODF4 8D F9 07 STA 2041</p> <p>ODF7 A9 0D LDA #13</p> <p>ODF9 A0 02 LDY #2</p> <p>ODFB 99 FA 07 SLOOP2 STA 2042,Y</p> <p>ODFE 88 DEY</p> <p>ODFF 10 FA BPL SLOOP2</p> <p>OE01 A9 1F LDA #31</p> <p>OE03 8D 15 D0 STA 53269</p> <p>OE06 A9 03 LDA #3</p> <p>OE08 8D 1D D0 STA 53277</p> <p>OE0B A2 02 LDX #2</p> <p>OE0D A9 32 LDA #50</p> <p>OE0F 9D C9 02 STA DELY+2,X</p> <p>OE12 CA DEX</p> <p>OE13 10 F8 BPL LP</p>
---	--	---

Chapter 3

```

OE15 60          RTS
: set paddle position
OE16          SETPOS      = *
OE16 A0 04      LDY #4
OE18 B9 31 0F  SETLOOP   LDA XPOST,Y
OE1B 99 CF 02      STA XPOS,Y
OE1E B9 36 0F      LDA XPOSHT,Y
OE21 99 D7 02      STA XPOSH,Y
OE24 B9 3B 0F      LDA YPOST,Y
OE27 99 DF 02      STA YPOS,Y
OE2A B9 34 0E      LDA SPRCOLOR,Y
OE2D 99 27 D0      STA 53287,Y
OE30 88          DEY
OE31 10 E5        BPL SETLOOP
OE33 60          RTS
OE34 01 00 03  SPRCOLOR .BYTE 1,0,3,3,3
: set up screen
OE39          SETSCREEN  = *
OE39 A9 93      LDA #"(CLR)"
OE3B 20 D2 FF    JSR $FFD2
OE3E A9 0B      LDA #BACKGROUND
OE40 8D 21 D0    STA 53281
OE43 A9 00      LDA #BORDER
OE45 8D 20 D0    STA 53280
OE48 A0 27      LDY #39
OE4A A9 A0 S1    LDA #160
OE4C 99 00 04    STA 1024,Y
OE4F 99 C0 07    STA 1984,Y
OE52 A9 0F      LDA #WALLCOLOR
OE54 99 00 D8    STA 55296,Y
OE57 99 C0 DB    STA 56256,Y
OE5A 88          DEY
OE5B 10 ED      BPL S1
OE5D A0 C8      LDY #200
OE5F A9 A0 S2    LDA #160
OE61 99 00 04    STA 1024,Y
OE64 99 27 04    STA 1063,Y
OE67 99 F8 06    STA 1784,Y
OE6A 99 1F 07    STA 1823,Y
OE6D A9 0F      LDA #WALLCOLOR
OE6F 99 00 D8    STA 1024+54272,Y
OE72 99 27 D8    STA 1063+54272,Y
OE75 99 F8 DA    STA 1784+54272,Y
OE78 99 1F DB    STA 1823+54272,Y
OE7B 98          TYA
OE7C 38          SEC
OE7D E9 28      SBC #40
OE7F A8          TAY
OE80 C9 D8      CMP #216
OE82 D0 DB      BNE S2
OE84 A2 04      LDX #4
OE86 20 9C 0C DRB JSR DRAWBAR
OE89 CA          DEX
OE8A 10 FA      BPL DRB
OE8C A2 00      LDX #0
OE8E A0 01      LDY #1
OE90 18          CLC
OE91 20 FO FF    JSR $FFF0
OE94 A9 51      LDA #<TITLE1
OE96 A0 0F      LDY #>TITLE1
OE98 20 1E AB    JSR $AB1E
OE9B A2 18      LDX #24
OE9D A0 0F      LDY #15
OE9F 18          CLC
OEAO 20 FO FF    JSR $FFF0
OEAA A9 74      LDA #<TITLE2
OEAD A0 0F      LDY #>TITLE2
OEAE 20 1E AB    JSR $AB1E
: negate a number
OEAB          NEGATE    = *
OEAB 49 FF      EOR #255
OEAD 18          CLC
OEAE 69 01      ADC #1
OEBO 60          RTS
: sprite data
OEB1 00 00 00  SPRDAT .BYTE 0,0,0,0,0,0,0
OEB9 00 00 00      .BYTE 0,0,0,0,0,0,0
OEC1 00 00 00      .BYTE 0,0,0,0,0,0,0
OEC9 00 00 00      .BYTE 0,0,0,0,0,0,0,0,0,0
OED1 00 00 3C      .BYTE 0,0,0,0,0,0,0,0,0,0
OED9 00 00 00      .BYTE 0,0,0,0,0,0,0,0
OEE1 00 00 00      .BYTE 0,0,0,0,0,0,0,0
OEE9 00 00 00      .BYTE 0,0,0,0,0,0,0,0,127
OEF1 C0 00 03      .BYTE 192,0,3,192,0,3,192,
0
OEF9 03 C0 00      .BYTE 3,192,0,3,192,0,3,1
92
OF01 00 03 C0      .BYTE 0,3,192,0,3,192,0,3
OF09 C0 00 03      .BYTE 192,0,3,192,0,3,192,
0
OF11 03 C0 00      .BYTE 3,192,0,3,192,0,3,1
92
OF19 00 03 C0      .BYTE 0,3,192,0,3,192,0,3
OF21 C0 00 03      .BYTE 192,0,3,192,0,3,192,
0
OF29 03 C0 00      .BYTE 3,192,0,3,192,0,3,8
OF31 21 1F 0C XPOST .BYTE 33,31,12,12,12
OF36 00 01 00 XPOST .BYTE 0,1,0,0,0
OF3B 82 82 82 YPOST .BYTE 130,130,130,140,1
50
OF40 0A 03 0D COLORS .BYTE 10,3,13,3,10
OF45 A0 E1 61 BRICKS .BYTE 160,225,97
OF48 E1 61 20 RESULTBRI .BYTE 225,97,32,32,32,32
OF4E 0A 14 1E SCORES .BYTE 10,20,30
OF51 9B 12 53 TITLE1 .ASC "[<->](RVS) SCORE
{9 SPACE} TIME 3:
00{4 SPACE} SCOR
E"
OF73 00          .BYTE 0
OF74 50 52 49 TITLE2 .ASC "PRISONBALL"
OF7E 00          .BYTE 0
OF7F 50 52 45 MESS3 .ASC "PRESS FIREBUTT
ON"
OF8F 00          .BYTE 0
OF90          SAVL      = *+6
OF90          COUNTERS = *+26
OF90          SCORE1L   = *+36
OF90          SCORE2L   = *+37
OF90          SCORE1H   = *+38
OF90          SCORE2H   = *+39
OF90          SCORED    = *+40
OF90          INDEX     = *+41
OF90          FLAG      = *+42
OF90          STACKPT   = *+43
OF90          STACK     = *+45

```

Q-Bird

Source code by Mike Sedore

```

0801          * = $801
;
;variables
;
0801          V          = $D000
0801          BDRCOLOR  = V+$20
0801          BCKCOLOR  = V+$21
0801          PRINT     = $FFD2
0801          PCURSOR   = $E510
0801          SPRITEMC  = V+$1C
0801          SPRITEON  = V+$15
0801          SPRHRZ    = V+$1D
0801          SPRVRT    = V+$17
0801          SMC1      = V+$25
0801          SMC2      = V+$26
0801          SCOLOR    = V+$27
0801          SPOINTER  = $7F8
0801          ILZRSPD   = $33C
0801          SID       = $D400
0801          RANDOM    = SID+$1B
0801          JOYSTICK  = $DC00
0801          SPRJMP    = $3FE
0801          SQRCOMP   = $3D0
0801          LEVEL     = $3E1
0801          PRIORITY  = V+$1B
0801          EGGPAUS  = $3C5
0801          IRQ       = $314
0801          IRQENTRY  = $EA31
0801          RESTORE   = $318
0801          SHIFT     = $28D
0801          DIRBJ     = $800D
0801          DIRLJ     = $8001
0801          DIRCJ     = $8002
;
;basic line
0801 08 08 00          .BYT $B,$8,$0,$0,$9E,$3
                        2,$30,$36,$31,$0,$
                        0,$0
;
;.i.
080D A9 00  START      LDA #0
080F 8D 20  DO         STA BDRCOLOR
0812 20 AD 14          JSR SETLIVES
0815 20 1B 08          JSR GRID
0818 4C C1 08          JMP SPRSETUP
081B A9 93  GRID       LDA #$93
081D 20 D2 FF          JSR PRINT
0820 A9 08             LDA #8
0822 20 D2 FF          JSR PRINT
0825 A9 8E             LDA #$8E
0827 20 D2 FF          JSR PRINT
;
;.print grid
082A A9 04             LDA #4
082C 85 02             STA $2
082E A9 0E  ROW       LDA #E
0830 38                SEC
0831 E5 02             SBC $2
0833 85 D3             STA $D3
0835 A5 02             LDA $2
0837 85 D6             STA $D6
0839 20 10 E5          JSR PCURSOR
083C A0 06             LDY #6
083E A2 0E             LDA #E
0840 BD 61 08  ASQUARE LDA SQRDATA,X
0843 20 D2 FF          JSR PRINT
0846 CA                DEX
0847 10 F7             BPL ASQUARE
0849 88                DEY
084A 10 F2             BPL COLUMN
084C A9 0D             LDA #D
084E 20 D2 FF          JSR PRINT
0851 20 D2 FF          JSR PRINT
0854 E6 02             INC $2
0856 E6 02             INC $2
0858 A5 02             LDA $2
085A C9 10             CMP #10
085C D0 D0             BNE ROW
085E 4C 70 08          JMP TEXT
0861 1D 91 B7  SQRDATA .ASC "{(RIGHT){UP}<
2 Y>|<5>{(3 LEFT
){DOWN}E{OFF
}{RVS){WHT}"
;
;.print screen text
0870 A0 09  TEXT       LDY #9
0872 A9 00             LDA #0
0874 85 02             STA $2
0876 A6 02             LDY #6
0878 BD AD 08          LDA TEXTDATA,X
087B C9 FF             CMP #FF
087D F0 08             BEQ NEXTMSSG
087F 20 D2 FF          JSR PRINT
0882 E6 02             INC $2
0884 4C 76 08          JMP NEXTCHAR
0887 88                NEXTMSSG
0888 10 03             BPL READCRSP
088A 4C 8F 13          JMP PRINTLVS
088D E8                INX
088E E6 02             INC $2
0890 BD AD 08          LDA TEXTDATA,X
0893 85 D3             STA $D3
0895 E8                INX
0896 E6 02             INC $2
0898 BD AD 08          LDA TEXTDATA,X
089B 85 D6             STA $D6
089D 48                PHA
089E 8A                TXA
089F 48                PHA
08A0 98                TYA
08A1 48                PHA
08A2 20 10 E5          JSR PCURSOR
08A5 68                PLA
08A6 A8                TAY
08A7 68                PLA
08A8 AA                TAX
08A9 68                PLA
08AA 4C 82 08          JMP CRSRSET
;
;.data for screen text

```

Chapter 3

```

08AD FF 21 05 TEXTDATA .BYT $FF,$21,$5
08B0 1C 51 2D .ASC "(RED)Q-BIRD"
08B7 FF 21 06 .BYT $FF,$21,$6
08BA A3 A3 A3 .ASC "[<6 T>]"
08C0 FF 23 08 .BYT $FF,$23,$8
08C3 1F 42 59 .ASC "(BLU);BY"
08C6 FF 22 0A .BYT $FF,$22,$A
08C9 4D 49 4B .ASC "MIKE"
08CD FF 21 0C .BYT $FF,$21,$C
08DD 53 45 44 .ASC "SEDORE"
08DE FF 04 11 .BYT $FF,$4,$11
08D9 1E 48 49 .ASC "(GRN)HISCORE:<6
        6->00000"
08E8 =
08EB FF 05 13 .BYT $FF,$5,$13
08EB 1E 53 43 .ASC "(GRN)SCORE:<6
        >00000"
08F8 =
08F8 FF 19 11 .BYT $FF,$19,$11
08FB 1E 4C 45 .ASC "(GRN)LEVEL:<6
        >01"
0905 FF 19 13 .BYT $FF,$19,$13
0908 1E 4C 49 .ASC "(GRN)LIVES:<6
        >1"
0910 FF .BYT $FF

;print dividing lines on screen
0911 A9 97 HRZLINES LDA #97
0913 20 D2 FF JSR PRINT
0916 A0 02 LDY #2
0918 A9 A4 LDA #9A4
091A 20 2E 09 JSR PRINTHRZ
091D A0 10 LDY #10
091F A9 A3 LDA #9A3
0921 20 2E 09 JSR PRINTHRZ
0924 A0 14 LDY #14
0926 A9 A4 LDA #9A4
0928 20 2E 09 JSR PRINTHRZ
092B 4C 42 09 JMP VRTLINE
092E 84 D6 PRINTHRZ STY $D6
0930 A0 00 LDY #0
0932 84 D3 STY $D3
0934 48 PHA
0935 20 10 E5 JSR PCURSOR
0938 68 PLA
0939 A2 26 LDX #926
093B 20 D2 FF PRHZ JSR PRINT
093E CA DEX
093F 10 FA BPL PRHZ
0941 60 RTS
0942 A9 1F VRTLINE LDA #91F
0944 85 D3 STA $D3
0946 A9 03 LDA #3
0948 85 D6 STA $D6
094A 20 10 E5 JSR PCURSOR
094D A2 0C LDX #9C
094F A9 A7 VRTLOOP LDA #9A7
0951 20 D2 FF JSR PRINT
0954 A9 9D LDA #99D
0956 20 D2 FF JSR PRINT
0959 A9 11 LDA #911
095B 20 D2 FF JSR PRINT
095E CA DEX
095F 10 EE BPL VRTLOOP
0961 60 RTS

;sprite set-up
0962 A9 FF SPRSET1 LDA #9FF
0964 8D 1C D0 STA SPRITEMC
0967 A9 00
0969 8D 15 D0
096C 8D 1D D0
096F 8D 17 D0
0972 A9 07
0974 8D 25 D0
0977 4C 7A 0B
;
;sprite data
097A 03 C0 00 SPRDAT1 .BYT 3,192,0,14,176,240,
58,175
0982 AC 36 6A .BYT 172,54,106,172,214,
170,176,58
098A AA 80 0E .BYT 170,176,14,170,192,
3,103,0
0992 03 77 00 .BYT 3,119,0,13,221,192,
3,51
099A 00 00 00 .BYT 0,0,0,0,0,0,0,0
09A2 00 00 00 .BYT 0,0,0,0,0,0,0,0
09AA 00 00 00 .BYT 0,0,0,0,0,0,0,0
09B2 00 00 00 .BYT 0,0,0,0,0,0,0,0
09BA 03 C3 00 SRPDAT2 .BYT 3,195,0,14,190,240,
58,170
09C2 EC D6 6B .BYT 236,214,107,172,54,
171,176,218
09CA AE 80 3E .BYT 174,176,62,170,192,
3,103,0
09D2 03 77 00 .BYT 3,119,0,3,119,0,13,2
21
09DA C0 03 33 .BYT 192,3,51,0,0,0,0,0
09E2 00 00 00 .BYT 0,0,0,0,0,0,0,0
09EA 00 00 00 .BYT 0,0,0,0,0,0,0,0
09F2 00 00 00 .BYT 0,0,0,0,0,0,0,0
09FA 00 0F 00 SPRDAT3 .BYT 0,15,0,60,58,192,23
5,234
0A02 B0 EA A6 .BYT 176,234,166,112,58,
170,92,58
0A0A AA 80 0E .BYT 170,176,14,170,192,
3,103,0
0A12 03 77 00 .BYT 3,119,0,13,221,192,
3,51
0A1A 00 00 00 .BYT 0,0,0,0,0,0,0,0
0A22 00 00 00 .BYT 0,0,0,0,0,0,0,0
0A2A 00 00 00 .BYT 0,0,0,0,0,0,0,0
0A32 00 00 00 .BYT 0,0,0,0,0,0,0,0
0A3A 03 0F 00 SPRDAT4 .BYT 3,15,0,62,250,192,2
38,170
0A42 B0 EB A6 .BYT 176,235,166,92,59,1
70,112,58
0A4A EA 9C 0E .BYT 234,156,14,170,240,
3,103,0
0A52 03 77 00 .BYT 3,119,0,3,119,0,13,2
21
0A5A C0 03 33 .BYT 192,3,51,0,0,0,0,0
0A62 00 00 00 .BYT 0,0,0,0,0,0,0,0
0A6A 00 00 00 .BYT 0,0,0,0,0,0,0,0
0A72 00 00 00 .BYT 0,0,0,0,0,0,0,0
0A7A 03 C0 00 SPRDAT5 .BYT 3,192,0,14,176,0,54,
108
0A82 00 EA AC .BYT 0,234,172,0,254,172
,0,234
0A8A AC 00 3E .BYT 172,0,62,176,0,234,
172,0
0A92 35 AC 00 .BYT 53,172,0,13,172,0,1
3,172
0A9A 00 0D AC .BYT 0,13,172,0,58,172,0,
234
0AA2 B0 00 3E .BYT 176,0,62,176,0,14,1
76,0
0AAA 03 AC 00 .BYT 3,172,0,0,240,0,0,0
0AB2 00 00 00 .BYT 0,0,0,0,0,0,0,0

```


The Source Code

```

OABA 03 00 00 SPRDAT6 .BYT 3,192,0,14,176,0,54,108 E,$0,$0
OAC2 00 EA AC .BYT 0,234,172,0,234,172,0,58 ;execute sprite set-up and jump into demo
OACA 00 00 OE .BYT 176,0,14,172,0,58,172,0 ;
OAD2 E5 AC 00 .BYT 229,172,0,61,172,0,13,172 ;
OADA 00 0D AC .BYT 0,13,172,0,58,176,0,234 ;
OAE2 00 00 EE .BYT 176,0,238,176,0,62,176,0 ;
OAEA 03 00 00 .BYT 3,176,0,0,192,0,0,0 ;
OAF2 00 00 00 .BYT 0,0,0,0,0,0,0,0 ;
OAF6 0F 00 00 SPRDAT7 .BYT 15,0,0,58,192,0,230,112 ;
OB02 00 EA AC .BYT 0,234,172,0,234,252,0,234 ;
OB0A AC 00 3A .BYT 172,0,58,240,0,234,172,0 ;
OB12 E9 70 00 .BYT 233,112,0,233,192,0,233,192 ;
OB1A 00 E9 C0 .BYT 0,233,192,0,234,176,0,58 ;
OB22 AC 00 3A .BYT 172,0,58,240,0,58,192,0 ;
OB2A EB 00 00 .BYT 235,0,0,60,0,0,0,0 ;
OB32 00 00 00 .BYT 0,0,0,0,0,0,0,0 ;
OB3A 0F 00 00 SPRDAT8 .BYT 15,0,0,58,192,0,230,112 ;
OB42 00 EA AC .BYT 0,234,172,0,234,172,0,58 ;
OB4A 00 00 EA .BYT 176,0,234,192,0,234,176,0 ;
OB52 E9 6C 00 .BYT 233,108,0,233,240,0,233,192 ;
OB5A 00 E9 C0 .BYT 0,233,192,0,58,176,0,58 ;
OB62 AC 00 3A .BYT 172,0,58,236,0,58,240,0 ;
OB6A 3B 00 00 .BYT 59,0,0,12,0,0,0,0 ;
OB72 00 00 00 .BYT 0,0,0,0,0,0,0,0 ;
;
;set sprite colors and pointers
;
OB7A A9 00 SPRSET2 LDA #0
OB7C 8D 26 D0 STA SMC2
OB7F A2 07 LDX #7
OB81 BD 93 0B SCPLOOP LDA SCDATA,X
OB84 9D 27 D0 STA SCOLOR,X
OB87 BD 9B 0B LDA SPDATA,X
OB8A 9D F8 07 STA SPOINTER,X
OB8D CA DEX
OB8E 10 F1 BPL SCPLOOP
OB90 4C A3 0B JMP SPRSET3
OB93 05 04 04 SCDATA .BYT 5,4,4,9,2,6,14
OB9B 8C 86 84 SPDATA .BYT $8C,$86,$84,$86,$8E
A,$88,$80,$8E
;
;set sprite locations
;
OBA3 A2 0F SPRSET3 LDX #$F
OBA5 BD B1 0B SLOOP LDA SLDATA,X
OBA8 9D 00 D0 STA V,X
OBA9 CA DEX
OBAC 10 F7 BPL SLOOP
OBAE 60 RTS
OBAF 00 BRK
OB80 00 BRK
OB81 78 66 6D SLDATA .BYT $78,$66,$6D,$47,$F
D,$47,$1D,$97
$0,$0,$0,$0,$A9,$9
OB89 00 00 00 .BYT
;
OC36 A9 00 LDA #0
OC38 A2 CB LDX #0
OC3A 9D 34 03 CTB1LOOP STA $334,X
OC3D CA DEX
OC3E E0 FF CPX #$FF
OC40 D0 F8 BNE CTB1LOOP
OC42 60 RTS
OC43 E8 36 03 BJCNTN INC $336
OC46 AE 36 03 NOINC LDX $336
OC49 E0 1A BRK CPX #$1A
OC4B D0 08 BNE MOVEBIRD
OC4D A9 00 LDA #0
OC4F 8D 36 03 STA $336
OC52 4C 15 16 JMP CHECKEGG
;
OCBC 20 62 09 SPRSETUP JSR SPRSET1
OCB4 4C 80 12 JMP DEMOMSSG
;
;start of bird routine
;
OCB7 4C E9 18 BIRD JMP PAUSGAME
OCBA AD 34 03 RTNBIRD LDA $334
OCBC C9 07 OBCD CMP #7
OCBF D0 0F OBCF BNE EXITBIRD
OBD1 A9 00 LDA #0
OBD3 8D 34 03 OBD3 STA $334
OBD6 EE 35 03 OBD6 INC $335
OBD9 AD 35 03 OBD9 LDA $335
OBCD C9 05 OBCD CMP #5
OBDE F0 03 OBDE BIRD1
OBE0 4C D8 15 EXITBIRD JMP CHKEGGP
OBE3 20 2A 0C BIRD1 JSR CBC
OBE6 F0 03 OBE6 BEQ BSTILL
OBE8 4C A9 11 OBE8 JMP SUPERJMP
OBEB 20 C0 11 BSTILL JSR CHKBTTN
OBEE 29 0F OBEE AND #$F
OBFO 49 0F OBFO EOR #$F
OBF2 85 03 OBF2 STA $3
OBF4 29 01 OBF4 AND #1
OBF6 F0 08 OBF6 BEQ JDOWN
OBF8 A9 00 OBF8 LDA #0
OBFA 8D 00 80 OBFA STA DIRBJ
OBFD 4C 8D 0C OBFD JMP SETBIRDP
OC00 A5 03 JDOWN LDA $3
OC02 29 02 OC02 AND #2
OC04 F0 08 OC04 BEQ JLEFT
OC06 A9 20 OC06 LDA #$20
OC08 8D 00 80 OC08 STA DIRBJ
OC0B 4C 8D 0C OC0B JMP SETBIRDP
OC0E A5 03 JLEFT LDA $3
OC10 29 04 OC10 AND #4
OC12 F0 08 OC12 BEQ JRIGHT
OC14 A9 40 OC14 LDA #$40
OC16 8D 00 80 OC16 STA DIRBJ
OC19 4C 8D 0C OC19 JMP SETBIRDP
OC1C A5 03 JRIGHT LDA $3
OC1E 29 08 OC1E AND #8
OC20 F0 BE OC20 BEQ EXITBIRD
OC22 A9 60 OC22 LDA #$60
OC24 8D 00 80 OC24 STA DIRBJ
OC27 4C 8D 0C OC27 JMP SETBIRDP
OC2A A9 00 OC2A LDA #0
OC2C 8D 35 03 OC2C STA $335
OC2F AD 36 03 OC2F LDA $336
OC32 60 OC32 RTS
OC33 8D 21 D0 CTB1 STA BCKCOLOR
;
;clear tape buffer for number storage
;
OC36 A9 00 LDA #0
OC38 A2 CB LDX #0
OC3A 9D 34 03 CTB1LOOP STA $334,X
OC3D CA DEX
OC3E E0 FF CPX #$FF
OC40 D0 F8 BNE CTB1LOOP
OC42 60 RTS
OC43 E8 36 03 BJCNTN INC $336
OC46 AE 36 03 NOINC LDX $336
OC49 E0 1A BRK CPX #$1A
OC4B D0 08 BNE MOVEBIRD
OC4D A9 00 LDA #0
OC4F 8D 36 03 STA $336
OC52 4C 15 16 JMP CHECKEGG
;

```

Chapter 3

```

;move bird 1 pixel of jump
OC55 8A MOVEBIRD TXA
OC56 18 CLC
OC57 6D 00 80 ADC DIRBJ
OC5A AA TAX
OC5B BD F3 0C LDA JMPDATA,X
OC5E AA TAX
OC5F BD 76 0C LDA XDIR,X
OC62 18 CLC
OC63 6D 0C D0 ADC V+$C
OC66 8D 0C D0 STA V+$C
OC69 BD 81 0C LDA YDIR,X
OC6C 18 CLC
OC6D 6D 0D D0 ADC V+$D
OC70 8D 0D D0 STA V+$D
OC73 4C CF 17 JMP BJSOUND
OC76 00 00 00 XDIR .BYT 0,0,0,0,255,255,255,
0,1,1,1
OC81 00 FF 01 YDIR .BYT 0,255,1,0,0,255,1,0,
0,255,1,0

;set bird sprite pointer to match direction of jump
OC8D AD 00 80 SETBIRDP LDA DIRBJ
OC90 FO 0C BEQ BIRDUR
OC92 C9 60 CMP #$60
OC94 FO 08 BEQ BIRDUR
OC96 A9 81 LDA #$81
OC98 8D FE 07 STA SPOINTER+6
OC9B 4C 43 0C JMP BJCNTR
OC9E A9 83 BIRDUR LDA #$83
OCA0 8D FE 07 STA SPOINTER+6
OCA3 4C 43 0C JMP BJCNTR
OCA6 CE FE 07 BIRDSIT DEC SPOINTER+6
OCA9 4C E0 0B JMP EXITBIRD

;translate sprite locations to a screen location
OCAC A9 04 SLTOSL LDA #4
OCAE 85 FE STA $FE
OCB0 A0 00 LDY #0
OCB2 84 FD STY $FD
OCB4 A2 04 LDX #4
OCB6 AD 3A 03 VRTADD LDA $33A
OCB9 18 CLC
OCBA 65 FD ADC $FD
OCBC 85 FD STA $FD
OCBE A5 FE LDA $FE
OCC0 69 00 ADC #0
OCC2 85 FE STA $FE
OCC4 CA DEX
OCC5 10 EF BPL VRTADD
OCC7 A5 FD LDA $FD
OCC9 18 CLC
OCCA 6D 3B 03 ADC $33B
OCCD 85 FD STA $FD
OCCF A5 FE LDA $FE
OCD1 69 00 ADC #0
OCD3 85 FE STA $FE
OCD5 60 RTS
OCD6 00 BRK
OCD7 00 BRK
OCD8 AD 0D 00 SLSU LDA V+$D
OCD8 = LV1 = *-1
OCD8 38 SEC
OCD8 E9 31 SBC #$31
OCDE 8D 3A 03 STA $33A
OCE1 AD 0C D0 LDA V+$C
OCE4 = LV2 = *-1
OCE4 4A LSR

OCE5 4A LSR
OCE6 4A LSR
OCE7 18 CLC
OCE8 69 0C ADC #$C
OCEA 8D 3B 03 STA $33B
OCED 20 AC 0C JSR SLTOSL
OCF0 4C 8C 0D JTR4 JMP CHKSQR

;data for direction of each dot of movement during a jump
OCF3 JMPDATA = *
OCF3 00 01 01 JDUP .BYT 0,1,1,1,1,1,1,1
OCFB 01 09 09 .BYT 1,9,9,9,9,9,9,9
OD03 09 08 08 .BYT 9,8,8,8,8,8,8,8
OD0B 08 00 00 .BYT 8,0,0,0,0,0,0,0
OD13 00 04 04 JDDOWN .BYT 0,4,4,4,4,4,4,4
OD1B 04 06 06 .BYT 4,6,6,6,6,6,6,6
OD23 06 02 02 .BYT 6,2,2,2,2,2,2,2
OD2B 02 00 00 .BYT 2,0,0,0,0,0,0,0
OD33 00 05 05 JDLEFT .BYT 0,5,5,5,5,5,5,5
OD3B 05 04 04 .BYT 5,4,4,4,4,4,4,4
OD43 04 06 06 .BYT 4,6,6,6,6,6,6,6
OD4B 06 00 00 .BYT 6,0,0,0,0,0,0,0
OD53 00 09 09 JDRIGHT .BYT 0,9,9,9,9,9,9,9
OD5B 09 08 08 .BYT 9,8,8,8,8,8,8,8
OD63 08 0A 0A .BYT 8,10,10,10,10,10,10,10
OD6B 0A 00 00 .BYT 10,0,0,0,0,0,0,0

;change square color
OD73 A5 FE CSQRCOL LDA $FE
OD75 18 CLC
OD76 69 D4 ADC #$D4
OD78 85 FE STA $FE
OD7A A0 02 LDA #2
OD7C B1 FD CLOOP LDA ($FD),Y
OD7E 4C 48 12 JMP CHKCOLR
OD81 20 01 17 CCC JSR INCSCORE
OD84 91 FD STA ($FD),Y
OD86 88 DEY
OD87 10 F3 BPL CLOOP
OD89 4C A6 0C CCDONE JMP BIRDSIT

;check if bird is on a square
OD8C B1 FD CHKSQR LDA ($FD),Y
OD8E C9 E9 CMP #$E9
OD90 F0 E1 BEQ CSQRCOL
OD92 4C 46 14 JMP DEATH

;start of lizard routine
OD95 EE 37 03 LIZARD INC $337
OD98 AD 37 03 LDA $337
OD9B C9 02 CMP #2
OD9D D0 2C BNE EXITLZRD
OD9F A9 00 LDA #0
ODA1 8D 37 03 STA $337
ODA4 EE 38 03 INC $338
ODA7 AD 38 03 LDA $338
ODAA C9 01 CMP #1
ODAC D0 1D BNE EXITLZRD
ODAE A9 00 LDA #0
ODB0 8D 38 03 STA $338
ODB3 A2 01 LDX #1
ODB5 FE 3C 03 LZRDNCNTR INC ILZRSPD,X
ODB8 BD 3C 03 LDA ILZRSPD,X
ODBB DD CD 0D CMP ILSDATA-1,X
ODBE D0 08 BNE NEXTLZRD
ODC0 A9 00 LDA #0
ODC2 9D 3C 03 STA ILZRSPD,X

```

The Source Code

```

ODC5 20 EA 0D      JSR  MOVELZRD
ODC8 4C BC 0E NEXTLZRD
ODCB 4C 15 11 EXITLZRD
ODCE 18 1C 20 ILSDATA .BYT $18,$1C,$20

```

;sound set-up

```

ODD1 A2 18 SOUNDSU LDX  #$18
ODD3 A9 00          LDA  #0
ODD5 9D 00 D4 CLEARSID STA SID,X
ODD8 CA            DEX
ODD9 10 FA          BPL  CLEARSID
ODDB A9 FF          LDA  #$FF
ODDD 8D 0F D4      STA  SID+15
ODE0 A9 80          LDA  #$80
ODE2 8D 12 D4      STA  SID+$12
ODE5 A9 00          LDA  #0
ODE7 4C 70 17      JMP  RNDMEM

```

;start of routine to handle 1 lizard

```

ODEA BD 40 03 MOVELZRD LDA  $340,X
ODED D0 0B          BNE  CLJUMP
ODEF 20 5E 0E      JSR  CLZRDJD
ODF2 0A            ASL
ODF3 0A            ASL
ODF4 0A            ASL
ODF5 0A            ASL
ODF6 0A            ASL
ODF7 20 47 0E      JSR  SETLZRPD
ODFA BD 50 03 CLJUMP LDA  $350,X
ODFD 8D 01 80      STA  DIRLJ
OE00 FE 40 03      INC  $340,X
OE03 BC 40 03      LDY  $340,X
OE06 C0 1E          CPY  #$1E
OE08 D0 06          BNE  LJND
OE0A A9 00          LDA  #0
OE0C 9D 40 03      STA  $340,X
OE0F 60            RTS
OE10 86 05 LJND    STX  $5
OE12 06 05          ASL  $5
OE14 8E 4F 03      STX  $34F

```

;move lizard 1 pixel during jump

```

OE17 98            TYA
OE18 18            CLC
OE19 6D 01 80      ADC  DIRLJ
OE1C A8            TAY
OE1D B9 F3 0C      LDA  JMPDATA,Y
OE20 D0 0A          BNE  LSM
OE22 48            PHA
OE23 BD F8 07      LDA  SPOINTER,X
OE26 29 FE          AND  #$FE
OE28 9D F8 07      STA  SPOINTER,X
OE2B 68            PLA
OE2C AA            TAX
OE2D A4 05          LDY  $5
OE2F BD 76 0C      LDA  XDIR,X
OE32 18            CLC
OE33 79 00 D0      STX  V,Y
OE36 99 00 D0      STA  V,Y
OE39 BD 81 0C      LDA  YDIR,X
OE3C 18            CLC
OE3D 79 01 D0      ADC  V+1,Y
OE40 99 01 D0      STA  V+1,Y
OE43 AE 4F 03      LDX  $34F
OE46 60            RTS

```

;set lizard sprite pointer to match direction of jump

```

OE47 9D 50 03 SETLZRPD STA  $350,X

```

```

OE4A C9 0A          CMP  #0
OE4C F0 0A          BEQ  LZRDUR
OE4E C9 60          CMP  #$60
OE50 F0 06          BEQ  LZRDUR
OE52 A9 85          LDA  #$85
OE54 9D F8 07      STA  SPOINTER,X
OE57 60            RTS
OE58 A9 87 LZRDUR  LDA  #$87
OE5A 9D F8 07      STA  SPOINTER,X
OE5D 60            RTS
OE5E 20 B3 0E CLZRDJD JSR  SETY

```

;take the determined locations for where the lizard
will end after the chosen direction for a jump, and
translate them into a screen location

```

OE61 AD 3A 03          LDA  $33A
OE64 38            SEC
OE65 E9 2C          SBC  #$2C
OE67 8D 3A 03      STA  $33A
OE6A AD 3B 03      LDA  $33B
OE6D 4A            LSR
OE6E 4A            LSR
OE6F 4A            LSR
OE70 18            CLC
OE71 69 0C          ADC  #$C
OE73 8D 3B 03      STA  $33B
OE76 86 FC          STX  $FC
OE78 20 AC 0C      JSR  SLTOSL
OE7B A6 FC          LDY  $FC

```

;check if chosen direction for lizard jump
will end it on a square

```

OE7D A0 00          LDY  #0
OE7F B1 FD          LDA  ($FD),Y
OE81 C9 E9          CMP  #$E9
OE83 D0 D9          BNE  CLZRDJD
OE85 AD FF 03      LDA  $3FF
OE88 60            RTS
OE89 86 05 RNDLJUMP STX  $5
OE8B 20 97 17      JSR  GETRND
OE8E 29 03          AND  #3
OE90 8D FF 03      STA  $3FF

```

;determine location where lizard will end after jump

```

OE93 AA            TAX
OE94 BD AB 0E      LDA  ADDX,X
OE97 18            CLC
OE98 79 01 D0      ADC  V+1,Y
OE9B 8D 3A 03      STA  $33A
OE9E BD AF 0E      LDA  ADDY,X
OEA1 18            CLC
OEA2 79 00 D0      ADC  V,Y
OEA5 8D 3B 03      STA  $33B
OEAB A6 05          LDY  $5
OEAA 60            RTS
OEAB F2 12 02 ADDX .BYT $F2,$12,$2,$2
OEAF 12 F2 EA ADDY .BYT $12,$F2,$EA,$1A
OEB3 86 05 SETY   STX  $5
OEB5 06 05          ASL  $5
OEB7 A4 05          LDY  $5
OEB9 4C 89 0E      JMP  RNDLJUMP

```

;move on to next lizard

```

OEBE E8            NEXTLZRD INX
OEBD E0 04          CPX  #4
OEBF F0 03          BEQ  LZRDDONE
OEC1 4C B5 0D      JMP  LZRDONTR
OEC4 4C CB 0D LZRDDONE JMP  EXITLZRD

```

Chapter 3

;transfer sprite data to new locations				00FF	03 57 00	.BYT	3,87,0,3,155,0,3,171		
				1007	00 0E 6C	.BYT	0,14,108,0,14,175,1 92,57		
0EC7	A2 00	MOVESPRD	LDX	#0	100F	BE B0 3A	.BYT	190,176,58,186,192, 57,190,176	
0EC9	BD 7A 09	MDLOOP	LDA	SPRDAT1,X					
0ECC	9D 00 20		STA	\$2000,X	1017	0E AA B0	.BYT	14,170,176,14,106,1 92,3,171	
0ECF	BD 7A 0A		LDA	SPRDAT5,X					
0ED2	9D 00 21		STA	\$2100,X	101F	00 00 FC	.BYT	0,0,252,0,0,0,0,0	
0ED5	BD E7 0E		LDA	SPRDAT9,X	1027	03 FF 00	SPRDAT14	.BYT	3,255,0,14,102,192, 58,170
0ED8	9D 00 22		STA	\$2200,X					
0EDB	BD A7 0F		LDA	SPRDAT12,X	102F	80 35 99	.BYT	176,53,153,112,53,8 5,112,13	
0EDE	9D C0 22		STA	\$22C0,X					
0EE1	CA		DEX		1037	55 C0 03	.BYT	85,192,3,87,0,3,155 0	
0EE2	DO E5		BNE	MDLOOP	103F	03 9B 00	.BYT	3,155,0,3,171,0,3,15 5	
0EE4	4C D1 0D		JMP	SOUNDSU					
;sprite data									
0EE7	00 FC 00	SPRDAT9	.BYT	0,252,0,3,171,0,14, 90	1047	00 0E AC	.BYT	0,14,172,0,14,108,0, 14	
0EEF	CO 39 AA		.BYT	192,57,170,176,58,1 70,176,234	104F	AC 00 0E	.BYT	172,0,14,108,0,14,1 72,0	
0EF7	A9 AC EA		.BYT	169,172,234,170,17 2,234,170,172	1057	03 6B 00	.BYT	3,107,0,3,171,0,3,1 55	
0EFF	39 AA B0		.BYT	57,170,176,57,105,1 76,14,86	105F	00 00 EA	.BYT	0,0,234,192,0,63,0,0	
0F07	CO 03 AB		.BYT	192,3,171,0,0,252,0, 0	1067	00 00 00	SPRDAT15	.BYT	0,0,0,0,0,0,0,48
0F0F	00 00 00		.BYT	0,0,0,0,0,0,0,0	106F	00 00 EC	.BYT	0,0,236,0,3,171,0,3	
0F17	00 00 00		.BYT	0,0,0,0,0,0,0,0	1077	6B 00 03	.BYT	107,0,3,107,0,3,91,0	
0F1F	00 00 00		.BYT	0,0,0,0,0,0,0,0	107F	03 9B 00	.BYT	3,155,0,0,252,0,0,0	
0F27	00 FC 00	SPRDAT10	.BYT	0,252,0,3,171,0,14,1 50	1087	00 00 00	.BYT	0,0,0,0,0,0,0,0	
0F2F	CO 3A A9		.BYT	192,58,169,176,58,1 70,176,234	108F	00 00 00	.BYT	0,0,0,0,0,0,0,0	
0F37	AA AC EA		.BYT	170,172,234,170,17 2,234,170,172	109F	00 00 00	.BYT	0,0,0,0,0,0,0,0	
0F3F	39 AA B0		.BYT	57,170,176,57,106,1 76,14,90	10A7	AD 01 D0	COBRADIR	LDA	V+1
0F47	CO 03 AB		.BYT	192,3,171,0,0,252,0, 0	10AA	18		CLC	
0F4F	00 00 00		.BYT	0,0,0,0,0,0,0,0	10AB	69 08		ADC	#8
0F57	00 00 00		.BYT	0,0,0,0,0,0,0,0	10AD	38		SEC	
0F5F	00 00 00		.BYT	0,0,0,0,0,0,0,0	10AE	ED F6 03		SBC	\$3F6
0F67	00 00 00	SPRDAT11	.BYT	0,0,0,0,0,0,0,0	10B1	FO 08		BEQ	CHKHORZ
0F6F	00 03 F3		.BYT	0,3,243,252,62,174, 171,239	10B3	90 03		BCC	CDOWN
0F77	FA 65 EA		.BYT	250,101,234,170,16 9,234,234,175	10B5	A9 00		LDA	#0
0F7F	3F AA FO		.BYT	63,170,240,58,175,0 15,240	10B7	60		RTS	
0F87	00 00 00		.BYT	0,0,0,0,0,0,0,0	10B8	A9 20	CDOWN	LDA	#\$20
0F8F	00 00 00		.BYT	0,0,0,0,0,0,0,0	10BA	60		RTS	
0F97	00 00 00		.BYT	0,0,0,0,0,0,0,0	10BB	AD 00 D0	CHKHORZ	LDA	V
0F9F	00 00 00		.BYT	0,0,0,0,0,0,0,0	10BE	18		CLC	
0FA7	00 00 00	SPRDAT12	.BYT	0,0,0,0,0,0,15,192	10BF	69 02		ADC	#2
0FAF	00 3A BC		.BYT	0,58,188,0,14,171,2 52,63	10C1	38		SEC	
0FB7	AE AB EF		.BYT	174,171,239,250,10 1,234,170,169	10C2	ED F5 03		SBC	\$3F5
0FBF	EA EA AF		.BYT	234,234,175,63,170, 240,14,171	10C5	90 03		BCC	CRIGHT
0FC7	00 0E AC		.BYT	0,14,172,0,58,240,0, 15	10C7	A9 40		LDA	#\$40
0FCF	00 00 00		.BYT	0,0,0,0,0,0,0,0	10C9	60		RTS	
0FD7	00 00 00		.BYT	0,0,0,0,0,0,0,0	10CA	A9 60	CRIGHT	LDA	#\$60
0FDf	00 00 00		.BYT	0,0,0,0,0,0,0,0	10CC	60		RTS	
0FE7	00 00 00	SPRDAT13	.BYT	0,0,0,0,252,0,3,171	;see if cobra needs new direction to jump				
0FEF	00 0E 66		.BYT	0,14,102,192,62,170 240,53	10CD	AD F0 03	CBRASTRRT	LDA	\$3F0
0FF7	A9 70 35		.BYT	169,112,53,153,112, 13,85,192	10DD	DO 06		BNE	CONTJCJMP
					10D2	20 36 11		JSR	FINDCDIR
					10D5	8D 02 80		STA	DIRCJ
					10D8	EE F0 03	CONTJCJMP	INC	\$3F0
					10DB	AE F0 03		LDX	\$3F0
					10DE	E0 20		CPX	#\$20
					10E0	F0 2B		BEQ	CJDONE
					10E2	8A		TXA	CLC
					10E3	18		CLC	
					10E4	6D 02 80		ADC	DIRCJ
					10E7	AA		TAX	
					10E8	BD F3 0C		LDA	JMPDATA,X
					10EB	AA		TAX	

The Source Code

```

10EC DO 08          BNE CJND
10EE AD F8 07      LDA SPOINTER
10F1 29 8C        AND #$8C
10F3 8D F8 07     STA SPOINTER

;move cobra 1 pixel of jump

10F6 BD 76 0C CJND LDA XDIR,X
10F9 18           CLC
10FA 6D 00 D0     ADC V
10FD 8D 00 D0     STA V
1100 BD 81 0C     LDA YDIR,X
1103 18           CLC
1104 6D 01 D0     ADC V+1
1107 8D 01 D0     STA V+1
110A 4C 4E 11 EXITCBRA JMP HAWK
110D A9 00        CJDONE LDA #0
110F 8D F0 03    STA $3F0
1112 4C 0A 11     JMP EXITCBRA

;start of cobra routine

1115 EE F1 03 COBRA INC $3F1
1118 AD F1 03     LDA $3F1
111B C9 0A        CMP #$A
111D D0 EB        BNE EXITCBRA
111F A9 00        LDA #0
1121 8D F1 03    STA $3F1
1124 EE F2 03     INC $3F2
1127 AD F2 03     LDA $3F2
112A C9 05        CMP #5
112C D0 DC        BNE EXITCBRA
112E A9 00        LDA #0
1130 8D F2 03    STA $3F2
1133 4C CD 10     JMP CBRASTRT
1136 A9 8D        FINDCDIR LDA #$8D
1138 8D F8 07    STA SPOINTER
113B 4C A7 10     JMP COBRADIR

;store position of bird from last square it was on
;for cobra to jump toward

113E AD 0C D0 MEMBIRDL LDA V+$C
1141 8D F5 03     STA $3F5
1144 AD 0D D0     LDA V+$D
1147 8D F6 03     STA $3F6
114A AD 00 DC     LDA JOYSTICK
114D 60           RTS

;start of hawk routine

114E EE FA 03 HAWK INC $3FA
1151 AD FA 03     LDA $3FA
1154 C9 06        CMP #6
1156 D0 0F        BNE EXITHAWK
1158 A9 00        LDA #0
115A 8D FA 03    STA $3FA
115D EE FB 03     INC $3FB
1160 AD FB 03     LDA $3FB
1163 C9 05        CMP #5
1165 F0 28        BEQ ANMTHAWK
1167 4C D2 11 EXITHAWK JMP BALL
116A A9 00        MOVEHAWK LDA #0
116C 8D FB 03     STA $3FB

;move hawk

116F EE 08 D0     INC V+8
1172 AD 08 D0     LDA V+8
1175 C9 01        CMP #1
1177 D0 EE        BNE EXITHAWK

;give hawk random row to fly across

1179 20 97 17 RNDROW JSR GETRND
117C 29 07        AND #7
117E C9 06        CMP #6
1180 80 F7        BCS RNDROW
1182 0A           ASL
1183 0A           ASL
1184 0A           ASL
1185 0A           ASL
1186 18           CLC
1187 69 4C        ADC #$4C
1189 8D 09 D0     STA V+9
118C 4C 67 11     JMP EXITHAWK
118F EE FC 03 ANMTHAWK INC $3FC
1192 AD FC 03     LDA $3FC
1195 C9 20        CMP #$20
1197 D0 D1        BNE MOVEHAWK

;animate hawk

1199 AD FC 07     LDA SPOINTER+4
119C 49 01        EOR #1
119E 8D FC 07     STA SPOINTER+4
11A1 A9 00        LDA #0
11A3 8D FC 03     STA $3FC
11A6 4C 6A 11     JMP MOVEHAWK

;make each movement of bird twice if super jump is on

11A9 AD FE 03 SUPERJMP LDA SPRJMP
11AC C9 01        CMP #1
11AE F0 03        BEQ SUPERON
11B0 4C 43 0C ICNTR JMP BJCNTR
11B3 EE FD 03 SUPERON INC $3FD
11B6 AD FD 03     LDA $3FD
11B9 29 01        AND #1
11BB F0 F3        BEQ ICNTR
11BD 4C 46 0C     JMP NOINC

;check button for super jump

11C0 A9 00        CHKBTNTN LDA #0
11C2 8D FE 03     STA $3FE
11C5 AD 00 DC     LDA JOYSTICK
11C8 29 10        AND #$10
11CA D0 03        BNE NOTPRSSD
11CC EE FE 03     INC SPRJMP
11CF 4C 3E 11 NOTPRSSD JMP MEMBIRDL

;start of ball routine

11D2 EE F3 03 BALL INC $3F3
11D5 AD F3 03     LDA $3F3
11D8 C9 05        CMP #5
11DA D0 0F        BNE EXITBALL
11DC A9 00        LDA #0
11DE 8D F3 03    STA $3F3
11E1 EE F4 03     INC $3F4
11E4 AD F4 03     LDA $3F4
11E7 C9 0A        CMP #$A
11E9 F0 03        BEQ MOVEBALL
11EB 4C C7 0B EXITBALL JMP BIRD
11EE A9 00        MOVEBALL LDA #0
11F0 8D F4 03    STA $3F4

;move ball down

11F3 EE 0B D0     INC V+8B
11F6 4C 39 12     JMP CHKBOT
11F9 C9 01        CMP #1
11FB F0 21        BEQ RNDCOL

```

Chapter 3

```

11FD C9 48      CMP  #548
11FF 90 1A      BCC  NOLEFT
:
:move ball left
:
1201 CE 0A DO BALLLEFT  DEC  V+5A
1204 EE EF 03      INC  $3EF
1207 AD EF 03      LDA  $3EF
120A C9 08      CMP  #8
120C DO 0D      BNE  NOLEFT
:
:animate ball
:
120E AD FD 07      LDA  SPOINTER+5
1211 49 01      EOR  #1
1213 8D FD 07      STA  SPOINTER+5
1216 A9 00      LDA  #0
1218 8D EF 03      STA  $3EF
121B 4C EB 11 NOLEFT  JMP  EXITBALL
:
:give ball random column to roll down
:
121E 20 97 17 RNDCOL  JSR  GETRND
1221 29 07      AND  #7
1223 C9 07      CMP  #7
1225 B0 F7      BCS  RNDCOL
1227 AA      TAX
1228 A9 00      LDA  #0
122A 18      CLC
122B 69 18      BCOL  ADC  #518
122D CA      DEX
122E 10 FA      BPL  BCOL
1230 18      CLC
1231 69 58      ADC  #558
1233 8D 0A DO      STA  V+5A
1236 4C 01 12     JMP  BALLLEFT
1239 AD 0B DO      LDA  V+5B
123C C9 A8      CMP  #5A8
123E DO B9      BNE  MOVELEFT
1240 A9 01      LDA  #1
1242 8D 0B DO     STA  V+5B
1245 4C 1E 12     JMP  RNDCOL
1248          CLEVEL =
1248          CLEVEL = *+3
1248          CLEVEL = *+14
:
:check if square has changed to final color
:
1248 29 0F      CHKCOLR  AND  #5F
124A C9 00      CMP  #0
124C DO 03      BNE  INCCOLR
124E 4C 89 0D   JMP  CCDONE
1251 85 04      INCCOLR  STA  $4
1253 E6 04      INC  $4
1255 C9 00      CMP  #0
1257 DO 0D      BNE  NFC
:
:check if board is complete
:
1259 EE DO 03      INC  SQRCOMP
125C AD DO 03      LDA  SQRCOMP
125F C9 7E      CMP  #57E
1261 DO 03      BNE  NFC
1263 4C 69 12     JMP  BRDCMPLT
1266 4C 81 0D     NFC  JMP  CCC
1269 A5 04      BRDCMPLT LDA  $4
126B 91 FD      STA  ($FD),Y
126D 20 01 17    JSR  INCSCORE
1270 4C 55 13     JMP  BCPAUSE
1273 AD E1 03     SETCHK LDA  LEVEL
1276 8D 56 12     STA  CLEVEL1
1279 8D 4B 12     STA  CLEVEL
127C 4C 82 13     JMP  ADJCMPL

127F 60      RTS
:
:print demo message
:
1280 A9 06      DEMOMSSG LDA  #6
1282 85 D3      STA  $D3
1284 A9 16      LDA  #516
1286 85 D6      STA  $D6
1288 20 10 E5    JSR  PCURSOR
128B A2 00      LDX  #0
128D 8D 9B 12    DMLoop LDA  DDMSSG,X
1290 20 D2 FF    JSR  PRINT
1293 E8      INX
1294 E0 3A      CPX  #53A
1296 D0 F5      BNE  DMLoop
1298 4C 3B 15    JMP  DEMOSET
129B 08 8E 9E    DDMSSG .BYT 8,$8E,$9E
129E 55 53 45    .ASC "USE STICK TO SE
LECT LEVEL"
12B7 0D 20 20    .BYT 13,32,32,32,32,3
2,32
12BF 50 52 45    .ASC "PRESS TRIGGE
R TO START"
:
:start of demo routine
:
12D5 20 48 15 DEMO  JSR  LZRDDEMO
12D8 29 10      AND  #510
12DA D0 0B      BNE  CHKSTICK
12DC 20 4A 13    JSR  CLEARBOT
12DF A9 40      LDA  #540
12E1 8D 15 D0    STA  SPRITEON
12E4 4C 24 17    JMP  BLANKSCR
12E7 AD 00 DC     CHKSTICK LDA  JOYSTICK
12EA 29 0F      AND  #5F
12EC 49 0F      EOR  #5F
12EE 29 03      AND  #3
12F0 C9 01      CMP  #1
12F2 D0 0D      BNE  LVLDOWN
12F4 AD E1 03    LDA  LEVEL
12F7 C9 0F      CMP  #5F
12F9 F0 DA      BEQ  DEMO
12FB EE E1 03    INC  LEVEL
12FE 4C 0F 13    JMP  PRINTLVL
1301 C9 02      LVLDOWN CMP  #2
1303 D0 D0      D0  D0 BNE  DEMO
1305 AD E1 03    LDA  LEVEL
1308 C9 02      CMP  #2
130A 90 C9      BCC  DEMO
130C CE E1 03    DEC  LEVEL
:
:print level to screen
:
130F AD E1 03    PRINTLVL LDA  LEVEL
1312 C9 0A      CMP  #5A
1314 B0 0E      BCS  GRTR10
1316 18      CLC
1317 69 30      ADC  #530
1319 8D C8 06    STA  $6C8
131C A9 30      LDA  #530
131E 8D C7 06    STA  $6C7
1321 4C D5 12    JTR1  JMP  DEMO
1324 18      CLC
1325 69 26      ADC  #526
1327 8D C8 06    STA  $6C8
132A A9 31      LDA  #531
132C 8D C7 06    STA  $6C7
132F 4C D5 12    JTR2  JMP  DEMO
1332 EE E3 03    DP1  INC  $3E3
1335 D0 FB      BNE  DP1
1337 EE E4 03    DEMOPAU INC  $3E4
133A AD E4 03    LDA  $3E4

```

The Source Code

```

133D C9 FF      CMP    #$FF
133F D0 F6      BNE    DEMOPAUS
1341 A9 08      LDA    #8
1343 8D E4 03   STA    $3E4
1346 AD 00 DC    LDA    JOYSTICK
1349 60          RTS

;clear bottom 2 rows of screen
134A A2 4F      CLEARBOT LDX    #$4F
134C A9 20      LDA    #$20
134E 9D 70 07   CBLOOP  STA    $770,X
1351 CA        DEX
1352 10 FA      BPL    CBLOOP
1354 60          RTS

;short pause after board is complete
1355 EE E3 03   BCPAUSE INC    $3E3
1358 D0 FB      BNE    BCPAUSE
135A EE E4 03   INC    $3E4
135D D0 F6      BNE    BCPAUSE
135F EE E5 03   INC    $3E5
1362 AD E5 03   LDA    $3E5
1365 C9 02      CMP    #2
1367 D0 EC      BNE    BCPAUSE
1369 A9 00      LDA    #0
136B 8D E5 03   STA    $3E5
136E 8D 15 D0   STR1   STA    SPRITEON
1371 4C 9C 13   JMP    LEVELINC
1374 AD E1 03   INCLEVEL LDA    LEVEL
1377 D0 03      BNE    LNINC
1379 EE E1 03   INC    LEVEL
137C 20 73 12   LNINC  JSR    SETCHK
137F 4C A9 13   JMP    PRNEWLVL
1382 EE 4B 12   ADJCMPL INC    CLEVEL
1385 C9 0F      CMP    #8
1387 D0 05      BNE    LEAVEIT
1389 A9 00      LDA    #0
138B 8D 4B 12   STA    CLEVEL
138E 60          LEAVEIT RTS

;print # of lives
138F          LIFE1   =    *+1
138F A9 30      PRINTLVS LDA    #$30
1391 20 D2 FF   JSR    PRINT
1394          LIFE2   =    *+1
1394 A9 35      LDA    #$35
1396 20 D2 FF   JSR    PRINT
1399 4C 11 09   JMP    HRZLINES

;set-up to start a new level
139C 20 1B 08   LEVELINC JSR    GRID
139F AD E1 03   LDA    LEVEL
13A2 C9 0F      CMP    #$F
13A4 F0 03      BEQ    PRNEWLVL
13A6 EE E1 03   INC    LEVEL
13A9 A9 60      PRNEWLVL LDA    #$60
13AB 8D 21 13   STA    JTR1
13AE 8D 2F 13   STA    JTR2
13B1 20 0F 13   JSR    PRINTLVL
13B4 A9 4C      LDA    #$4C
13B6 8D 21 13   STA    JTR1
13B9 8D 2F 13   STA    JTR2
13BC 20 E0 13   JSR    GETREADY
13BF 20 C0 16   JSR    INCLIVES
13C2 A9 FF      LDA    #$FF
13C4 8D 15 D0   STA    SPRITEON
13C7 AD E1 03   LDA    LEVEL
13CA 85 04      STA    $4

13CC A9 00      LDA    #0
13CE A2 CB      LDX    #0CB
13D0 9D 34 03   CTB2   STA    $334,X
13D3 CA        DEX
13D4 E0 FF      CPX    #$FF
13D6 D0 F8      BNE    CTB2
13D8 A5 04      LDA    $4
13DA 8D E1 03   STA    LEVEL
13DD 4C 62 15   JTR3   JMP    SETSPEED

;print "get ready"
13E0 A9 0E      GETREADY LDA    #0E
13E2 85 D3      STA    $D3
13E4 A9 16      LDA    #016
13E6 85 D6      STA    $D6
13E8 20 4A 13   JSR    CLEARBOT
13EB 20 10 E5   JSR    PCURSOR
13EE A2 00      LDX    #0
13F0 BD 0E 14   GRLOOP  LDA    DATREADY,X
13F3 20 D2 FF   JSR    PRINT
13F6 E8        INX
13F7 E0 0A      CPX    #0A
13F9 D0 F5      BNE    GRLOOP
13FB A9 60      LDA    #060
13FD 8D 6E 13   STA    STR1
1400 20 55 13   JSR    BCPAUSE
1403 A9 8D      LDA    #08D
1405 8D 6E 13   STA    STR1
1408 20 BA 14   JSR    CHANGETO
140B 4C 73 12   JMP    SETCHK
140E 9E        DATREADY .BYT $9E
140F 47 45 54   .ASC "GET READY"

;bird fall after death
1418 EE D4 03   BIRDFALL INC    $3D4
141B D0 FB      BNE    BIRDFALL
141D EE D5 03   INC    $3D5
1420 AD D5 03   LDA    $3D5
1423 C9 03      CMP    #3
1425 D0 F1      BNE    BIRDFALL
1427 A9 00      LDA    #0
1429 8D D5 03   STA    $3D5
142C EE D0 D0   INC    V+$D
142F AD D0 D0   LDA    V+$D
1432 49 FF      EOR    #$FF
1434 8D 0F D4   STA    SID+$F
1437 AD D0 D0   LDA    V+$D
143A C9 FF      CMP    #$FF
143C D0 DA      BNE    BIRDFALL
143E A9 00      LDA    #0
1440 8D 1B D0   STA    PRIORITY
1443 4C 68 14   JMP    LOSELIFE
1446 A9 40      LDA    #040
1448 8D 1B D0   STA    PRIORITY
144B 4C 18 14   JMP    BIRDFALL
144E AD D0 03   NEWSTART LDA    $D0COMP
1451 85 FC      STA    $FC
1453 A9 60      LDA    #060
1455 8D DD 13   STA    JTR3
1458 20 A9 13   JSR    PRNEWLVL
145B A9 4C      LDA    #$4C
145D 8D DD 13   STA    JTR3
1460 A5 FC      LDA    $FC
1462 8D D0 03   STA    SORCOMP
1465 4C C7 0B   JSR    BIRD
1468 8D 15 D0   LOSELIFE STA    SPRITEON

;decrease lives and check for end of game
146B AD 18 07   LDA    $718

```

Chapter 3

```

146E C9 30          CMP    # $30
1470 D0 15          BNE   LIVES1
1472 AD 17 07       LDA   $717
1475 C9 30          CMP    # $30
1477 D0 03          BNE   LIVES2
1479 4C FC 14       JMP   GAMEOVER
147C A9 39          LDA   # $39
147E 8D 18 07       STA   $718
1481 CE 17 07       DEC   $717
1484 4C 8A 14       JMP   STRLIVES
1487 CE 18 07       DEC   $718
148A A9 40          LDA   # $40
148C 8D 15 D0       STA   SPRITEON
148F A9 A9          LDA   # $A9
1491 8D 0C D0       STA   V+ $C
1494 A9 9E          LDA   # $9E
1496 8D 0D D0       STA   V+ $D
1499 A9 80          LDA   # $80
149B 8D FE 07       STA   SPOINTER+6
149E AD 17 07       LDA   $717
14A1 8D 90 13       STA   LIFE1
14A4 AD 18 07       LDA   $718
14A7 8D 95 13       STA   LIFE2
14AA 4C 4E 14       JMP   NEWSTART
14AD A9 30          LDA   # $30
14AF 8D 90 13       STA   LIFE1
14B2 A9 35          LDA   # $35
14B4 8D 95 13       STA   LIFE2
14B7 4C C7 0E       JMP   MOVESPRD

;print "change to (a square)"
14BA A9 0C          CHANGETO LDA   # $C
14BC 85 D3          STA   $D3
14BE A9 16          LDA   # $16
14C0 85 D6          STA   $D6
14C2 20 10 E5       JSR   PCURSOR
14C5 A2 00          LDX   # 0
14C7 BD D5 14       LDA   DCHANGE,X
14CA 20 D2 FF       JSR   PRINT
14CD E8              INX
14CE E0 17          CPX   # $17
14D0 D0 F5          BNE   CTLOOP
14D2 4C EC 14       JMP   PRCOLOR
14D5 05 43 48       DCHANGE .ASC "(WHT)CHANGE TO"

14DF 20 12 E9       .BYT $20,$12,$E9,$20,$9
14E6 9D 9D 9D       .BYT $9D,$9D,$9D,$98,$
      B7,$B7

;print goal color to change the squares to
14EC AD E1 03       PRCOLOR LDA   LEVEL
14EF 18             CLC
14F0 69 01          ADC   # 1
14F2 8D 86 DB       STA   $DB86
14F5 8D 87 DB       STA   $DB87
14F8 8D 88 DB       STA   $DB88
14FB 60             RTS

;print "game over"
14FC 20 4A 13       GAMEOVER JSR   CLEARBOT
14FF A9 0E          LDA   # $E
1501 85 D3          STA   $D3
1503 A9 16          LDA   # $16
1505 85 D6          STA   $D6
1507 20 10 E5       JSR   PCURSOR
150A A2 00          LDX   # 0
150C BD 31 15       GOLLOOP LDA   DATAGMVR,X
150F 20 D2 FF       JSR   PRINT

1512 E8              INX
1513 E0 0A          CPX   # $A
1515 D0 F5          BNE   GOLOOP
1517 EE D7 03       GOPAUSE INC   $3D7
151A D0 FB          BNE   GOPAUSE
151C EE D8 03       INC   $3D8
151F D0 F6          BNE   GOPAUSE
1521 EE D9 03       INC   $3D9
1524 AD D9 03       LDA   $3D9
1527 C9 05          CMP   # 5
1529 D0 EC          BNE   GOPAUSE

;start game from beginning
152B A2 00          VRESTORE LDX   # 0
152D 9A             TXS
152E 4C 0D 08       JMP   START
1531 9F             .BYT $9F
1532 47 41 4D       DATAGMVR .ASC "GAME OVER"
153B A9 0E          DEMOSET LDA   # $E
153D 8D 15 D0       STA   SPRITEON
1540 A9 60          LDA   # $60
1542 8D CB 0D       STA   EXTLZRD
1545 4C D5 12       JMP   DEMO

;move lizards in demo
1548 20 95 0D       LZRDDEMO JSR   LIZARD
154B EE EC 03       INC   $3EC
154E D0 F8          BNE   LZRDDEMO
1550 EE ED 03       INC   $3ED
1553 AD ED 03       LDA   $3ED
1556 C9 09          CMP   # 9
1558 D0 EE          BNE   LZRDDEMO
155A A9 00          LDA   # 0
155C 8D ED 03       STA   $3ED
155F 4C 37 13       JMP   DEMOPAUS
1562 A9 4C          SETSPEED LDA   # $4C
1564 8D CB 0D       STA   EXTLZRD

;set speed of level
1567 A9 10          LDA   # $10
1569 38             SEC
156A ED E1 03       SBC   LEVEL
156D 4A             LSR
156E 8D 7E 15       STA   LVLSPEED
1571 EE 7E 15       INC   LVLSPEED
1574 4C C7 0B       JMP   BIRD

;pause to control speed of level
1577 EE C1 03       MAINPAUS INC   $3C1
157A AD C1 03       AD    $3C1
157D              LVLSPD   = *+1
157E C9 00          CMP   # 0
157F D0 F6          BNE   MAINPAUS
1581 A9 00          LDA   # 0
1583 8D C1 03       STA   $3C1
1586 EE 34 03       INC   $334
1589 4C CA 0B       JMP   RTNBIRD

;select random location for egg
158C 20 97 17       EGGRND JSR   GETRND
158F C9 B0          CMP   # $B0
1591 B0 F9          BCS   EGGRND
1593 29 F8          AND   # $F8
1595 18             CLC
1596 69 06          ADC   # 6
1598 8D 0D 50       STA   $500D
159B 20 97 17       JSR   GETRND

```


The Source Code

```

159E 29 F8          AND  #F8
15A0 18            CLC
15A1 69 01        ADC  #1
15A3 8D 0C 50    STA  $500C
15A6 20 D8 0C    JSR  SLSU
15A9 A0 00        LDY  #0
15AB B1 FD        LDA  ($FD),Y
15AD C9 E9        CMP  #E9
15AF D0 DB        BNE  EGGRND
15B1 AD 0C 50    LDA  $500C
15B4 8D 0E D0    STA  V+$E
15B7 AD 0D 50    LDA  $500D
15BA 8D 0F D0    STA  V+$F
15BD A9 4C        LDA  #$4C
15BF 8D F0 0C    STA  JTR4
15C2 4C CD 15    JMP  PUTINDO
15C5 A9 60        CSLSUTR LDA  #60
15C7 8D F0 0C    STA  JTR4
15CA 4C 0A 16    JMP  PUTIN50
15CD A9 D0        PUTINDO LDA  #D0
15CF 8D E3 0C    STA  LV2
15D2 8D DA 0C    STA  LV1
15D5 4C C7 0B    JMP  BIRD
15D8

;counter until egg pause ends
1637 EE C6 03    EGGPCNTR INC  $3C6
163A AD C8 03    LDA  $3C6
163D D0 30        BNE  EXITEC
163F EE C7 03    INC  $3C7
1642 AD C7 03    LDA  $3C7
1645 C9 16        CMP  #16
1647 D0 26        BNE  EXITEC
1649 A9 00        LDA  #0
164B 8D C7 03    STA  $3C7
164E EE C8 03    INC  $3C8
1651 AD C8 03    LDA  $3C8
1654 C9 0B        CMP  #B
1656 90 08        BCC  NOFLASH

;flash bird color
1658 AD 2D D0    LDA  SCOLOR+6
165B 49 08        EOR  #8
165D 8D 2D D0    STA  SCOLOR+6
1660 AD C8 03    NOFLASH LDA  $3C8
1663 C9 10        CMP  #10
1665 D0 08        BNE  EXITEC
1667 A9 00        LDA  #0
1669 8D C8 03    STA  $3C8
166C 8D C5 03    STA  EGGPAUS
166F 4C D5 16    EXITEC JMP  ACounter
1672 EE 2E D0    COLISION INC  SCOLOR+7

;check if egg pause is on
15D8 AD C5 03    CHKEGGP LDA  EGGPAUS
15DB F0 03        BEQ  EGGPAUSE
15DD 4C 37 16    JMP  EGGPCNTR

;counter until egg is to be put on or taken off screen
15E0 EE CA 03    EGGPAUSE INC  $3CA
15E3 AD CA 03    LDA  $3CA
15E6 C9 70        CMP  #70
15E8 D0 17        BNE  EXITEGGP
15EA A9 00        LDA  #0
15EC 8D CA 03    STA  $3CA
15EF EE CB 03    INC  $3CB
15F2 AD CB 03    LDA  $3CB
15F5 D0 0A        BNE  EXITEGGP

;see if egg is to be put on or off screen
15F7 AD 0F D0    LDA  V+$F
15FA F0 08        BEQ  PUTEGGON
15FC A9 00        LDA  #0
15FE 8D 0F D0    STA  V+$F
1601 4C 72 16    EXITEGGP JMP  COLISION
1604 20 C5 15    PUTEGGON JSR  CSLSUTR
1607 4C 01 16    JMP  EXITEGGP
160A A9 50        PUTIN50 LDA  #50
160C 8D E3 0C    STA  LV2
160F 8D DA 0C    STA  LV1
1612 4C 8C 15    JMP  EGGRND

;see if bird has landed on egg
1615 AD 0C D0    CHEKKEGG LDA  V+$C
1618 CD 0E D0    CMP  V+$E
161B D0 17        BNE  NOTONEGG
161D AD 0D D0    LDA  V+$D
1620 CD 0F D0    CMP  V+$F
1623 D0 0F        BNE  NOTONEGG
1625 A2 02        LDX  #2
1627 20 EC 16    JSR  INCSCR
162A A9 7F        LDA  #$7F
162C 8D C5 03    STA  EGGPAUS
162F A9 00        LDA  #0
1631 8D 0F D0    STA  V+$F
1634 4C D8 0C    NOTONEGG JMP  SLSU

1637 EE C6 03    EGGPCNTR INC  $3C6
163A AD C8 03    LDA  $3C6
163D D0 30        BNE  EXITEC
163F EE C7 03    INC  $3C7
1642 AD C7 03    LDA  $3C7
1645 C9 16        CMP  #16
1647 D0 26        BNE  EXITEC
1649 A9 00        LDA  #0
164B 8D C7 03    STA  $3C7
164E EE C8 03    INC  $3C8
1651 AD C8 03    LDA  $3C8
1654 C9 0B        CMP  #B
1656 90 08        BCC  NOFLASH

;flash bird color
1658 AD 2D D0    LDA  SCOLOR+6
165B 49 08        EOR  #8
165D 8D 2D D0    STA  SCOLOR+6
1660 AD C8 03    NOFLASH LDA  $3C8
1663 C9 10        CMP  #10
1665 D0 08        BNE  EXITEC
1667 A9 00        LDA  #0
1669 8D C8 03    STA  $3C8
166C 8D C5 03    STA  EGGPAUS
166F 4C D5 16    EXITEC JMP  ACounter
1672 EE 2E D0    COLISION INC  SCOLOR+7

;see if bird has run into any predators
1675 A2 0A        LDX  #A
1677 BD 00 D0    CLSNLOOP LDA  V,X
167A 18            CLC
167B 7D 9F 16    ADC  DISTANCE,X
167E 38            SEC
167F ED 0C D0    SBC  V+$C
1682 C9 05        CMP  #5
1684 B0 12        BCS  NEXTCCHK
1686 BD 01 D0    LDA  V+1,X
1689 18            CLC
168A 7D A0 16    ADC  DISTANCE+1,X
168D 38            SEC
168E ED 0D D0    SBC  V+$D
1691 C9 05        CMP  #5
1693 B0 03        BCS  NEXTCCHK
1695 4C 68 17    JMP  CHKSUPER
1698 CA            NEXTCCHK DEX
1699 CA            DEX
169A 10 DB        BPL  CLSNLOOP
169C 4C 95 D0    JMP  LIZARD
169F 04 0B FF    DISTANCE .BYT 4,11,255,10,255,10,
255,10,5,5,3,4

16AB A9 60        DEATHSET LDA  #80
16AD 8D 6E 13    STA  STR1
16B0 A9 00        LDA  #0
16B2 8D 0F D4    STA  SID+$F
16B5 20 55 13    JSR  BCPAUSE
16B8 A9 8D        LDA  #8D
16BA 8D 6E 13    STA  STR1
16BD 4C 46 14    JMP  DEATH

;increase # of lives
16C0 EE 95 13    INCLIVES INC  LIFE2
16C3 AD 95 13    LDA  LIFE2
16C6 C9 3A        CMP  #3A
16C8 D0 08        BNE  LT10
16CA A9 30        LDA  #30
16CC 8D 95 13    STA  LIFE2

```

Chapter 3

```

16CF EE 90 13      INC  LIFE1
16D2 4C 7A 0B LT10 JMP  SPRSET2
16D5 EE C1 03 ACOUNTER INC  $3C1
16D8 AD C1 03      LDA  $3C1
16DB C9 04         CMP  #4
16DD D0 08         BNE  NOBIRD
16DF A9 00         LDA  #0
16E1 8D C1 03      STA  $3C1
16E4 4C C7 0B      JMP  BIRD
16E7 4C D8 15 NOBIRD JMP  CHKEGGP

:
: add to score
16EA A2 03 ADDSCORE LDX  #3
16EC FE 03 07 INCSCR  INC  $703,X
16EF BD 03 07      LDA  $703,X
16F2 C9 3A         CMP  #$3A
16F4 D0 08         BNE  EXITSCR
16F6 A9 30         LDA  #$30
16F8 9D 03 07      STA  $703,X
16FB CA            DEX
16FC 10 EE         BPL  INCSCR
16FE 4C 16 17 EXITSCR JMP  SCRTOMEM

:
: determine value to add to score
1701 C0 00 INCSCORE  CPY  #0
1703 D0 0E         BNE  EXITIS
1705 84 FC         STY  $FC
1707 A4 04         LDY  $4
1709 20 EA 16 NOOFINC JSR  ADDSCORE
170C 88           DEY
170D C0 01         CPY  #1
170F D0 F8         BNE  NOOFINC
1711 A4 FC         LDY  $FC
1713 A5 04 EXITIS  LDA  $4
1715 60           RTS

:
: put score into memory
1716 A2 04 SCRTOMEM LDX  #4
1718 BD 03 07 SMLOOP LDA  $703,X
171B 9D F3 08      STA  SCRMEM,X
171E CA            DEX
171F 10 F7         BPL  SMLOOP
1721 4C 34 17      JMP  HISCORE

:
: blank out score
1724 A2 04 BLANKSCR LDX  #4
1726 A9 30         LDA  #$30
1728 9D 03 07 BSLOOP STA  $703,X
172B CA            DEX
172C 10 FA         BPL  BSLOOP
172E 20 73 12      JSR  SETCHK
1731 4C 74 13      JMP  INCLEVEL

:
: check if high score is beaten
1734 A2 00 HISCORE  LDX  #0
1736 38           SEC
1737 BD B4 06      LDA  $6B4,X
173A FD 03 07      SBC  $703,X
173D 30 09         BMI  ISHIGH
173F E9 01         SBC  #1
1741 10 13         BPL  NOTHIGH
1743 E8           INX
1744 E0 05         CPX  #5
1746 D0 EE         BNE  HSCHECK

:
: store new high score

```

```

1748 A2 04 ISHIGH  LDX  #4
174A BD 03 07 IHLOOP LDA  $703,X
174D 9D B4 06      STA  $6B4,X
1750 9D E3 08      STA  HISCRMEM,X
1753 CA            DEX
1754 10 F4         BPL  IHLOOP
1756 60           RTS

:
: ignore collision if bird is high in super jump
1757 AD 36 03 SPRON  LDA  $336
175A C9 06         CMP  #6
175C 90 07         BCC  STILLDIE
175E C9 15         CMP  #$15
1760 B0 03         BCS  STILLDIE
1762 4C 98 16      JMP  NEXTCCHK
1765 4C AB 16 STILLDIE JMP  DEATHSET
1768 AD FE 03 CHKSUPER LDA  SPRJMP
176B D0 EA         BNE  SPRON
176D 4C AB 16      JMP  DEATHSET

:
: fill section of memory with random numbers
1770 A9 30 RNDMEM   LDA  #$30
1772 8D 7C 17      STA  CRRNTMEM+2
1775 A2 00 RNDLOOP1 LDX  #0
1777 AD 1B D4 RNDLOOP2 LDA  RANDOM
177A 9D 00 30 CRRNTMEM STA  $3000,X
177D E8           INX
177E D0 F7         BNE  RNDLOOP2
1780 EE 7C 17      INC  CRRNTMEM+2
1783 AD 7C 17      LDA  CRRNTMEM+2
1786 C9 50         CMP  #$50
1788 D0 EB         BNE  RNDLOOP1
178A A9 30         LDA  #$30
178C 8D B0 17      STA  RNDLOC+2
178F A9 00         LDA  #0
1791 8D AF 17      STA  RNDLOC+1
1794 4C B2 17      JMP  SOUNDSU2

:
: return accumulator with a random number
1797 AD AF 17 GETRND  LDA  RNDLOC+1
179A 18           CLC
179B 69 01         ADC  #1
179D 8D AF 17      STA  RNDLOC+1
17A0 AD B0 17      LDA  RNDLOC+2
17A3 69 00         ADC  #0
17A5 C9 4F         CMP  #$4F
17A7 D0 02         BNE  NOWRAP
17A9 A9 30         LDA  #$30
17AB 8D B0 17 NOWRAP STA  RNDLOC+2
17AE AD 00 30 RNDLOC LDA  $3000
17B1 60           RTS

:
: sound set-up
17B2 A9 E1 SOUNDSU2 LDA  #$E1
17B4 8D 14 D4      STA  SID+$14
17B7 A9 21         LDA  #$21
17B9 8D 12 D4      STA  SID+$12
17BC A9 0F         LDA  #$F
17BE 8D 18 D4      STA  SID+$18
17C1 A9 00         LDA  #0
17C3 8D 05 D4      STA  SID+5
17C6 8D FE 2F      STA  $2FFE
17C9 8D FF 2F      STA  $2FFF
17CC 4C F9 18      JMP  SETRSTR

:
: sound for bird jump
17CF AE 36 03 BJSOUND LDX  $336

```

The Source Code

```

17D2  BD DB 17      LDA  DATABJS,X
17D5  8D OF D4      STA  SID+$F
17D8  4C E0 08      JMP  EXITBIRD
17DB  00 80 7C DATABJS .BYT 0,$80,$7C,$78,$74
17E0  70 6E 6C      .BYT $70,$6E,$6C,$6A,$66
17E6  00 00 00      .BYT 0,0,0,0,0,0,0
17EE  00 00 00      .BYT 0,0,0,0,0,0
;sound set-up
17F5  A9 08  SOUNDSU4 LDA  #8
17F7  8D 09 D4      STA  SID+9
17FA  8D 0A D4      STA  SID+$A
17FD  A9 77      LDA  #$77
17FF  8D 06 D4      STA  SID+6
1802  8D 0D D4      STA  SID+$D
1805  A9 41      LDA  #$41
1807  8D 0B D4      STA  SID+$B
180A  A9 4C      LDA  #$4C
180C  8D DD 13      STA  JTR3
180F  A9 00      LDA  #0
1811  8D OF D4      STA  SID+$F
1814  4C C2 18      JMP  SETIRQ

;tune played in interrupt
1817  AD FE 2F TUNE  LDA  $2FFE
181A  C9 01      CMP  #1
181C  D0 1B      BNE  TUNE1
181E  AE FF 2F      LDX  $2FFF
1821  BD BA 18      LDA  DTUNE,X
1824  8D 08 D4      STA  SID+8
1827  A9 28      LDA  #$28
1829  8D 01 D4      STA  SID+1
182C  A9 81      LDA  #$81
182E  8D 04 D4      STA  SID+4
1831  A9 80      LDA  #$80
1833  8D 04 D4      STA  SID+4
1836  4C B7 18      JMP  EXITTUNE
1839  C9 0F  TUNE1  CMP  #$F
183B  D0 28      BNE  TUNE2
183D  A9 40      LDA  #$40
183F  8D 0B D4      STA  SID+$B
1842  A9 41      LDA  #$41
1844  8D 0B D4      STA  SID+$B
1847  A9 1E      LDA  #$1E
1849  8D 01 D4      STA  SID+1
184C  EE FF 2F      INC  $2FFF
184F  AE FF 2F      LDX  $2FFF
1852  BD BA 18      LDA  DTUNE,X
1855  8D 08 D4      STA  SID+8
1858  A9 81      LDA  #$81
185A  8D 04 D4      STA  SID+4
185D  A9 80      LDA  #$80
185F  8D 04 D4      STA  SID+4
1862  4C B7 18      JMP  EXITTUNE
1865  C9 10  TUNE2  CMP  #$10
1867  D0 0F      BNE  TUNE3
1869  AE FF 2F      LDX  $2FFF
186C  BD BA 18      LDA  DTUNE,X
186F  18      CLC
1870  69 02      ADC  #2
1872  8D 08 D4      STA  SID+8
1875  4C B7 18      JMP  EXITTUNE
1878  C9 18  TUNE3  CMP  #$18
187A  D0 23      BNE  TUNE4
187C  A9 40      LDA  #$40
187E  8D 0B D4      STA  SID+$B
1881  A9 41      LDA  #$41
1883  8D 0B D4      STA  SID+$B
1886  AE FF 2F      LDX  $2FFF

1889  BD BA 18      LDA  DTUNE,X
188C  18      CLC
188D  69 04      ADC  #4
188F  8D 08 D4      STA  SID+$8
1892  A9 81      LDA  #$81
1894  8D 04 D4      STA  SID+4
1897  A9 80      LDA  #$80
1899  8D 04 D4      STA  SID+4
189C  4C B7 18      JMP  EXITTUNE
189F  C9 1E  TUNE4  CMP  #$1E
18A1  D0 14      BNE  EXITTUNE
18A3  EE FF 2F      INC  $2FFF
18A6  A9 00      LDA  #0
18A8  8D FE 2F      STA  $2FFE
18AB  AD FF 2F      LDA  $2FFF
18AE  C9 08      CMP  #8
18B0  D0 05      BNE  EXITTUNE
18B2  A9 00      LDA  #0
18B4  8D FF 2F      STA  $2FFF
18B7  4C D3 18 EXITTUNE JMP  EXITIRQ
18BA  04 08 0C DTUNE .BYT 4,8,12,16,20,16,12,8

;set new irq interrupt vector
18C2  78  SETIRQ  SEI
18C3  A9 D9      LDA  #<INTRRPT
18C5  8D 14 03      STA  IRQ
18C8  A9 18      LDA  #>INTRRPT
18CA  8D 15 03      STA  IRQ+1
18CD  58      CLI
18CE  A9 00      LDA  #0
18D0  4C 33 0C      JMP  CTB1
18D3  EE FE 2F EXITIRQ INC  $2FFE
18D6  4C 31 EA      JMP  IRQENTRY
18D9  EE FD 2F INTRRPT INC  $2FFD
18DC  AD FD 2F      LDA  $2FFD
18DF  29 01      AND  #1
18E1  D0 03      BNE  JMPTUNE
18E3  4C 31 EA      JMP  IRQENTRY
18E6  4C 17 18 JMPTUNE JMP  TUNE

;check shift key to pause all game action
18E9  AD 8D 02 PAUSGAME LDA  SHIFT
18EC  D0 03      BNE  PAUSEON
18EE  4C 77 15      JMP  MAINPAUS
18F1  A9 00  PAUSEON LDA  #0
18F3  8D 0F D4      STA  SID+$F
18F6  4C E9 18      JMP  PAUSGAME

;set new vector for non-maskable interrupt (restore key)
18F9  A9 2B  SETRSTR LDA  #<VRESTORE
18FB  8D 18 03      STA  RESTORE
18FE  A9 15      LDA  #>VRESTORE
1900  8D 19 03      STA  RESTORE+1
1903  A9 00      LDA  #0
1905  8D 10 D0      STA  V+$10
1908  8D 1B D0      STA  V+$1B
190B  4C F5 17      JMP  SOUNDSU4

```

Bump-N-Run

Source code by Tim J. Midkiff

```

C000          * = $C000
C000          = $C700
C000          M
C000          VSP
C000          ABSVSP
C000          COUNTER
C000          MSGBT
C000          JOY
C000          SPXYNO
C000          COL
C000          COLREG
C000          SPCTRL
C000          BALLMSK
C000          BALLVX
C000          BALLVY
C000          FIREON
C000          SCORE
C000          = M+31

; reset game

C000 A9 00 RSTGAME LDA #0
C002 8D 20 D0 STA $D020
C005 8D 21 D0 STA $D021
C008 8D 20 C7 STA $C720
C00B 8D 21 C7 STA $C721
C00E AD 16 03 LDA $0316
C011 8D 18 03 STA $0318
C014 AD 17 03 LDA $0317
C017 8D 19 03 STA $0319
C01A A9 14 LDA #$14
C01C 8D 18 D0 STA $D018
C01F A9 80 LDA #$80
C021 8D 91 02 STA $0291
C024 A9 93 LDA #$93
C026 20 D2 FF JSR $FFD2
C029 A9 00 LDA #0
C02B A2 80 LDX #$80
C02D 9D 80 3F LPO STA $3F80,X
C030 CA DEX
C031 10 FA BPL LPO
C033 A9 18 LDA #$18
C035 8D 99 3F STA $3F99
C038 8D A2 3F STA $3FA2
C03B A9 3C LDA #$3C
C03D 8D 9C 3F STA $3F9C
C040 8D 9F 3F STA $3F9F
C043 A2 33 LDX #$33
C045 BD 4C C6 LP12 LDA SP12,X
C048 9D C4 3F STA $3FC4,X
C04B CA DEX
C04C 10 F7 BPL LP12
C04E 20 1A C5 JSR INTSCN
C051 78 SEI
C052 A9 05 LDA #5
C054 8D 14 03 STA $0314
C057 A9 C2 LDA #$C2
C059 8D 15 03 STA $0315
C05C 58 CLI

; reset after goal

C05D 20 80 C2 RSTGOAL JSR INTSP
C060 A9 FF LDA #$FF

C062 8D 18 C7 STA BALLMSK
C065 A9 00 LDA #0
C067 8D 15 C7 STA COL
C06A 8D 16 C7 STA COLREG
C06D 8D 17 C7 STA SPCTRL
C070 8D 1F C7 STA SCORE
C073 8D 19 C7 STA BALLVX
C076 8D 1A C7 STA BALLVY
C079 A2 05 LDX #5
C07B A9 00 LDA #0
C07D 9D 00 C7 STA VSP,X
C080 9D 06 C7 STA ABSVSP,X
C083 A9 5A LDA #$5A
C085 9D 0C C7 STA COUNTER,X
C088 CA DEX
C089 10 F0 BPL INTVELS

; main loop

C08B AD 16 C7 MNLP LDA COLREG
C08E 29 03 AND #3
C090 C9 03 CMP #3
C092 D0 50 BNE NOCOL
C094 AD 15 C7 LDA COL
C097 D0 48 BNE PATCH
C099 AE 00 C7 LDX VSP
C09C AC 01 C7 LDY $C701
C09F AD 02 C7 LDA $C702
C0A2 8D 00 C7 STA VSP
C0A5 AD 03 C7 LDA $C703
C0A8 8D 01 C7 STA $C701
C0AB 8E 02 C7 STX $C702
C0AE 8C 03 C7 STY $C703
C0B1 AD 17 C7 LDA SPCTRL
C0B4 F0 1B BEQ ENDCTRL
C0B6 C9 01 CMP #1
C0B8 F0 D0 BEQ SP1CTRL
C0BA A9 01 LDA #1
C0BC 8D 17 C7 STA SPCTRL
C0BF A9 0B LDA #$0B
C0C1 8D 15 D0 STA $D015
C0C4 4C D1 C0 JMP ENDCTRL
C0C7 A9 03 LDA #3
C0C9 8D 17 C7 STA SPCTRL
C0CC A9 13 LDA #$13
C0CE 8D 15 D0 STA $D015
C0D1 A2 03 LDX #3
C0D3 BD 06 C7 ENDCTRL LDA ABSVSP,X
C0D6 9D 0C C7 STCNT STA COUNTER,X
C0D9 CA DEX
C0DA 10 F7 BPL STCNT
C0DC A9 01 LDA #1
C0DE 8D 15 C7 STA COL
C0E1 4C 2A C1 PATCH JMP MOVEMENT
C0E4 AD 17 C7 NOCOL LDA SPCTRL
C0E7 D0 18 BNE CTRL
C0E9 AD 04 C7 LDA $C704
C0EC CD 19 C7 CMP BALLVX
C0EF D0 0B BNE NOMSK
C0F1 AD 05 C7 LDA $C705
C0F4 CD 1A C7 CMP BALLVY
C0F7 D0 03 BNE NOMSK

```

The Source Code

```

COF9 4C 01 C1      JMP  CTRL
COFC A9 FF          NOMSK
COFE 8D 18 C7      LDA  BALLMSK
C101 A9 00          CTRL  LDA  #0
C103 8D 15 C7      STA  COL
C106 AD 16 C7      LDA  COLREG
C109 2D 18 C7      AND  BALLMSK
C10C 29 07          AND  #7
C10E C9 05          CMP  #5
C110 D0 0A          BNE  CAPSP2
C112 A9 01          LDA  #1
C114 8D 17 C7      STA  SPCTRL
C117 A9 0B          LDA  #50B
C119 8D 15 D0      STA  $D015
C11C C9 06          CAPSP2 CMP  #6
C11E D0 0A          BNE  MOVEMENT
C120 A9 03          LDA  #3
C122 8D 17 C7      STA  SPCTRL
C125 A9 13          LDA  #$13
C127 8D 15 D0      STA  $D015
C12A A2 00          MOVEMENT LDX  #0
C12C 8E 14 C7      STA  SPXYNO
C12F A9 09          LDA  #9
C131 8D 12 C7      STA  MSGBT
C134 AD 01 DC       LDA  $DC01
C137 8D 13 C7      STA  JOY
C13A 20 DF C2      JSR  CKVEL
C13D A2 02          LDX  #2
C13F 8E 14 C7      STA  SPXYNO
C142 A9 12          LDA  #$12
C144 8D 12 C7      STA  MSGBT
C147 AD 00 DC       LDA  $DC00
C14A 8D 13 C7      STA  JOY
C14D 20 DF C2      JSR  CKVEL
C150 A2 04          LDX  #4
C152 8E 14 C7      STA  SPXYNO
C155 A9 24          LDA  #$24
C157 8D 12 C7      STA  MSGBT
C15A AD 04 C7      LDA  $C704
C15D 8D 19 C7      STA  BALLVX
C160 AD 05 C7      LDA  $C705
C163 8D 1A C7      STA  BALLVY
C166 BD 0C C7      LDA  COUNTER,X
C169 DD 06 C7      CMP  ABSVSP,X
C16C F0 06          BEQ  BVELY
C16E DE 0C C7      DEC  COUNTER,X
C171 4C 7E C1      JMP  CKBVX
C174 A9 5A          BVELY  LDA  #$5A
C176 9D 0C C7      STA  COUNTER,X
C179 A0 02          LDY  #2
C17B 20 E4 C3      JSR  MVSPY
C17E EE 14 C7      CKBVX INC  SPXYNO
C181 AE 14 C7      INC  SPXYNO
C184 BD 0C C7      LDA  COUNTER,X
C187 DD 06 C7      CMP  ABSVSP,X
C18A F0 06          BEQ  BVELX
C18C DE 0C C7      DEC  COUNTER,X
C18F 4C 02 C2      JMP  REPEAT
C192 A9 5A          BVELX  LDA  #$5A
C194 9D 0C C7      STA  COUNTER,X
C197 A0 02          LDY  #2
C199 20 20 C4      JSR  MVSPX
C19C AE 1F C7      LDX  SCORE
C19F F0 61          BEQ  REPEAT
C1A1 A9 00          LDA  #0
C1A3 8D 15 D0      STA  $D015
C1A6 FE 1F C7      INC  SCORE,X
C1A9 A2 01          LDX  #1
C1AB 18            CLC
C1AC 20 F0 FF      JSR  $FFF0
C1AF AE 1F C7      LDX  SCORE
C1B2 BD 1F C7      LDA  SCORE,X

C1B5 0A            ASL  A
C1B6 AA            TAX
C1B7 BD 80 C6      LDA  NOS,X
C1BA 20 D2 FF      JSR  $FFD2
C1BD E8            INX
C1BE BD 80 C6      LDA  NOS,X
C1C1 20 D2 FF      JSR  $FFD2
C1C4 A2 64          LDX  #$64
C1C6 A0 FF          LDY  #$FF
C1C8 88            DEY
C1C9 D0 FD          BNE  YLP
C1CB CA            DEX
C1CC D0 F8          BNE  XLP
C1CE AE 1F C7      LDX  SCORE
C1D1 BD 1F C7      LDA  SCORE,X
C1D4 C9 0F          CMP  #50F
C1D6 D0 27          BNE  NOEND
C1D8 A2 0C          LDX  #50C
C1DA A0 10          LDY  #$10
C1DC 18            CLC
C1DD 20 F0 FF      JSR  $FFF0
C1E0 A2 2C          LDX  #$2C
C1E2 20 E3 C5      JSR  TITLE
C1E5 A2 0E          LDX  #50E
C1E7 A0 0B          LDY  #50B
C1E9 18            CLC
C1EA 20 F0 FF      JSR  $FFF0
C1ED A2 41          LDX  #$41
C1EF 20 E3 C5      JSR  TITLE
C1F2 AD 01 DC       LDA  $DC01
C1F5 0D 00 DC       ORA  $DC00
C1F8 29 10          AND  #510
C1FA D0 F6          BNE  ENDLP
C1FC 4C 00 C0      JMP  RSTGAME
C1FF 4C 5D C0       JMP  RSTGAL
C202 4C 8B C0       JMP  MNLP
;
; handle irq sprite collisions
;
C205 AD 1E D0      IRQ  LDA  $D01E
C208 8D 16 C7      STA  COLREG
C20B 29 03          AND  #3
C20D C9 03          CMP  #3
C20F D0 0E          BNE  RETIRQ
C211 A2 00          LDX  #0
C213 20 22 C2      JSR  SEP
C216 E8            INX
C217 20 22 C2      JSR  SEP
C21A A9 0F          LDA  #50F
C21C 8D 19 D0      STA  $D019
C21F 4C 31 EA       RETIRQ JMP  $EA31
;
; separate players after collision
;
C222 BD 00 D0      SEP  LDA  $D000,X
C225 DD 02 D0      CMP  $D002,X
C228 F0 55          BEQ  ENDSEP
C22A 10 2B          BPL  P1GTP2
C22C DE 00 D0      DEC  $D000,X
C22F DE 06 D0      DEC  $D006,X
C232 10 0C          BPL  SCNT1
C234 E0 01          CPX  #1
C236 F0 08          BEQ  SCNT1
C238 A9 F6          LDA  #5F6
C23A 2D 10 D0      AND  $D010
C23D 8D 10 D0      STA  $D010
C240 FE 02 D0      SCNT1 INC  $D002,X
C243 FE 08 D0      INC  $D008,X
C246 D0 37          BNE  ENDSEP
C248 E0 01          CPX  #1
C24A F0 33          BEQ  ENDSEP
C24C A9 12          LDA  #$12

```

Chapter 3

```

C24E 0D 10 D0      ORA  $D010
C251 8D 10 D0      STA  $D010
C254 4C 7F C2      JMP  ENDSEP
C257 FE 00 D0 P1GTP2  INC  $D000,X
C25A FE 06 D0      INC  $D006,X
C25D D0 0C          BNE  SCNT2
C25F E0 01          CPX  #1
C261 F0 08          BEQ  SCNT2
C263 A9 09          LDA  #9
C265 0D 10 D0      ORA  $D010
C268 8D 10 D0      STA  $D010
C26B DE 02 D0 SCNT2  DEC  $D002,X
C26E DE 08 D0      DEC  $D008,X
C271 10 0C          BPL  ENDSEP
C273 E0 01          CPX  #1
C275 F0 08          BEQ  ENDSEP
C277 A9 ED          LDA  #$ED
C279 2D 10 D0      AND  $D010
C27C 8D 10 D0      STA  $D010
C27F 60          ENDSEP
:
: initialize sprites
:
C280 A9 4A          INTSP LDA  #$4A
C282 8D 00 D0      STA  $D000
C285 8D 06 D0      STA  $D006
C288 A9 90          LDA  #$90
C28A 8D 01 D0      STA  $D001
C28D 8D 03 D0      STA  $D003
C290 8D 05 D0      STA  $D005
C293 8D 07 D0      STA  $D007
C296 8D 09 D0      STA  $D009
C299 8D 0B D0      STA  $D00B
C29C A9 0F          LDA  #$0F
C29E 8D 02 D0      STA  $D002
C2A1 8D 08 D0      STA  $D008
C2A4 A9 AC          LDA  #$AC
C2A6 8D 0A D0      STA  $D00A
C2A9 8D 04 D0      STA  $D004
C2AC A9 12          LDA  #$12
C2AE 8D 10 D0      STA  $D010
C2B1 A9 FF          LDA  #$FF
C2B3 8D F8 07      STA  $07F8
C2B6 8D F9 07      STA  $07F9
C2B9 A9 FE          LDA  #$FE
C2BB 8D FA 07      STA  $07FA
C2BE 8D FB 07      STA  $07FB
C2C1 8D FC 07      STA  $07FC
C2C4 A9 04          LDA  #4
C2C6 8D 27 D0      STA  $D027
C2C9 A9 07          LDA  #7
C2CB 8D 28 D0      STA  $D028
C2CE A9 01          LDA  #1
C2D0 8D 29 D0      STA  $D029
C2D3 8D 2A D0      STA  $D02A
C2D6 8D 2B D0      STA  $D02B
C2D9 A9 07          LDA  #7
C2DB 8D 15 D0      STA  $D015
C2DE 60          RTS
:
: update velocity and sprites
:
C2DF BD 0C C7 CKVEL  LDA  COUNTER,X
C2E2 DD 06 C7      CMP  ABSVSP,X
C2E5 F0 0C          BEQ  VELY
C2E7 DE 0C C7      DEC  COUNTER,X
C2EA 4E 13 C7      LSR  JOY
C2ED 4E 13 C7      LSR  JOY
C2F0 4C 00 C3      JMP  CKVELX
C2F3 A9 5A          LDA  #$5A
C2F5 9D 0C C7      STA  COUNTER,X
C2F8 20 AD C3      JSR  VELSP
C2FB AO 00          LDY  #0
C2FD 20 E4 C3      JSR  MVSPY
C300 EE 14 C7 CKVELX  INC  SPXYNO
C303 AE 14 C7      LDX  SPXYNO
C306 BD 0C C7      LDA  COUNTER,X
C309 DD 06 C7      CMP  ABSVSP,X
C30C F0 0C          BEQ  VELX
C30E DE 0C C7      DEC  COUNTER,X
C311 4E 13 C7      LSR  JOY
C314 4E 13 C7      LSR  JOY
C317 4C 27 C3      JMP  CKFIRE
C31A A9 5A          LDA  #$5A
C31C 9D 0C C7      STA  COUNTER,X
C31F 20 AD C3      JSR  VELSP
C322 AO 00          LDY  #0
C324 20 20 C4      JSR  MVSPX
C327 AD 14 C7 CKFIRE  LDA  SPXYNO
C32A 4D 17 C7      EOR  SPCTRL
C32D D0 7D          BNE  ENDCV
C32F AE 14 C7      LDX  SPXYNO
C332 4E 13 C7      LSR  JOY
C335 B0 08          BCS  NOFIRE
C337 BD 1B C7      LDA  FIREON,X
C33A F0 08          BEQ  NEWFIRE
C33C 4C AC C3      JMP  ENDCV
C33F A9 00          LDA  #0
C341 9D 1B C7      STA  FIREON,X
C344 4C AC C3      JMP  ENDCV
C347 BD 00 C7 NEWFIRE  LDA  VSP,X
C34A 8D 05 C7      STA  $C705
C34D BD 06 C7      LDA  ABSVSP,X
C350 8D 08 C7      STA  $C708
C353 BD FF C6      LDA  $C6FF,X
C356 8D 04 C7      STA  $C704
C359 BD 05 C7      LDA  $C705,X
C35C 8D 0A C7      STA  $C70A
C35F A9 00          LDA  #0
C361 9D FF C6      STA  $C6FF,X
C364 9D 00 C7      STA  VSP,X
C367 9D 05 C7      STA  $C705,X
C36A 9D 06 C7      STA  ABSVSP,X
C36D BD 00 D0      LDA  $D000,X
C370 8D 05 D0      STA  $D005
C373 8D 08 D0      STA  $D008
C376 BD FF CF      LDA  $CFFF,X
C379 8D 04 D0      STA  $D004
C37C 8D 0A D0      STA  $D00A
C37F AD 12 C7      LDA  MSGBT
C382 2D 10 D0      AND  $D010
C385 F0 08          BEO  BLT
C387 AD 10 D0      LDA  $D010
C38A 09 24          ORA  #$24
C38C 8D 10 D0      STA  $D010
C38F 4C 9A C3      JMP  CNTCV
C392 AD 10 D0 BLT    LDA  $D010
C395 29 DB          AND  #$DB
C397 8D 10 D0      STA  $D010
C39A A9 07          LDA  #7
C39C 8D 15 D0      STA  $D015
C39F A9 00          LDA  #0
C3A1 8D 17 C7      STA  SPCTRL
C3A4 AD 12 C7      LDA  MSGBT
C3A7 49 FF          EOR  #$FF
C3A9 8D 18 C7      STA  BALLMSK
C3AC 60          RTS
:
: add acceleration
:
C3AD AC 14 C7 VELSP   LDY  SPXYNO
C3B0 B9 00 C7      LDA  VSP,Y
C3B3 D0 05          BNE  VNO
C3B5 A2 00          LDX  #0

```

The Source Code

```

C3B7 4C C5 C3      JMP  VEND
C3BA 29 80  VNO    AND  #$80
C3BC  F0 05      BEQ  VYPOS
C3BE  A2 01      LDX  #1
C3C0  4C C5 C3    JMP  VEND
C3C3  A2 02      LDX  #2
C3C5  BC F9 C5    LDY  JOY0,X
C3C8  4E 13 C7    LSR  JOY
C3CB  B0 03      BCS  VEL24
C3CD  BC FC C5    LDY  JOY13,X
C3D0  4E 13 C7    LSR  JOY
C3D3  B0 03      BCS  JYEND
C3D5  BC FF C5    LDY  JOY24,X
C3D8  AE 14 C7    LDX  SPXYNO
C3DB  98
C3DC  18
C3DD  7D 00 C7   ADC  VSP,X
C3E0  9D 00 C7   STA  VSP,X
C3E3  60          RTS

; move sprite vertically

C3E4  BD 00 C7   MVSPY
C3E7  F0 33      BEQ  MYST
C3E9  29 80      AND  #$80
C3EB  F0 1A      BEQ  MYDN
C3ED  BD 01 D0   LDA  $D001,X
C3F0  D9 02 C6   CMP  YBND,Y
C3F3  B0 06      BCS  DECY
C3F5  20 0E C5   JSR  NEGV
C3F8  4C 1C C4   JMP  MYST
C3FB  DE 01 D0   DECY
C3FE  DE 07 D0   DEC  $D007,X
C401  20 EA C4   JSR  STRING
C404  4C 1F C4   JMP  MYEND
C407  BD 01 D0   MYDN
C40A  C8
C40B  D9 02 C6   CMP  YBND,Y
C40E  90 06      BCC  INCY
C410  20 0E C5   JSR  NEGV
C413  4C 89 C4   JMP  MXNST
C416  FE 01 D0   INCY
C419  FE 07 D0   INC  $D007,X
C41C  20 EA C4   MYST
C41F  60          MYEND

; move sprite horizontally

C420  BD 00 C7   MVSPX
C423  F0 60      BEQ  MXST
C425  10 31      BPL  MXRT
C427  BD FF CF   LDA  $CFFF,X
C42A  D9 06 C6   CMP  XBND,Y
C42D  B0 13      BCS  DECX
C42F  AD 10 D0   LDA  $D010
C432  2D 12 C7   AND  MSGBT
C435  D0 0B      BNE  DECX
C437  20 B4 C4   JSR  CKSCLT
C43A  B0 06      BCS  DECX
C43C  20 0E C5   JSR  NEGV
C43F  4C 85 C4   JMP  MXST
C442  DE FF CF   DECX
C445  DE 05 D0   DEC  $D005,X
C448  10 3F      BPL  MXNST
C44A  AD 12 C7   LDA  MSGBT
C44D  49 FF      EOR  #$FF
C44F  2D 10 D0   AND  $D010
C452  8D 10 D0   STA  $D010
C455  4C 89 C4   JMP  MXNST
C458  BD FF CF   MXRT
C45B  C8
C45C  D9 06 C6   CMP  XBND,Y
C45F  90 13      BCC  INCX
C461  AD 10 D0   LDA  $D010
C464  2D 12 C7   AND  MSGBT
C467  F0 0B      BEQ  INCX
C469  20 8D C4   JSR  CKSCLT
C46C  B0 06      BCS  INCX
C46E  20 0E C5   JSR  NEGV
C471  4C 89 C4   JMP  MXNST
C474  FE FF CF   INCX
C477  FE 05 D0   INC  $D005,X
C47A  D0 09      BNE  MXST
C47C  AD 12 C7   LDA  MSGBT
C47F  0D 10 D0   ORA  $D010
C482  8D 10 D0   STA  $D010
C485  20 EA C4   MXST
C488  60
C489  20 EA C4   MXNST
C48C  60

; check left border

C48D  E0 05      CKSCLT
C48F  D0 57      BNE  ENDCLC
C491  AD 17 C7   LDA  SPCTRL
C494  D0 52      BNE  ENDCLC
C496  BD FF CF   LDA  $CFFF,X
C499  C9 36      CMP  #$36
C49B  10 06      BPL  CKGLLT
C49D  20 DB C4   JSR  CKSC
C4A0  4C B3 C4   JMP  ENDSCL

; check left goal

C4A3  C9 38      CKGLLT
C4A5  30 3F      BMI  ENDSEC
C4A7  A9 9C      LDA  #$9C
C4A9  20 D2 FF   JSR  $FFD2
C4AC  A9 01      LDA  #1
C4AE  8D 1F C7   STA  SCORE
C4B1  A0 0A      LDY  #$0A
C4B3  60          RTS

; check right goal

C4B4  E0 05      CKSCLT
C4B6  D0 30      BNE  ENDCLC
C4B8  AD 17 C7   LDA  SPCTRL
C4BB  D0 2B      BNE  ENDCLC
C4BD  BD FF CF   LDA  $CFFF,X
C4C0  C9 24      CMP  #$24
C4C2  30 06      BMI  CKGLRT
C4C4  20 DB C4   JSR  CKSC
C4C7  4C DA C4   JMP  ENDCSR

; check right goal

C4CA  C9 20      CKGLRT
C4CC  10 18      BPL  ENDSEC
C4CE  A9 9E      LDA  #$9E
C4D0  20 D2 FF   JSR  $FFD2
C4D3  A9 02      LDA  #2
C4D5  8D 1F C7   STA  SCORE
C4D8  A0 23      LDY  #$23
C4DA  60          RTS

; check score

C4DB  AD 05 D0   CKSC
C4DE  C9 7B      CMP  #$7B
C4E0  30 06      BMI  ENDCLC
C4E2  C9 A6      CMP  #$A6
C4E4  10 02      BPL  ENDCLC

```

Chapter 3

```

C4E6 38      ENDSEC      SEC
C4E7 60      RTS
C4E8 18      ENDCLC     CLC
C4E9 60      RTS

: update velocity
C4EA BD 00 C7 STRING    LDA  VSP,X
C4ED 30 0D      BMI  NEGVSP
C4EF C9 52      CMP  #52
C4F1 30 05      BMI  STPABS
C4F3 A9 52      LDA  #52
C4F5 9D 00 C7   STA  VSP,X
C4F8 9D 06 C7   STA  ABSVSP,X
C4FB 60      RTS
C4FC C9 AE      NEGVSP  CMP  #SAE
C4FE 10 05      BPL  STNABS
C500 A9 AE      LDA  #SAE
C502 9D 00 C7   STA  VSP,X
C505 49 FF      STNABS  EOR  #5FF
C507 9D 06 C7   STA  ABSVSP,X
C50A FE 06 C7   INC  ABSVSP,X
C50D 60      RTS

: reverse velocity
C50E BD 00 C7 NEGV     LDA  VSP,X
C511 49 FF      EOR  #5FF
C513 9D 00 C7   STA  VSP,X
C516 FE 00 C7   INC  VSP,X
C519 60      RTS

: initialize screen
C51A A9 81      INTSCN  LDA  #81
C51C 20 D2 FF   JSR  $FFD2
C51F A2 00      LDX  #0
C521 A0 02      LDY  #2
C523 18      CLC
C524 20 F0 FF   JSR  $FFF0
C527 A9 B0      LDA  #80
C529 20 D2 FF   JSR  $FFD2
C52C A2 21      LDX  #21
C52E 20 C7 C5   JSR  HLINE
C531 A9 AE      LDA  #SAE
C533 20 D2 FF   JSR  $FFD2
C536 A2 02      LDX  #2
C538 20 F0 FF   JSR  $FFF0
C53B A9 ED      LDA  #8E
C53D 20 D2 FF   JSR  $FFD2
C540 A2 21      LDX  #21
C542 20 C7 C5   JSR  HLINE
C545 A9 BD      LDA  #8D
C547 20 D2 FF   JSR  $FFD2
C54A A2 01      LDX  #1
C54C C8      INY
C54D 20 F0 FF   JSR  $FFF0
C550 A2 0A      LDX  #0A
C552 20 E3 C5   JSR  TITLE
C555 A2 01      LDX  #1
C557 A0 0F      LDY  #0F
C559 20 F0 FF   JSR  $FFF0
C55C A2 21      LDX  #21
C55E 20 E3 C5   JSR  TITLE
C561 A2 01      LDX  #1
C563 A0 1C      LDY  #1C
C565 20 F0 FF   JSR  $FFF0
C568 A2 15      LDX  #15
C56A 20 E3 C5   JSR  TITLE
C56D A9 99      LDA  #99
C56F 20 D2 FF   JSR  $FFD2
C572 A2 03      LDX  #3

C574 A0 02      LDY  #2
C576 20 F0 FF   JSR  $FFF0
C579 A9 B0      LDA  #80
C57B 20 D2 FF   JSR  $FFD2
C57E A2 21      LDX  #21
C580 20 C7 C5   JSR  HLINE
C583 A9 AE      LDA  #SAE
C585 20 D2 FF   JSR  $FFD2
C588 A2 16      LDX  #16
C58A 20 F0 FF   JSR  $FFF0
C58D A9 AD      LDA  #AD
C58F 20 D2 FF   JSR  $FFD2
C592 A2 21      LDX  #21
C594 20 C7 C5   JSR  HLINE
C597 A9 BD      LDA  #8D
C599 20 D2 FF   JSR  $FFD2
C59C A2 04      LDX  #4
C59E 20 F0 FF   JSR  $FFF0
C5A1 A9 05      LDA  #5
C5A3 20 D0 C5   JSR  VLINE
C5A6 A2 10      LDX  #10
C5A8 20 F0 FF   JSR  $FFF0
C5AB A9 05      LDA  #5
C5AD 20 D0 C5   JSR  VLINE
C5B0 A2 04      LDX  #4
C5B2 A0 25      LDY  #25
C5B4 20 F0 FF   JSR  $FFF0
C5B7 A9 05      LDA  #5
C5B9 20 D0 C5   JSR  VLINE
C5BC A2 10      LDX  #10
C5BE 20 F0 FF   JSR  $FFF0
C5C1 A9 05      LDA  #5
C5C3 20 D0 C5   JSR  VLINE
C5C6 60      RTS

: draw horizontal line
C5C7 A9 C0      HLINE  LDA  #C0
C5C9 20 D2 FF   JSR  $FFD2
C5CC CA      DEX
C5CD 10 F8      BPL  HLINE
C5CF 60      RTS

: draw vertical line
C5D0 8D 22 C7   VLINE  STA  C722
C5D3 A9 DD      VCHAR  LDA  #DD
C5D5 20 D2 FF   JSR  $FFD2
C5D8 E8      INX
C5D9 18      CLC
C5DA 20 F0 FF   JSR  $FFF0
C5DD CE 22 C7   DEC  C722
C5E0 10 F1      BPL  VCHAR
C5E2 60      RTS

: print string
C5E3 BC 0A C6    TITLE  LDY  STRINGS,X
C5E6 CA      CA      DEX
C5E7 BD 0A C6    LDA  STRINGS,X
C5EA 20 D2 FF   JSR  $FFD2
C5ED CA      DEX
C5EE BD 0A C6    TITLP  LDA  STRINGS,X
C5F1 20 D2 FF   JSR  $FFD2
C5F4 CA      DEX
C5F5 88      DEY
C5F6 D0 F6      BNE  TITLP
C5F8 60      RTS

: data
C5F9 00 01 FF   JOYO  .BYT 0,1,-1
C5FC FF FF FE   JOY13 .BYT -1,-1,-2

```


The Source Code

C5FF	01 02 01	JOY24	.BYT	1,2,1
C602	4F D2 48	YBND	.BYT	79,210,72,217
C606	2C 2D 25	XBND	.BYT	44,45,37,52
C60A	30 20 3A	STRINGS	.ASC	"0 :ELPRUP"
C613	9C 09		.BYT	156,9
C615	30 20 3A		.ASC	"0 :WOLLEY"
C61E	9E 09		.BYT	158,9
C620	4E 55 52		.ASC	"NUR-N-PMUB"
C62A	9F 0A		.BYT	159,10
C62C	52 45 56		.ASC	"REVO EMAG"
C635	9F 09		.BYT	159,9
C637	54 45 53		.ASC	"TESER OT ERIF SS ERP"
C64A	9A 13		.BYT	154,19
C64C	18 00 00	SP12	.BYT	24,0,0,24
C650	00 00 18		.BYT	0,0,24,0,0,24,0,0
C658	3C 00 00		.BYT	60,0,0,102,0,0,195,0
C660	01 81 80		.BYT	1,129,128,63,0,252, 63,0
C668	FC 01 81		.BYT	252,1,129,128,0,195 ,0,0
C670	66 00 00		.BYT	102,0,0,60,0,0,24,0
C678	00 18 00		.BYT	0,24,0,0,24,0,0,24
C680	A0 30 A0	NOS	.BYT	160,48,160,49
C684	A0 32 A0		.BYT	160,50,160,51
C688	A0 34 A0		.BYT	160,52,160,53
C68C	A0 36 A0		.BYT	160,54,160,55
C690	A0 38 A0		.BYT	160,56,160,57
C694	31 30 31		.BYT	49,48,49,49
C698	31 32 31		.BYT	49,50,49,51
C69C	31 34 31		.BYT	49,52,49,53

Ringside Karate

Source code by Anthony Bertram

3000	* = \$3000	3148	FC 00 01	.BYTE	252,0,1,85,0,3,255,0
; white sprites		3150	03 FA 00	.BYTE	3,250,0,3,251,0,3,1
3000	00 OF C0				86
	.BYTE	3158	00 03 AA	.BYTE	0,3,170,0,2,170,0,2
3008	FO 00 3F	3160	AO 00 03	.BYTE	160,0,3,160,0,5,231,
	.BYTE				0
3010	3B A8 00	3168	15 75 00	.BYTE	21,117,0,29,85,0,29,
	.BYTE				85
3018	00 2A 00	3170	00 17 55	.BYTE	0,23,85,0,21,169,0,
3020	70 01 57				53
	.BYTE	3178	A9 00 0F	.BYTE	169,0,15,87,0,1,87,0
3028	5D 55 03	3180	00 00 00	.BYTE	0,0,0,0,0,0,0,0
	.BYTE	3188	00 00 00	.BYTE	0,0,0,0,0,0,0,0
3030	00 36 A8	3190	00 00 00	.BYTE	0,0,0,0,0,128,0,0
	.BYTE	3198	00 00 00	.BYTE	0,0,0,0,0,0,0,0
3038	70 00 15	31A0	00 00 00	.BYTE	0,0,0,0,255,194,128,
	.BYTE				85
3040	00 00 00	31A8	6A 80 55	.BYTE	106,128,85,106,128,
3048	00 00 00				95,192,0
3050	00 00 00	31B0	70 00 00	.BYTE	112,0,0,192,0,0,0,0
3058	00 00 00	31B8	00 00 00	.BYTE	0,0,0,0,0,0,0,0
3060	00 02 80	31C0	00 01 57	.BYTE	0,1,87,0,3,255,0,1
	.BYTE	31C8	57 00 01	.BYTE	87,0,1,87,0,1,87,0
3068	00 00 6C	31D0	01 55 00	.BYTE	1,85,0,1,85,0,1,85
3070	C0 00 00	31D8	00 01 5D	.BYTE	0,1,93,0,1,92,0,1
3078	00 00 00	31E0	5C 00 05	.BYTE	92,0,5,92,0,21,112,0
3080	00 15 70	31E8	55 C0 01	.BYTE	85,192,1,87,0,5,92,0
	.BYTE	31F0	15 70 0F	.BYTE	21,112,15,10,112,25
3088	5C 01 55				5,2,175
3090	55 57 05	31F8	FF 0F F0	.BYTE	255,15,240,0,0,0,0,0
	.BYTE	3200	00 00 00	.BYTE	0,0,0,0,0,0,0,0
3098	05 70 15	3208	00 00 00	.BYTE	0,0,0,0,192,0,0,112
	.BYTE	3210	00 00 5C	.BYTE	0,0,92,0,0,87,0,0
30A0	05 15 00	3218	55 C0 00	.BYTE	85,192,0,21,192,0,5
30A8	00 01 17				3,192
30B0	17 3F C1	3220	00 35 C0	.BYTE	0,53,192,0,5,192,0,5
	.BYTE	3228	C0 00 05	.BYTE	192,0,5,192,0,5,192,
30B8	3F 00 00				0
30C0	00 00 00	3230	C5 C0 00	.BYTE	197,192,0,197,192,0,
30C8	00 00 00				250,176
30D0	00 00 C0	3238	00 FF F0	.BYTE	0,255,240,0,0,0,0,0
30D8	C0 00 00	3240	00 3F 00	.BYTE	0,63,0,0,85,64,0,255
	.BYTE	3248	C0 00 FE	.BYTE	192,0,254,128,0,254,
30E0	00 70 00				192,0
	.BYTE	3250	EE A0 00	.BYTE	238,160,0,234,128,0,
30E8	00 00 70				170,128
30F0	70 00 00	3258	00 A8 00	.BYTE	0,168,0,0,168,0,0,2
	.BYTE				34
30F8	00 00 00	3260	D5 00 7A	.BYTE	213,0,122,213,1,95,
3100	00 15 70	3268	55 50 05	.BYTE	85,1
	.BYTE				85,80,5,85,80,5,85,
3108	5F 01 55				80
3110	55 57 01	3270	15 D5 54	.BYTE	21,213,84,23,197,87
	.BYTE				21,165
3118	01 70 15	3278	5D 05 A1	.BYTE	93,5,161,117,0,0,21
	.BYTE				3,0
3120	05 01 70	3280	00 00 00	.BYTE	0,0,0,0,0,0,0,0
3128	5C 01 00	3288	00 00 00	.BYTE	0,0,0,0,0,0,0,0
3130	00 6F F1	3290	A0 00 00	.BYTE	160,0,0,160,0,2,176,
	.BYTE				0
3138	0F 00 00	3298	6A C0 00	.BYTE	106,192,0,107,0,0,1
3140	00 00 00				08,0

The Source Code

32A0	00 70 00	.BYTE	0,112,0,0,192,0,24,0	3460	00 00 00	.BYTE	0,0,0,0,0,0,0,0
32A8	01 58 00	.BYTE	1,88,0,21,88,0,85,92	3468	03 F0 00	.BYTE	3,240,0,5,84,0,15,2
32B0	05 55 5C	.BYTE	5,85,92,21,85,240,8				52
			5,92	3470	00 0F E8	.BYTE	0,15,232,0,15,236,0,
32B8	00 55 70	.BYTE	0,85,112,0,85,192,0,				14
			0	3478	EA 00 0E	.BYTE	234,0,14,168,0,10,1
32C0	00 03 55	.BYTE	0,3,85,0,0,85,0,0				68,2
32C8	55 00 00	.BYTE	85,0,0,85,0,1,85,0	3480	00 00 00	.BYTE	0,0,0,0,0,0,0,0
32D0	01 57 00	.BYTE	1,87,0,1,92,0,1,80	3488	00 00 00	.BYTE	0,0,0,0,0,0,0,0
32D8	00 05 70	.BYTE	0,5,112,0,5,112,0,5	3490	00 00 00	.BYTE	0,0,0,0,0,0,0,0
32E0	70 00 15	.BYTE	112,0,21,192,0,21,1	3498	00 00 00	.BYTE	0,0,0,0,0,0,0,0
			92,0	34A0	00 00 00	.BYTE	0,0,0,0,0,0,0,0
32E8	15 C0 00	.BYTE	21,192,0,21,192,0,2	34A8	00 00 00	.BYTE	0,0,0,0,0,0,0,0
			3,0	34B0	00 00 00	.BYTE	0,0,0,0,0,0,0,0
32F0	00 57 00	.BYTE	0,87,0,0,87,0,0,171	34B8	00 00 00	.BYTE	0,0,0,0,0,0,0,2
32F8	0F 00 FF	.BYTE	15,0,255,255,0,0,0,0	34C0	00 0A 80	.BYTE	0,10,128,0,14,128,0,
3300	57 00 00	.BYTE	87,0,0,124,0,0,112,0				7
3308	00 C0 00	.BYTE	0,192,0,0,0,0,0,0	34C8	97 00 05	.BYTE	151,0,5,213,0,21,85,
3310	00 00 00	.BYTE	0,0,0,0,0,0,0,0				0
3318	00 00 00	.BYTE	0,0,0,0,0,0,0,0	34D0	55 55 00	.BYTE	85,85,0,85,92,1,93,
3320	00 00 00	.BYTE	0,0,0,0,0,0,0,0				92
3328	00 00 00	.BYTE	0,0,0,0,0,0,0,0	34D8	03 C5 5C	.BYTE	3,197,92,13,117,92,
3330	00 00 00	.BYTE	0,0,0,0,15,252,0,255,2				13,95
			55	34E0	FF 35 55	.BYTE	255,53,85,85,53,213
3338	FF FC 0F	.BYTE	255,252,15,255,0,0,				85,53
			0,0	34E8	F5 55 35	.BYTE	245,85,53,197,85,18
3340	00 FC 00	.BYTE	0,252,0,1,85,0,3,255				1,192,0
3348	00 03 FA	.BYTE	0,3,250,0,3,251,0,3	34F0	B5 C0 FF	.BYTE	181,192,255,186,20
3350	BA 80 03	.BYTE	186,128,3,170,0,2,1				7,255,255,255
			70,0	34F8	FF 00 00	.BYTE	255,0,0,0,0,0,0,2
3358	02 A0 00	.BYTE	2,160,0,3,160,0,1,2	3500	00 00 00	.BYTE	0,0,0,0,0,0,0,0
			24	3508	00 C0 00	.BYTE	0,192,0,0,112,0,0,95
3360	00 05 75	.BYTE	0,5,117,85,5,85,85,5	3510	00 00 55	.BYTE	0,0,85,168,0,21,168,
3368	D5 FF 05	.BYTE	213,255,5,117,192,1				0
			95,192	3518	0F 00 00	.BYTE	15,0,0,53,192,0,213,
3370	03 55 6A	.BYTE	3,85,106,0,213,106,				112
			0,85	3520	00 55 50	.BYTE	0,85,80,0,85,92,0,80
3378	C0 00 55	.BYTE	192,0,85,192,0,255,	3528	57 00 40	.BYTE	87,0,64,21,192,0,21,
			192,0				160
3380	00 00 00	.BYTE	0,0,0,0,0,0,0,0	3530	FF 05 A8	.BYTE	255,5,168,255,1,252
3388	00 00 00	.BYTE	0,0,0,0,176,0,0,172				255,255
3390	00 00 80	.BYTE	0,0,176,0,0,176,0,0	3538	FC 00 00	.BYTE	252,0,0,0,0,0,0,2
3398	80 00 00	.BYTE	176,0,0,0,176,0,0,176,	3540	00 00 00	.BYTE	0,0,0,0,0,0,0,0
			0	3548	00 00 00	.BYTE	0,0,0,0,0,0,0,0
33A0	00 80 00	.BYTE	0,176,0,0,176,0,0,0	3550	00 00 00	.BYTE	0,0,0,0,0,0,0,0
33A8	00 00 00	.BYTE	0,0,0,0,0,240,0,0	3558	00 00 00	.BYTE	0,0,0,0,0,0,0,0
33B0	A0 00 00	.BYTE	160,0,0,128,0,0,32,0	3560	00 00 00	.BYTE	0,0,0,0,0,0,0,0
33B8	00 00 00	.BYTE	0,0,0,0,0,0,0,11	3568	00 00 00	.BYTE	0,0,0,0,0,0,0,0
33C0	00 15 7F	.BYTE	0,21,127,0,85,85,0,	3570	00 00 00	.BYTE	0,0,0,0,0,0,0,0
			85	3578	00 00 00	.BYTE	0,0,0,0,0,0,0,0
33C8	55 00 55	.BYTE	85,0,85,85,1,85,85,1	3580	00 00 00	.BYTE	0,0,0,0,0,0,0,0
33D0	55 FF 01	.BYTE	85,255,1,95,0,1,92,0	3588	00 00 00	.BYTE	0,0,0,0,0,0,0,0
33D8	01 70 00	.BYTE	1,112,0,1,112,0,1,11	3590	00 00 00	.BYTE	0,0,0,0,0,0,0,0
			2	3598	00 00 00	.BYTE	0,0,0,0,0,0,0,0
33E0	00 01 70	.BYTE	0,1,112,0,1,112,0,1	35A0	00 00 00	.BYTE	0,0,0,0,0,0,0,0
33E8	70 00 01	.BYTE	112,0,1,112,0,1,112,	35A8	00 00 00	.BYTE	0,0,0,0,0,0,0,0
			0	35B0	00 00 00	.BYTE	0,0,0,0,0,0,0,0
33F0	01 70 0F	.BYTE	1,112,15,2,179,255,	35B8	00 00 00	.BYTE	0,0,0,0,0,0,0,0
			3,255	35C0	00 00 00	.BYTE	0,0,0,0,0,0,0,0
33F8	F0 00 00	.BYTE	240,0,0,0,0,0,0,15	35C8	00 00 00	.BYTE	0,0,0,0,0,0,0,0
3400	00 00 00	.BYTE	0,0,0,0,0,0,80,0	35D0	00 00 00	.BYTE	0,0,0,0,0,0,0,0
3408	00 50 00	.BYTE	0,80,0,0,80,0,0,80	35D8	00 00 00	.BYTE	0,0,0,0,128,0,30,160
3410	00 00 50	.BYTE	0,0,80,0,0,80,0,0	35E0	00 DE A1	.BYTE	0,222,161,85,222,17
3418	50 00 00	.BYTE	80,0,0,160,0,0,32,0				1,85,223
3420	00 20 00	.BYTE	0,32,0,0,0,0,0,0	35E8	EA D5 DF	.BYTE	234,213,223,171,95,
3428	00 00 00	.BYTE	0,0,0,0,0,0,0,0				223,237,85
3430	00 00 00	.BYTE	0,0,0,192,0,0,240,0	35F0	FF FF 55	.BYTE	255,255,85,0,3,213,
3438	00 00 00	.BYTE	0,0,0,0,0,0,0,0				0,0
3440	00 00 00	.BYTE	0,0,0,0,0,0,0,0	35F8	FF 00 00	.BYTE	255,0,0,0,0,0,0,0
3448	00 00 00	.BYTE	0,0,0,0,0,0,0,0	3600	00 00 00	.BYTE	0,0,0,0,0,0,0,0
3450	00 00 00	.BYTE	0,0,0,0,0,0,0,0	3608	00 00 00	.BYTE	0,0,0,0,0,0,0,0
3458	00 00 00	.BYTE	0,0,0,0,0,0,0,0	3610	00 00 00	.BYTE	0,0,0,0,0,0,0,0

Chapter 3

3618	00 15 C0	.BYTE 0,21,192,0,85,112,1,85	3778	C0 00 00	.BYTE 192,0,0,0,0,0,0,15
3620	70 D5 5D	.BYTE 112,213,93,112,213,113,112,215	3780	00 00 00	.BYTE 0,0,0,0,0,0,0
3628	C1 70 D5	.BYTE 193,112,213,127,112,211,217,85,242	3788	00 00 00	.BYTE 0,0,0,0,0,0,0
3630	5D 55 5E	.BYTE 93,85,94,87,245,90,214,173	3790	00 00 00	.BYTE 0,0,0,0,0,0,2
3638	5A F6 AF	.BYTE 90,246,175,255,63,240,0,0	3798	00 00 00	.BYTE 0,0,0,0,0,0,0
	; black player sprites		37A0	00 00 00	.BYTE 0,0,0,0,2,129,85,2
3640	00 00 00	.BYTE 0,0,0,0,0,0,0,0	37A8	AB FF 02	.BYTE 171,255,2,171,255,0,1,95
3648	00 00 00	.BYTE 0,0,0,0,0,0,0,0	37B0	00 00 07	.BYTE 0,0,7,0,0,1,0,0
3650	00 00 00	.BYTE 0,0,0,0,0,0,0,0	37B8	00 00 00	.BYTE 0,0,0,0,0,0,0,0
3658	00 00 00	.BYTE 0,0,0,0,0,0,0,0	37C0	00 00 00	.BYTE 0,0,0,0,0,63,0
3660	B0 00 02	.BYTE 176,0,2,176,0,0,172,0	37C8	00 55 40	.BYTE 0,85,64,0,255,192,0,175
3668	00 2B 00	.BYTE 0,43,0,0,11,0,0,3	37D0	C0 00 EF	.BYTE 192,0,239,192,0,174,192,0
3670	00 00 03	.BYTE 0,0,3,0,0,0,0,0	37D8	AA C0 00	.BYTE 170,192,0,170,128,0,10,128
3678	00 00 00	.BYTE 0,0,0,0,0,0,0,11	37E0	00 0A 40	.BYTE 0,10,64,0,121,240,0,247
3680	03 F0 00	.BYTE 3,240,0,5,84,0,15,252	37E8	FC 00 FF	.BYTE 252,0,255,223,0,255,223,0
3688	00 0A FC	.BYTE 0,10,252,0,14,252,0,42	37F0	FF 7F 00	.BYTE 255,127,0,234,255,0,234,244
3690	EC 00 0A	.BYTE 236,0,10,172,0,10,168,0	37F8	00 3F 50	.BYTE 0,63,80,0,63,240,0,0
3698	00 A8 00	.BYTE 0,168,0,0,164,0,7,159	3800	00 00 00	.BYTE 0,0,0,0,0,0,0,0
36A0	C0 1F 7F	.BYTE 192,31,127,192,127,247,240,255	3808	00 00 00	.BYTE 0,0,0,0,0,0,0,0
36A8	DF F0 AF	.BYTE 223,240,175,127,240,170,253,192	3810	00 02 00	.BYTE 0,2,0,0,10,0,42
36B0	2A 77 C0	.BYTE 42,119,192,3,223,192,3,255	3818	00 0A AA	.BYTE 0,0,170,0,0,171,0,0
36B8	00 03 FF	.BYTE 0,3,255,0,1,87,0,15	3820	AC 00 00	.BYTE 172,0,0,172,0,0,172,0
36C0	00 00 00	.BYTE 0,0,0,0,0,0,0,0	3828	00 AC 00	.BYTE 0,172,0,0,172,0,172,0
36C8	00 00 00	.BYTE 0,0,0,0,0,0,0,0	3830	00 00 AC	.BYTE 0,0,172,0,0,172,0,1
36D0	00 00 00	.BYTE 0,0,0,0,0,0,0,0	3838	5F 00 03	.BYTE 95,0,3,255,0,0,0,0
36D8	00 00 00	.BYTE 0,0,0,0,0,2,0,0	3840	2A B0 00	.BYTE 42,176,0,21,112,0,4,2,176
36E0	02 00 00	.BYTE 2,0,0,2,0,0,2,0	3848	00 2A B0	.BYTE 0,42,176,0,42,176,0,170
36E8	00 02 00	.BYTE 0,2,0,0,2,0,0,2	3850	B0 00 AA	.BYTE 176,0,170,176,0,170,176,0
36F0	00 00 02	.BYTE 0,0,2,0,0,5,0,0	3858	BA B0 00	.BYTE 188,176,0,202,176,0,10,176
36F8	0F 00 02	.BYTE 15,0,0,0,0,0,0,15	3860	00 0A AC	.BYTE 0,10,172,0,2,171,0,0
3700	02 AB 00	.BYTE 2,171,0,10,170,192,10,170	3868	AA C0 00	.BYTE 170,192,0,42,176,0,10,172
3708	C0 2A AA	.BYTE 192,42,170,176,42,170,176,42	3870	F0 02 A8	.BYTE 240,2,168,255,2,92,255,245
3710	AA B0 AA	.BYTE 170,176,170,250,172,171,2,172	3878	7C 00 0F	.BYTE 124,0,15,252,0,0,0,0
3718	AB 02 AC	.BYTE 171,2,172,172,0,171,172,0	3880	00 00 00	.BYTE 0,0,0,0,0,0,0,0
3720	AB AC 00	.BYTE 171,172,0,171,176,0,43,176	3888	00 00 00	.BYTE 0,0,0,0,0,0,0,0
3728	00 2B B0	.BYTE 0,43,176,0,43,176,240,43	3890	0A 00 00	.BYTE 10,0,0,10,192,0,2,176
3730	B3 FC 2B	.BYTE 179,252,43,127,255,23,252,3	3898	00 00 AD	.BYTE 0,0,173,0,0,43,0,0
3738	FF 00 00	.BYTE 255,0,0,0,0,0,0,15	38A0	0B 00 00	.BYTE 11,0,0,1,27,192,0,26
3740	02 AB 00	.BYTE 2,171,0,10,170,192,10,170	38A8	BC 00 1A	.BYTE 188,0,26,171,0,10,170,240
3748	C0 2A AA	.BYTE 192,42,170,176,42,170,176,42	38B0	0A AA AC	.BYTE 10,170,172,0,170,170,10,10
3750	AA B0 AA	.BYTE 170,176,170,250,176,171,2,176	38B8	AA 00 02	.BYTE 170,0,2,170,0,0,170,0
3758	AB 02 B0	.BYTE 171,2,176,172,2,176,172,2	38C0	03 F0 00	.BYTE 3,240,0,5,84,0,15,252
3760	B0 AC 02	.BYTE 176,172,2,176,176,1,0,176,176	38C8	00 0A FC	.BYTE 0,10,252,0,14,252,0,42
3768	0A B0 B0	.BYTE 10,176,176,42,192,191,26,192	38D0	EC 00 0A	.BYTE 236,0,10,172,0,10,168,0
3770	BF C6 C0	.BYTE 191,198,192,127,253,192,240,15	38D8	00 A8 00	.BYTE 0,168,0,0,168,0,85,169
			38E0	00 FD A7	.BYTE 0,253,167,192,255,9,5,192,95
			38E8	FF F0 0F	.BYTE 255,240,15,255,240,15,255,252

The Source Code

38F0	3F FF FC	.BYTE	63,255,252,127,255, 124,159,250	3A78	F0 00 00	.BYTE	240,0,0,0,0,0,15
38F8	FC A7 FA	.BYTE	252,167,250,244,16 9,252,80,0	3A80	00 00 00	.BYTE	0,0,0,0,0,0,0
3900	00 00 2A	.BYTE	0,0,42,0,0,10,0,0	3A88	00 00 00	.BYTE	0,0,0,0,0,0,0
3908	02 00 00	.BYTE	2,0,0,0,0,0,0,0	3A90	00 00 00	.BYTE	0,0,0,0,0,0,0
3910	00 00 00	.BYTE	0,0,0,0,0,0,0,0	3A98	00 00 00	.BYTE	0,0,0,0,0,0,0
3918	00 00 00	.BYTE	0,0,0,0,0,0,0,0	3AA0	00 00 00	.BYTE	0,0,0,0,0,0,0
3920	00 00 00	.BYTE	0,0,0,0,0,0,0,0	3AA8	00 00 00	.BYTE	0,0,0,0,0,0,0
3928	00 00 00	.BYTE	0,0,0,0,0,0,0,0	3AB0	00 00 00	.BYTE	0,0,0,0,0,0,0
3930	00 00 00	.BYTE	0,0,0,0,15,252,255,2 55	3AB8	00 00 00	.BYTE	0,0,0,0,0,0,11
3938	FF FF C0	.BYTE	255,255,192,63,0,0, 0,0	3AC0	00 00 00	.BYTE	0,0,0,0,0,0,0
3940	AA C0 00	.BYTE	170,192,0,170,192,0 170,192	3AC8	00 00 00	.BYTE	0,0,0,0,0,0,0
3948	00 AA C0	.BYTE	0,170,192,0,170,176 0,42	3AD0	00 00 00	.BYTE	0,0,0,0,0,0,0
3950	B0 00 0A	.BYTE	176,0,10,176,0,10,1 76,0	3AD8	00 00 00	.BYTE	0,0,0,0,0,0,0
3958	02 AC 00	.BYTE	2,172,0,2,172,0,2,1 72	3AE0	00 00 00	.BYTE	0,0,0,0,0,0,15
3960	00 00 AB	.BYTE	0,0,171,0,0,171,0,0	3AE8	C0 00 15	.BYTE	192,0,21,80,0,63,24 0,0
3968	AB 00 00	.BYTE	171,0,0,171,0,0,43,0	3AF0	2B F0 00	.BYTE	43,240,0,59,240,0,1 71,176
3970	00 2A 00	.BYTE	0,42,0,0,42,0,255,2 13	3AF8	00 2A B0	.BYTE	0,42,176,0,42,160,0, 2
3978	00 FF FF	.BYTE	0,255,255,0,0,0,0,0	3B00	00 00 00	.BYTE	0,0,0,0,0,0,0
3980	00 00 00	.BYTE	0,0,0,0,0,0,0,0	3B08	00 00 00	.BYTE	0,0,0,1,0,0,7,0
3988	00 00 00	.BYTE	0,0,0,0,0,0,11,0	3B10	28 0F 00	.BYTE	40,15,0,42,173,0,42, 164
3990	00 2B 00	.BYTE	0,43,0,0,11,0,0,11	3B18	00 00 F0	.BYTE	0,0,240,0,0,172,0,2
3998	00 00 0B	.BYTE	0,0,11,0,0,11,0,0	3B20	AB 00 0A	.BYTE	171,0,10,170,0,10,1 70,0
39A0	0B 00 00	.BYTE	11,0,0,11,0,0,11,0	3B28	2A FA 00	.BYTE	42,250,0,171,2,5,17 1,0
39A8	00 00 00	.BYTE	0,0,0,0,0,0,15	3B30	15 AC 0F	.BYTE	21,172,15,63,176,25 5,63,255
39B0	00 00 0A	.BYTE	0,0,10,0,2,0,0	3B38	FF 00 00	.BYTE	255,0,0,0,0,0,0,2
39B8	08 00 00	.BYTE	8,0,0,0,0,0,0,11	3B40	02 A0 00	.BYTE	2,160,0,2,144,0,94,1 12
39C0	00 3F 00	.BYTE	0,63,0,0,85,64,0,255	3B48	00 FD F4	.BYTE	0,253,244,0,255,253 0,95
39C8	C0 00 AF	.BYTE	192,0,175,192,0,239 192,2	3B50	FF 40 0F	.BYTE	255,64,15,255,192,1 5,247,28
39D0	AE C0 00	.BYTE	174,192,0,170,192,0 170,128	3B58	0F F1 1C	.BYTE	15,241,28,15,254,17 6,245,122
39D8	00 0A 80	.BYTE	0,10,128,0,10,64,85, 89	3B60	B4 AA AA	.BYTE	180,170,170,172,17 0,171,172,170
39E0	F0 FF F7	.BYTE	240,255,247,240,25 5,255,240,85	3B68	AF AC AA	.BYTE	175,172,170,163,17 2,0,3,174
39E8	FD F0 00	.BYTE	253,240,0,247,240,5 3,95,240	3B70	FF F3 AE	.BYTE	255,243,174,255,25 5,94,255,255
39F0	AF FF 70	.BYTE	175,255,112,175,25 5,240,5,85	3B78	FF 00 00	.BYTE	255,0,0,0,0,0,0,2
39F8	C0 03 FF	.BYTE	192,3,255,192,1,85, 192,0	3B80	00 00 00	.BYTE	0,0,0,0,0,0,0
3A00	00 00 00	.BYTE	0,0,0,0,0,0,0,0	3B88	00 00 00	.BYTE	0,0,0,0,0,0,0
3A08	0A 00 00	.BYTE	10,0,0,10,0,0,10,0	3B90	00 00 00	.BYTE	0,0,0,0,0,0,0
3A10	00 0A 00	.BYTE	0,10,0,0,10,0,0,10	3B98	00 00 00	.BYTE	0,0,0,0,0,0,0
3A18	00 00 0A	.BYTE	0,0,10,0,0,5,0,0	3BA0	00 00 00	.BYTE	0,0,0,0,0,0,0
3A20	04 00 00	.BYTE	4,0,0,4,0,0,0,0	3BA8	00 00 00	.BYTE	0,0,0,0,0,0,0
3A28	00 00 00	.BYTE	0,0,0,0,0,0,0,0	3BB0	00 00 00	.BYTE	0,0,0,0,0,0,0
3A30	00 00 00	.BYTE	0,0,0,0,0,0,3,0,0	3BB8	00 00 00	.BYTE	0,0,0,0,0,0,11
3A38	0F 00 00	.BYTE	15,0,0,0,0,0,0,0	3BC0	00 00 00	.BYTE	0,0,0,0,0,0,0
3A40	FE AB 00	.BYTE	254,171,0,170,170,1 92,170,170	3BC8	00 00 00	.BYTE	0,0,0,0,0,0,0
3A48	C0 AA AA	.BYTE	192,170,170,192,17 0,170,176,192	3BD0	00 00 00	.BYTE	0,0,0,0,0,0,0
3A50	AA B0 C0	.BYTE	170,176,192,10,176, 192,10,176	3BD8	00 00 00	.BYTE	0,0,0,0,0,0,0
3A58	C0 02 B0	.BYTE	192,2,176,192,2,176 0,2	3BE0	00 00 00	.BYTE	0,0,0,0,0,0,0
3A60	B0 00 02	.BYTE	176,0,2,176,0,2,176, 0	3BE8	00 00 00	.BYTE	0,0,0,0,0,0,0
3A68	02 B0 00	.BYTE	2,176,0,2,176,0,2,1 76	3BF0	00 00 00	.BYTE	0,0,0,0,0,0,0
3A70	3F C2 B0	.BYTE	63,194,176,255,253, 112,192,255	3BF8	00 00 00	.BYTE	0,0,0,0,0,0,0
				3C00	00 00 00	.BYTE	0,0,0,0,0,0,0
				3C08	00 00 00	.BYTE	0,0,0,0,0,0,0
				3C10	00 00 00	.BYTE	0,0,0,0,0,0,44,0
				3C18	00 AB 00	.BYTE	0,171,0,2,170,192,2, 170
				3C20	BF 02 BA	.BYTE	191,2,186,169,2,142 1,169,2
				3C28	83 E9 02	.BYTE	131,233,2,254,169,7 9,170,165

Chapter 3

```

3C30 7A AA 9A          .BYTE 122,170,154,90,175,
                        234,90,186
3C38 A5 FF FA          .BYTE 165,255,250,175,0,1
                        5,252,0
3C40 00 00 00          .BYTE 0,0,0,0,0,0,0
3C48 00 00 00          .BYTE 0,0,0,0,0,0,0
3C50 00 00 00          .BYTE 0,0,0,0,0,0,0
3C58 00 00 00          .BYTE 0,0,0,0,2,0,0,10
3C60 B4 FF CA          .BYTE 180,255,202,183,25
                        5,106,183,253
3C68 AB F7 5F          .BYTE 171,247,95,106,247,
                        255,219,247
3C70 FF FF FF          .BYTE 255,255,255,255,19
                        2,0,255,0
3C78 00 00 00          .BYTE 0,0,0,0,0,0,0,0
;
3C80 EA                NOP
3C81 EA                NOP
;
;*=15490
;
3C82          TESTBT1  = $FB
3C82          TESTBT2  = $FC
3C82          SCOR1    = 828
3C82          SCOR2    = 830
3C82          LINPRT   = $BDCD
3C82          V        = 53248
3C82          S        = 54272
3C82          JY1      = 56321
3C82          JY2      = 56320
3C82          PLOT     = $FFFO
3C82          CHROUT   = $FFD2
;
3C82 20 A0 40 BEGINING JSR  SCREEN
3C85 20 93 40          JSR  SID
3C88 A9 00            LDA  #0
3C8A 8D 3C 03        STA  828
3C8D 8D 3D 03        STA  829
3C90 8D 3E 03        STA  830
3C93 8D 3F 03        STA  831
;
;set up sprites
;
3C96 A9 FF          START  LDA  #255
3C98 8D 15 D0      STA  V+21
3C9B 8D 1C D0      STA  V+28
3C9E A9 01          LDA  #1
3CA0 8D 25 D0      STA  V+37
3CA3 A9 00          LDA  #0
3CA5 8D 26 D0      STA  V+38
3CA8 A2 00          LDX  #0
3CAA A9 0A          SPRCOL LDA  #10
3CAC 9D 27 D0      STA  V+39,X
3CAF E8            INX
3CB0 E0 08          CPX  #8
3CB2 D0 F6          BNE  SPRCOL
;
;starting positions
;
3CB4 A9 C0          DEFAULT LDA  #192
3CB6 8D F8 07      STA  2040
3CB9 A9 C1          LDA  #193
3CBB 8D F9 07      STA  2041
3CBE A9 C2          LDA  #194
3CC0 8D FA 07      STA  2042
3CC3 A9 C3          LDA  #195
3CC5 8D FB 07      STA  2043
3CC8 A9 D9          LDA  #217
3CCA 8D FC 07      STA  2044
3CCD A9 DA          LDA  #218
3CCF 8D FD 07      STA  2045
3CD2 A9 DB          LDA  #219
;
3CD4 8D FE 07      STA  2046
3CD7 A9 DC          LDA  #220
3CD9 8D FF 07      STA  2047
3CDC A9 B4          LDA  #180
3CDE 8D 05 D0      STA  V+5
3CE1 8D 07 D0      STA  V+7
3CE4 8D 0D D0      STA  V+13
3CE7 8D 0F D0      STA  V+15
3CEA A9 A0          LDA  #160
3CEC 8D 01 D0      STA  V+1
3CEF 8D 03 D0      STA  V+3
3CF2 8D 09 D0      STA  V+9
3CF5 8D 0B D0      STA  V+11
3CF8 A9 51          LDA  #81
3CFA 8D 00 D0      STA  V
3CFD 8D 04 D0      STA  V+4
3D00 A9 69          LDA  #105
3D02 8D 02 D0      STA  V+2
3D05 8D 06 D0      STA  V+6
3D08 A9 B4          LDA  #180
3D0A 8D 08 D0      STA  V+8
3D0D 8D 0C D0      STA  V+12
3D10 A9 CC          LDA  #204
3D12 8D 0A D0      STA  V+10
3D15 8D 0E D0      STA  V+14
;
;main loop
;
3D18 AD 00 DC JOY2   LDA  JY2
3D1B 29 10          AND  #16
3D1D D0 35          BNE  STIC2
3D1F A5 FC          LDA  TESTBT2
3D21 D0 31          BNE  STIC2
3D23 AD 00 DC          LDA  JY2
3D26 29 0F          AND  #15
3D28 C9 0A          CMP  #10
3D2A D0 08          BNE  LOKIK2
3D2C A9 E2          LDA  #226
3D2E 20 CE 3E       JSR  GEN2
3D31 4C BF 3D       JMP  JOY1
3D34 C9 09          CMP  #9
3D36 D0 08          BNE  PUNCH2
3D38 A9 EA          LDA  #234
3D3A 20 CE 3E       JSR  GEN2
3D3D 4C BF 3D       JMP  JOY1
3D40 C9 0B          CMP  #11
3D42 D0 08          BNE  BLOCK2
3D44 A9 DE          LDA  #222
3D46 20 CE 3E       JSR  GEN2
3D49 4C BF 3D       JMP  JOY1
3D4C A9 E6          LDA  #230
3D4E 20 CE 3E       JSR  GEN2
3D51 4C BF 3D       JMP  JOY1
;
;move black
;
3D54 AD 00 DC STIC2  LDA  JY2
3D57 29 10          AND  #16
3D59 F0 04          BEQ  FOB3
3D5B A9 00          LDA  #0
3D5D 85 FC          STA  TESTBT2
3D5F AD 00 DC FOB3  LDA  JY2
3D62 29 0F          AND  #15
3D64 C9 0F          CMP  #15
3D66 D0 03          BNE  FOB4
3D68 4C BF 3D       JMP  JOY1
3D6B 38            SEC
3D6C C9 08          CMP  #8
3D6E 90 30          BCC  BACKW2
3D70 AD 0E D0      LDA  V+14
3D73 18            CLC
3D74 C9 28          CMP  #40

```

The Source Code

```

3076 B0 03          BCS  FORWD2
3078 4C BF 3D      JMP  JOY1
307B AD 08 00      LDA  V+8
307E 38            SEC
307F ED 00 00      SBC  V
3082 C9 14          CMP  #20
3084 B0 03          BCS  FORL2
3086 4C BF 3D      JMP  JOY1
3089 A2 05          LDX  #5
308B CE 08 00      DEC  V+8
308E CE 0A 00      DEC  V+10
3091 CE 0C 00      DEC  V+12
3094 CE 0E 00      DEC  V+14
3097 CA            DEX
3098 D0 F1          BNE  FAL2
309A EE FF 07      INC  2047
309D 4C BF 3D      JMP  JOY1
30A0 AD 0E 00      LDA  V+14
30A3 38            SEC
30A4 C9 EB          CMP  #235
30A6 90 03          BCC  BACKWD2
30A8 4C BF 3D      JMP  JOY1
30AB A2 05          LDX  #5
30AD EE 08 00      INC  V+8
30B0 EE 0A 00      INC  V+10
30B3 EE 0C 00      INC  V+12
30B6 EE 0E 00      INC  V+14
30B9 CA            DEX
30BA D0 F1          BNE  BAL2
30BC EE FF 07      INC  2047
;
30BF AD 01 00      LDA  JY1
30C2 29 10          AND  #16
30C4 D0 35          BNE  STICK
30C6 A5 FB          LDA  TESTBT1
30C8 D0 31          BNE  STICK
30CA AD 01 00      LDA  JY1
30CD 29 0F          AND  #15
30CF C9 06          CMP  #6
30D1 D0 08          BNE  LOKIK
30D3 A9 C9          LDA  #201
30D5 20 AE 3E      JSR  GEN1
30D8 4C 6C 3E      JMP  ENLP
30DB C9 07          CMP  #7
30DD D0 08          BNE  PUNCH
30DF A9 C5          LDA  #197
30E1 20 AE 3E      JSR  GEN1
30E4 4C 6C 3E      JMP  ENLP
30E7 C9 05          CMP  #5
30E9 D0 08          BNE  BLOCK
30EB A9 D1          LDA  #209
30ED 20 AE 3E      JSR  GEN1
30F0 4C 6C 3E      JMP  ENLP
30F3 A9 CD          LDA  #205
30F5 20 AE 3E      JSR  GEN1
30F8 4C 6C 3E      JMP  ENLP
;
;move white
30FB AD 01 00      LDA  JY1
30FE 29 10          AND  #16
3E00 F0 04          BEQ  FOB1
3E02 A9 00          LDA  #0
3E04 85 FB          STA  TESTBT1
3E06 AD 01 00      LDA  JY1
3E09 29 0F          AND  #15
3E0B C9 0F          CMP  #15
3E0D D0 03          BNE  FOB2
3E0F 4C 6C 3E      JMP  ENLP
3E12 38            SEC
3E13 C9 08          CMP  #8
3E15 90 25          BCC  FORW

3E17 AD 00 00      LDA  V
3E1A 18            CLC
3E1B C9 2D          CMP  #45
3E1D B0 03          BCS  BACKWD
3E1F 4C 6C 3E      JMP  ENLP
3E22 A2 05          LDX  #5
3E24 CE 00 00      DEC  V
3E27 CE 02 00      DEC  V+2
3E2A CE 04 00      DEC  V+4
3E2D CE 06 00      DEC  V+6
3E30 CA            DEX
3E31 D0 F1          BNE  BAL
3E33 EE FA 07      INC  2042
3E36 EE FA 07      INC  2042
3E39 4C 6C 3E      JMP  ENLP
3E3C AD 08 00      LDA  V+8
3E3F 38            SEC
3E40 ED 00 00      SBC  V
3E43 C9 14          CMP  #20
3E45 B0 03          BCS  FORL
3E47 4C 6C 3E      JMP  ENLP
3E4A AD 00 00      LDA  V
3E4D 38            SEC
3E4E C9 DC          CMP  #220
3E50 90 03          BCC  FORWD
3E52 4C 6C 3E      JMP  ENLP
3E55 A2 05          LDX  #5
3E57 EE 00 00      INC  V
3E5A EE 02 00      INC  V+2
3E5D EE 04 00      INC  V+4
3E60 EE 06 00      INC  V+6
3E63 CA            DEX
3E64 D0 F1          BNE  FAL
3E66 EE FA 07      INC  2042
3E69 EE FA 07      INC  2042
;
;collision checks
3E6C A2 64          ENLP
3E6E 20 BA 3F      LDX  #100
3E71 AD 1E 00      JSR  TIME
3E74 AA            LDA  V+30
3E75 29 12          AND  #18
3E77 C9 12          CMP  #18
3E79 F0 73          BEQ  COLL
3E7B 8A            TXA
3E7C 29 48          EN
3E7E C9 48          AND  #72
3E7F C9 48          CMP  #72
3E80 F0 6C          BEQ  COLL
3E82 8A            TXA
3E83 29 18          AND  #24
3E85 C9 18          CMP  #24
3E87 F0 65          BEQ  COLL
3E89 8A            TXA
3E8A 29 42          AND  #66
3E8C C9 42          CMP  #66
3E8E F0 5E          BEQ  COLL
3E90 A5 FB          ENO
3E92 A8            LDA  TESTBT1
3E93 A9 CO          TXA
3E95 20 AE 3E      LDA  #192
3E98 98            JSR  GEN1
3E99 85 FB          TXA
3E9B A5 FC          STA  TESTBT1
3E9D A8            LDA  TESTBT2
3E9E A9 D9          TXA
3EAO 20 CE 3E      LDA  #217
3EA3 98            JSR  GEN2
3EA4 85 FC          TXA
3EA6 A2 32          STA  TESTBT2
3EA8 20 BA 3F      LDX  #50
3EAB 4C 18 3D      JSR  TIME
3EAC 4C 18 3D      JMP  JOY2

```

Chapter 3

```

;animation subroutines
3EAE A2 00 GEN1 LDX #0
3EB0 9D F8 07 GN1 STA 2040,X
3EB3 18 CLC
3EB4 69 01 ADC #1
3EB6 E8 INX
3EB7 E0 04 CPX #4
3EB9 D0 F5 BNE GN1
3EBB A9 01 LDA #1
3EBD 85 FB STA TESTBT1
3EBF AD F8 07 LDA 2040
3EC2 C9 C0 CMP #192
3EC4 F0 07 BEQ NOSND
3EC6 C9 D5 CMP #213
3EC8 F0 03 BEQ NOSND
3ECA 20 6A 40 JSR SWISH
3ECD 60 NOSND RTS

;
3ECE A2 00 GEN2 LDX #0
3ED0 9D FC 07 GN2 STA 2044,X
3ED3 18 CLC
3ED4 69 01 ADC #1
3ED6 E8 INX
3ED7 E0 04 CPX #4
3ED9 D0 F5 BNE GN2
3EDB A9 01 LDA #1
3EDD 85 FC STA TESTBT2
3EDF AD FC 07 LDA 2044
3EE2 C9 EE CMP #238
3EE4 F0 E7 BEQ NOSND
3EE6 C9 D9 CMP #217
3EE8 F0 E3 BEQ NOSND
3EEA 20 6A 40 JSR SWISH
3EED 60 RTS

;
;collision routine
3EEE AD F8 07 COLL LDA 2040
3EF1 C9 CD CMP #205
3EF3 F0 70 BEQ SC4
3EF5 AD FC 07 LDA 2044
3EF8 C9 E6 CMP #230
3EFA F0 69 BEQ SC4
3EFC AD F8 07 LDA 2040
3EFF C9 C0 CMP #192
3F01 D0 51 BNE SC2
3F03 AD FC 07 LDA 2044
3F06 C9 D9 CMP #217
3F08 F0 4A BEQ SC2
3FOA 20 44 40 JSR WACK
3F0D A2 0A LDX #10
3F0F CE 00 D0 BALC LDY V
3F12 CE 02 D0 DEC V+2
3F15 CE 04 D0 DEC V+4
3F18 CE 06 D0 DEC V+6
3F1B CA DEX
3F1C D0 F1 BNE BALC
3F1E EE FA 07 INC 2042
3F21 EE FA 07 INC 2042
3F24 A9 D5 LDA #213
3F26 20 AE 3E JSR GEN1
3F29 A9 D9 LDA #217
3F2B 20 CE 3E JSR GEN2
3F2E 20 10 40 JSR BLACKSCORE
3F31 A2 FF LDX #255
3F33 20 BA 3F JSR TIME
3F36 A2 FF LDX #255
3F38 20 BA 3F JSR TIME
3F3B A2 FF LDX #255
3F3D 20 BA 3F JSR TIME

3F40 A9 03 LDA #3
3F42 CD 3F 03 CMP 831
3F45 F0 03 BEQ BLWNS
3F47 4C B4 3C JDF JMP DEFAULT
3F4A A9 E8 BLWNS LDA #232
3F4C CD 3E 03 CMP 830
3F4F D0 F6 BNE JDF
3F51 4C E5 3F JMP BLACKWINS
3F54 AD FC 07 SC2 LDA 2044
3F57 C9 D9 CMP #217
3F59 D0 D0 BNE SC3
3F5B AD F8 07 LDA 2040
3F5E C9 C0 CMP #192
3F60 D0 06 BNE SC3
3F62 4C 90 3E JMP EN0
3F65 4C 90 3E SC4 JMP EN0
3F68 AD FC 07 SC3 LDA 2044
3F6B C9 D9 CMP #217
3F6D D0 F6 BNE SC4
3F6F AD F8 07 LDA 2040
3F72 C9 C0 CMP #192
3F74 F0 EF BEQ SC4
3F76 20 44 40 JSR WACK
3F79 A2 0A LDX #10
3F7B EE 08 D0 BALC2 INC V+8
3F7E EE 0A D0 INC V+10
3F81 EE 0C D0 INC V+12
3F84 EE 0E D0 INC V+14
3F87 CA DEX
3F88 D0 F1 BNE BALC2
3F8A A9 EE LDA #238
3F8C 20 CE 3E JSR GEN2
3F8F A9 C0 LDA #192
3F91 20 AE 3E JSR GEN1
3F94 20 FC 3F JSR WHITESCORE
3F97 A2 FF LDX #255
3F99 20 BA 3F JSR TIME
3F9C A2 FF LDX #255
3F9E 20 BA 3F JSR TIME
3FA1 A2 FF LDX #255
3FA3 20 BA 3F JSR TIME
3FA6 A9 03 LDA #3
3FA8 CD 3D 03 CMP 829
3FAB F0 03 BEQ WHTWNS
3FAD 4C B4 3C JDF2 JMP DEFAULT
3FB0 A9 E8 WHTWNS LDA #232
3FB2 CD 3C 03 CMP 828
3FB5 D0 F6 BNE JDF2
3FB7 4C C3 3F JMP WHITWINS

;
;time delay subroutine
3FBA A0 FF LDY #255
3FBC 88 TIME DEY
3FBD D0 FD BNE LPY
3FBF CA DEX
3FC0 D0 F8 BNE TIME
3FC2 60 RTS

;
;game end routines
3FC3 A2 05 WHITWINS LDX #5
3FC5 A0 0B LDY #11
3FC7 18 CLC
3FC8 20 F0 FF JSR PLOT
3FCB A0 00 LDY #0
3CFD B9 9F 45 MSSG LDA LABEL2,Y
3FD0 C9 00 CMP #0
3FD2 F0 06 BEQ WAITS
3FD4 20 D2 FF JSR CHROUT
3FD7 C8 INY
3FD8 D0 F3 BNE MSSG

```


The Source Code

```

3FDA A9 00  WAITS      LDA  #0
3FDC 85 C6          STA  198
3FDE A5 C6  WT       LDA  198
3FE0 F0 FC          BEQ  WT
3FE2 4C 82 3C      JMP  BEGINING
3FE5 A2 05  BLACKWINS LDX  #5
3FE7 A0 0B          LDY  #11
3FE9 18            CLC
3FEA 20 F0 FF      JSR  PLOT
3FED A0 00          LDY  #0
3FEF B9 AC 45 BMSG   LDA  TABLE3,Y
3FF2 C9 00          CMP  #0
3FF4 F0 E4          BEQ  WAITS
3FF6 20 D2 FF      JSR  CHROUT
3FF9 C8            INY
3FFA D0 F3          BNE  BMSG
;
;scoring subroutines
;
3FFC 18  WHITESCOR   CLC
3FFD AD 3C 03      LDA  828
4000 69 64          ADC  #100
4002 8D 3C 03      STA  828
4005 AD 3D 03      LDA  829
4008 69 00          ADC  #0
400A 8D 3D 03      STA  829
400D 4C 21 40      JMP  SCORPLOT
4010 18  BLACKSCOR  CLC
4011 AD 3E 03      LDA  830
4014 69 64          ADC  #100
4016 8D 3E 03      STA  830
4019 AD 3F 03      LDA  831
401C 69 00          ADC  #0
401E 8D 3F 03      STA  831
4021 A2 08  SCORPLOT LDX  #8
4023 A0 22          LDY  #34
4025 18            CLC
4026 20 F0 FF      JSR  PLOT
4029 AD 3D 03      LDA  829
402C AE 3C 03      LDY  #28
402F 20 CD BD      JSR  LINPRT
4032 A2 0D          LDY  #13
4034 A0 22          LDY  #34
4036 18            CLC
4037 20 F0 FF      JSR  PLOT
403A AD 3F 03      LDA  831
403D AE 3E 03      LDY  #30
4040 20 CD BD      JSR  LINPRT
4043 60            RTS
;
;sound subroutines
;
4044 A9 0F  WACK     LDA  #15
4046 8D 18 D4       STA  S+24
4049 8D 0C D4       STA  S+12
404C A9 07          LDA  #7
404E 8D 0D D4       STA  S+13
4051 A9 1C          LDA  #28
4053 8D 08 D4       STA  S+8
4056 A9 00          LDA  #0
4058 8D 07 D4       STA  S+7
405B A9 81          LDA  #129
405D 8D 0B D4       STA  S+11
4060 A2 0A          LDY  #10
4062 20 BA 3F      JSR  TIME
4065 A9 80          LDA  #128
4067 8D 0B D4       STA  S+11
406A A9 E1  SWISH   LDA  #225
406C 8D 00 D4       STA  S
406F A9 1E          LDA  #30
4071 8D 01 D4       STA  S+1

4074 A9 44          LDA  #68
4076 8D 05 D4       STA  S+5
4079 A9 00          LDA  #0
407B 8D 06 D4       STA  S+6
407E A9 07          LDA  #7
4080 8D 18 D4       STA  S+24
4083 A9 81          LDA  #129
4085 8D 04 D4       STA  S+4
4088 A2 0A          LDY  #10
408A 20 BA 3F      JSR  TIME
408D A9 80          LDA  #128
408F 8D 04 D4       STA  S+4
4092 60            RTS
4093 A2 00  SID     LDY  #0
4095 A9 00          LDA  #0
4097 9D 00 D4 CLRSD STA  S,X
409A E8            INX
409B E0 18          CPY  #24
409D D0 F8          BNE  CLRSD
409F 60            RTS
;
;print background
;
40A0 A9 00  SCREEN  LDA  #0
40A2 8D 21 D0       STA  53281
40A5 A9 0F          LDA  #15
40A7 8D 20 D0       STA  53280
40AA A0 00          LDY  #0
40AC B9 DC 40 T1    JSR  TABLE,Y
40AF 20 D2 FF      JSR  CHROUT
40B2 C8            INY
40B3 D0 F7          BNE  T1
40B5 B9 DC 41 T2    LDA  TABLE+256,Y
40B8 20 D2 FF      JSR  CHROUT
40BB C8            INY
40BC D0 F7          BNE  T2
40BE B9 DC 42 T3    LDA  TABLE+512,Y
40C1 20 D2 FF      JSR  CHROUT
40C4 C8            INY
40C5 D0 F7          BNE  T3
40C7 B9 DC 43 T4    LDA  TABLE+768,Y
40CA 20 D2 FF      JSR  CHROUT
40CD C8            INY
40CE D0 F7          BNE  T4
40D0 B9 DC 44 T5    LDA  TABLE+1024,Y
40D3 F0 06          BEQ  QUIT
40D5 20 D2 FF      JSR  CHROUT
40D8 C8            INY
40D9 D0 F5          BNE  T5
40DB 60            RTS
40DC          =
40DC 93 20 12      .ASC  "(CLR) {RVS}{WH
T}{22 SPACE}"
40FE 20 20 20      .ASC  "{16 SPACE}{OFF
}"
4108 12 20 92      .ASC  "{RVS}{OFF}{
9 SPACE}{RVS}{<4
>}{11 SPACE}{OFF
}"
4123 20 20 20      .ASC  "{9 SPACE}{RVS}
{WHT}{OFF}{7 SP
ACE}{RVS}
"
4139 20 92 20      .ASC  "{OFF}{9 SPACE}
{RVS}<4>{11 SP
ACE}{OFF}"
4153 20 20 20      .ASC  "{9 SPACE}{RVS}
{OFF}{OFF}{7 SP
ACE}{RVS}
"
4169 20 92 20      .ASC  "{OFF}{9 SPACE}
<>{RVS}{9 SPA
CE}{OFF}&{2 SPA
CE}"

```

Chapter 3

4183	20 20 20	.ASC	"{8 SPACE}{RVS}{WHT}{OFF}{7 SPACE}{RVS}"	43C3	20 92 9E	.ASC	"{OFF}{YEL}N{RED}Q{RVS}{8 SPACE}{7 SPACE}"
4198	20 92 20	.ASC	"{OFF}{23 SPACE}"	43DD	20 20 20	.ASC	"{7 SPACE}{OFF}{RED}Q{YEL}M{RVS}{WHT}{OFF}{7 SPACE}{RVS}"
41B1	20 20 20	.ASC	"{6 SPACE}{SHIF T-SPACE}{RVS}{WHT}{OFF}{7 SPACE}{RVS}"	43F7	20 92 1C	.ASC	"{OFF}{RED}Q{RVS}{8 SPACE}{19 SPACE}"
41C5	20 92 20	.ASC	"{OFF}{30 SPACE}"	4411	20 20 20	.ASC	"{8 SPACE}{OFF}{RED}Q{RVS}{WHT}{OFF}{7 SPACE}{RVS}"
41E5	12 05 20	.ASC	"{RVS}{WHT}{OFF}{RVS}{WHITE}{OFF}{RVS}"	4429	20 92 12	.ASC	"{OFF}{RVS}{8 SPACE}{20 SPACE}"
41F4	20 92 20	.ASC	"{OFF}{24 SPACE}"	4442	20 20 20	.ASC	"{8 SPACE}{OFF}{RVS}{WHT}{OFF}{7 SPACE}{RVS}"
420E	20 20 20	.ASC	"{6 SPACE}{RVS}{OFF}{RVS}{SCORE}{OFF}{RVS}"	4459	20 9B 20	.ASC	"{8}{23 SPACE}"
4222	20 92 20	.ASC	"{OFF}{25 SPACE}"	4472	20 20 20	.ASC	"{7 SPACE}{WHT}{OFF}{7 SPACE}{RVS}"
423D	20 20 20	.ASC	"{6 SPACE}{RVS}{OFF}{7 SPACE}{RVS}"	4485	20 9B 20	.ASC	"{8}{23 SPACE}"
424D	20 20 92	.ASC	"{2 SPACE}{OFF}{3}"	449E	20 20 20	.ASC	"{7 SPACE}{WHT}{OFF}{7 SPACE}{RVS}"
4267	20 D1 20	.ASC	"Q Q Q Q{RVS}{WHT}{OFF}{2 SPACE}{000}{2 SPACE}{RVS}"	44B1	20 9B 20	.ASC	"{5}{22 SPACE}"
4296	20 D1 20	.ASC	"Q Q Q {RVS}{WHT}{OFF}{7 SPACE}{RVS}"	44C9	20 20 20	.ASC	"{8 SPACE}{WHT}{OFF}{7 SPACE}{RVS}"
42AA	20 92 20	.ASC	"{OFF}{3}"	44DD	20 92 20	.ASC	"{OFF}{4}{4}{7 SPACE}"
42C3	20 D1 20	.ASC	"Q Q Q Q{RVS}{WHT}{OFF}{RVS}{8}{BLACK}{OFF}{RVS}{WHT}"	44F7	20 20 20	.ASC	"{5 SPACE}{OFF}{RVS}{WHT}{OFF}{7 SPACE}{RVS}"
42DC	20 92 96	.ASC	"{OFF}{3}Q{RVS}{2}{OFF}{14 Y}"	450F	20 92 20	.ASC	"{OFF}{4}{RVS}{5 SPACE}{RING SIDE}{2 SPACE}{KARATE}"
42F6	B7 B7 B7	.ASC	"{6 Y}{2}{OFF}{3}Q{RVS}{WHT}{OFF}{RVS}{8}{SCORE}{OFF}{RVS}{OFF}"	452A	20 20 20	.ASC	"{5 SPACE}{OFF}{2 SPACE}{RVS}{WHT}{OFF}{7 SPACE}{RVS}"
4316	20 92 20	.ASC	"{OFF}{YEL}N{3}Q{RVS}{2}{0}Q Q Q{2}{RVS}{OFF}{3}Q{YEL}M{3}Q{RVS}{WHT}{OFF}{7 SPACE}{RVS}"	453F	20 92 20	.ASC	"{OFF}{4}{RVS}{5 SPACE}{OFF}{18 SPACE}"
4332	20 D1 20	.ASC	"Q Q Q Q{2}{RVS}{OFF}{3}Q{YEL}M{3}Q{RVS}{WHT}{OFF}{7 SPACE}{RVS}"	4559	20 20 20	.ASC	"{6 SPACE}{RVS}{OFF}{2 SPACE}{RVS}{WHT}{OFF}{7 SPACE}{RVS}"
434F	20 92 9E	.ASC	"{OFF}{YEL}N{3}Q{YEL}N{RVS}{2}{OFF}{YEL}{14 Y}"	4571	92 20 12	.ASC	"{OFF}{RVS}{WHT}{22 SPACE}"
436A	B7 B7 B7	.ASC	"{8 Y}{2}{RVS}{OFF}{YEL}M{3}Q{YEL}M{RVS}{WHT}{OFF}{2 SPACE}{000}{2 SPACE}{RVS}"	458B	20 20 20	.ASC	"{16 SPACE}{2 UP}{OFF}"
4389	20 92 96	.ASC	"{OFF}{3}Q{YEL}N{RED}Q{RVS}{2}{OFF}{RED}Q Q Q Q Q Q Q Q"	459E	00	.BYTE	0
43A4	D1 20 D1	.ASC	"Q Q Q Q Q Q Q Q{RVS}{OFF}{RED}Q{YEL}M{3}Q{RVS}{WHT}{OFF}{7 SPACE}{RVS}"	459F	9C 57 48	TABLE2	"{PUR}WHITE WIN SI"
				45AB	00	.BYTE	0
				45AC			"{PUR}BLACK WIN SI"
				454C	9C 42 4C	TABLE3	0
				4588	00	.BYTE	0

Eagles and Gators

Source code by Philip I. Nelson

```

0801          * = $0801
: assorted declarations
0801          PRTCOL      = $0286
0801          BUTN1      = $0367
0801          BUTN2      = $036A
0801          MYRAS1     = $4A88
0801          MYRAS2     = $4A89
0801          GATSCR     = $4A8A
0801          EAGSCR     = $4A8B
0801          RASTER     = $D012
0801          ENABLE     = $D015
0801          YEXPAN     = $D017
0801          VICIRQ     = $D019
0801          XEXPAN     = $D01D
0801          BORDER     = $D020
0801          BACKGR     = $D021
0801          JOY1       = $DC00
0801          JOY2       = $DC01
0801          TIMALO     = $DC04
0801          RESET     = $FCE2
0801          CHROUT    = $FFD2
0801          PLOT      = $FFF0
0801          SCNKEY    = $FFF9F
0801          GETIN     = $FFE4
0801          UDTIM     = $FFEA

: begin with false basic line
: so you can load and run like basic
0801 18 08 00          .BYT 24,8,0,0,158
0806 32 30 37          .ASC "2076:"
080B 8F 22 14          .BYT 143,34,20,20,20,20
0811 45 41 47          .ASC "EAGLES"
0817 00 00 00          .BYT 0,0,0,0,0

: relocate main program to $4000
081C A0 00            LDY #0
081E A9 56            LDA #56
0820 85 03            STA $03
0822 A9 08            LDA #8
0824 85 04            STA $04
0826 A9 00            LDA #0
0828 85 05            STA $05
082A A9 40            LDA #40
082C 85 06            STA $06
082E A9 E3            LDA #E3
0830 85 FB            STA $FB
0832 A9 1C            LDA #1C
0834 85 FC            STA $FC
0836 A5 03            MOV1 LDA $03
0838 E5 FB            SBC $FB
083A A5 04            LDA $04
083C E5 FC            SBC $FC
083E B0 13            BCS MOVDUN
0840 B1 03            LDA ($03),Y
0842 91 05            STA ($05),Y
0844 E6 03            INC $03
0846 D0 02            BNE MOV2
0848 E6 04            INC $04

084A E6 05            MOV2 INC $05
084C D0 02            BNE MOV3
084E E6 06            INC $06
0850 4C 36 08 MOV3   JMP MOV1
: execute at new location
0853          MOVDUN    = *
0853 4C 00 40        JMP $4000

: $4000 to $44ff
: main code starts here. this part
: handles title screen display and
: animation, instruction screen,
: skill level screen, and main
: setup for game.
4000          * = $4000
: clear variable storage area
4000 A9 3B            LDA #3B
4002 85 FB            STA $FB
4004 A9 03            LDA #3
4006 85 FC            STA $FC
4008 A0 BF            LDY #BF
400A A9 00            LDA #0
400C 91 FB            STA ($FB),Y
400E 88              DEY
400F D0 F9            BNE CLRVAR
: clear sid chip - silence
4011 A9 FF            LDA #FF
4013 85 FB            STA $FB
4015 A9 D3            LDA #D3
4017 85 FC            STA $FC
4019 A0 19            LDY #19
401B A9 00            LDA #0
401D 91 FB            STA ($FB),Y
401F 88              DEY
4020 D0 F9            BNE CLRSID
: print title screen and prompts
4022 A9 93            LDA #93
4024 D0 22 FF        JSR CHROUT
4027 A9 00            LDA #0
4029 8D 20 D0        STA BORDER
402C 8D 21 D0        STA BACKGR
402F 18              CLC
4030 A0 0B            LDY #0B
4032 A2 05            LDX #5
4034 D0 F0 FF        JSR PLOT
4037 A9 00            LDA #0
4039 85 FB            STA $FB
403B A9 45            LDA #45
403D 85 FC            STA $FC
403F D0 AD 42        JSR PRTT
4042 18              CLC
4043 A0 0C            LDY #0C
4045 A2 08            LDX #8
4047 D0 F0 FF        JSR PLOT
404A A9 14            LDA #14
404C 85 FB            STA $FB
404E A9 45            LDA #45
4050 85 FC            STA $FC
4052 D0 AD 42        JSR PRTT

```

Chapter 3

```

; relocate sprite shape data
; and set up sounds
4055 20 1E 43      JSR  SHPREL
4058 18            CLC
4059 A0 0A        LDY  #$0A
405B A2 0E        LDX  #$0E
405D 20 F0 FF     JSR  PLOT
4060 A9 26        LDA  #$26
4062 85 FB        STA  $FB
4064 A9 45        LDA  #$45
4066 85 FC        STA  $FC
4068 20 AD 42     JSR  PRITIT
406B 20 CD 42     JSR  SETTIT
; animation loop for title screen
406E 18            ANITIT
406F A0 0D        CLC
4071 A2 11        LDY  #$0D
4073 20 F0 FF     JSR  PLOT
4076 A9 3B        LDA  #$3B
4078 85 FB        STA  $FB
407A A9 45        LDA  #$45
407C 85 FC        STA  $FC
407E A0 00        LDY  #0
; move sprites on title screen
4080 20 BD 42     JSR  MOVTTT
; random printing color
4083 AD 04 DC     LDA  TIMALO
4086 8D 86 02     STA  PRTCOL
4089 B1 FB        LDA  ($FB),Y
408B C9 7E        CMP  #$7E
408D F0 07        BEQ  KEY1
408F 20 D2 FF     JSR  CHROUT
4092 C8            INY
4093 4C 80 40     JMP  TITMOR
; check for y or n keypress
4096 20 9F FF KEY1 JSR  SCNKEY
4099 20 E4 FF     JSR  GETIN
409C C9 00        CMP  #0
409E F0 CE        BEQ  ANITIT
; is it 'y'
40A0 C9 59        CMP  #$59
40A2 F0 78        BEQ  INSTRU
; is it 'n'
40A4 C9 4E        CMP  #$4E
40A6 F0 03        BEQ  PIKSKL
40A8 4C 6E 40     JMP  ANITIT
;
;
; display pick-skill-level screen
40AB A9 93        PIKSKL LDA  #$93
40AD 20 D2 FF     JSR  CHROUT
40B0 20 74 42     JSR  INSINT
40B3 20 6D 41     JSR  MAINSET
40B6 18            CLC
40B7 A0 11        LDY  #$11
40B9 A2 09        LDX  #9
40BB 20 F0 FF     JSR  PLOT
40BE A9 4C        LDA  #$4C
40C0 85 FB        STA  $FB
40C2 A9 45        LDA  #$45
40C4 85 FC        STA  $FC
40C6 20 AD 42     JSR  PRITIT
40C9 18            CLC
40CA A0 14        LDY  #$14
40CC A2 0B        LDX  #$0B
40CE 20 F0 FF     JSR  PLOT
40D1 A9 5E        LDA  #$5E
40D3 85 FB        STA  $FB
40D5 A9 45        LDA  #$45
40D7 85 FC        STA  $FC
40D9 20 AD 42     JSR  PRITIT

40DC 18
40DD A0 11
40DF A2 0F
40E1 20 F0 FF     JSR  PLOT
40E4 A9 68        LDA  #$68
40E6 85 FB        STA  $FB
40E8 A9 45        LDA  #$45
40EA 85 FC        STA  $FC
40EC 20 AD 42     JSR  PRITIT
; press 1-9 for skill or q to quit
; scan keyboard
40EF 20 9F FF KEY2 JSR  SCNKEY
40F2 20 E4 FF     JSR  GETIN
40F5 C9 00        CMP  #0
40F7 F0 F6        BEQ  KEY2
; is it 'q'
40F9 C9 51        CMP  #$51
40FB D0 03        BNE  KEY3
; quit - jump to power-on reset
40FD 4C E2 FC     JMP  RESET
; reject if lower than '1'
4100 C9 31        CMP  #$31
4102 B0 03        BCS  KEY4
4104 4C EF 40     JMP  KEY2
; reject if higher than '9'
4107 C9 3A        CMP  #$3A
4109 B0 E4        BCS  KEY2
410B 38            SEC
410C E9 30        SBC  #$30
; record skill level
410E 8D 3F 03     STA  $033F
; clear screen
4111 A9 93        LDA  #$93
4113 20 D2 FF     JSR  CHROUT
4116 00 00 00     JSR  SETSID
; go begin game
4119 00 00 00     JMP  MAINLOOP
;
;
; display instruction screen
411C A9 93        INSTRU LDA  #$93
411E 20 D2 FF     JSR  CHROUT
4121 20 74 42     JSR  INSINT
4124 18            CLC
4125 A0 04        LDY  #4
4127 A2 00        LDX  #0
4129 20 F0 FF     JSR  PLOT
412C A9 7C        LDA  #$7C
412E 85 FB        STA  $FB
4130 A9 45        LDA  #$45
4132 85 FC        STA  $FC
4134 A9 05        LDA  #5
4136 85 FD        STA  $FD
4138 A9 47        LDA  #$47
413A 85 FE        STA  $FE
413C A0 00        LDY  #0
413E B1 FB        LDA  ($FB),Y
4140 20 D2 FF     JSR  CHROUT
4143 A5 FB        LDA  $FB
4145 C9 FF        CMP  #$FF
4147 90 F2        BCC  NO1
4149 E6 FC        INC  $FC
414B E6 FB        INC  $FB
414D A5 FB        LDA  $FB
414F C5 FD        CMP  $FD
4151 A5 FC        LDA  $FC
4153 E5 FE        SBC  $FE
4155 90 E7        BCC  INSTR1
; wait for either joystick button
4157 00 00 00     JSR  SCNJOY
415A AD 67 03     LDA  BUTN1

```

The Source Code

```

415D C9 00      CMP #0
415F F0 09      BEQ INSTRDN
4161 AD 6A 03   LDA BUTN2
4164 C9 00      CMP #0
4166 F0 02      BEQ INSTRDN
4168 D0 ED      BNE INSTRMO
; exit to pick-skill-level screen
416A 4C AB 40  INSTRDN  JMP PIKSKL
;
;
; main setup for game
416D          MAINSET = *
; gator shape pointer shadow reg's
; are located at $0341-$0347.
; use shapes $e4-$f1.
416D A9 40      LDA #$40
416F 85 FB      STA $FB
4171 A9 03      LDA #3
4173 85 FC      STA $FC
4175 A9 E4      LDA #$E4
4177 85 FD      STA $FD
4179 A0 08      LDY #8
417B A5 FD      MSET1  LDA $FD
417D 91 FB      STA ($FB),Y
417F E6 FD      INC $FD
4181 E6 FD      INC $FD
4183 88         DEY
4184 C0 01      CPY #1
4186 D0 F3      BNE MSET1
; gator color shadow registers
; are from $0348-$034f. all 13.
4188 A9 EA      LDA #$EA
418A 91 FB      STA ($FB),Y
418C A0 08      LDY #8
418E A9 48      LDA #$48
4190 85 FB      STA $FB
4192 A9 0D      MSET2  LDA $0D
4194 91 FB      STA ($FB),Y
4196 88         DEY
4197 D0 F9      BNE MSET2
; eagle shape pointer shadow reg's
; are from $0351-$0357.
; use shapes $f2-$ff.
4199 A9 50      LDA #$50
419B 85 FB      STA $FB
419D A9 F2      LDA #$F2
419F 85 FD      STA $FD
41A1 A0 08      LDY #8
41A3 A5 FD      MSET3  LDA $FD
41A5 91 FB      STA ($FB),Y
41A7 E6 FD      INC $FD
41A9 E6 FD      INC $FD
41AB 88         DEY
41AC C0 01      CPY #1
41AE D0 F3      BNE MSET3
; eagle color shadow registers
; are from $0359-$0360. all 0.
41B0 A9 F2      LDA #$F2
41B2 91 FB      STA ($FB),Y
41B4 A9 59      LDA #$59
41B6 85 FB      STA $FB
41B8 A0 07      LDY #7
41BA A9 00      MSET4  LDA #0
41BC 91 FB      STA ($FB),Y
41BE 88         DEY
41BF D0 F9      BNE MSET4
41C1 A9 5B      LDA #$5B
41C3 8D 6C 03   STA $036C
41C6 A9 4E      LDA #$4E
41C8 8D 6E 03   STA $036E
41CB A9 03      LDA #3
41CD 8D 6D 03   STA $036D
41D0 8D 6F 03   STA $036F
; gator vertical position shadows
; start at $0372. each gator begins
; $0a pixels lower than the last.
41D3 A9 71      LDA #71
41D5 85 FB      STA $FB
41D7 A9 32      LDA #32
41D9 85 FD      STA $FD
41DB A0 08      LDY #8
41DD A5 FD      MSET5  LDA $FD
41DF 91 FB      STA ($FB),Y
41E1 18         CLC
41E2 69 0A      ADC #$0A
41E4 85 FD      STA $FD
41E6 88         DEY
41E7 D0 F4      BNE MSET5
; eagle vertical position shadows
; start at $037a. each one begins
; $0a pixels higher than the last.
41E9 A9 79      LDA #79
41EB 85 FB      STA $FB
41ED A9 E6      LDA #$E6
41EF 85 FD      STA $FD
41F1 A0 08      LDY #8
41F3 A5 FD      MSET6  LDA $FD
41F5 91 FB      STA ($FB),Y
41F7 38         SEC
41F8 E9 0A      SBC #$0A
41FA 85 FD      STA $FD
41FC 88         DEY
41FD D0 F4      BNE MSET6
; eagle direction flags start at
; $0382. gator direction flags
; start at $038a. for these,
; 255=down, 1=up and 0=clamped.
; creatures alternate directions
; at beginning of game.
41FF A9 81      LDA #81
4201 85 FB      STA $FB
4203 A0 10      LDY #10
4205 A9 FF      MSET7  LDA $FF
4207 91 FB      STA ($FB),Y
4209 88         DEY
420A A9 01      LDA #1
420C 91 FB      STA ($FB),Y
420E 88         DEY
420F D0 F4      BNE MSET7
; pointers to data areas
4211 A9 82      LDA #82
4213 8D 92 03   STA $0392
4216 A9 8A      LDA #8A
4218 8D 94 03   STA $0394
421B A9 72      LDA #72
421D 8D 96 03   STA $0396
4220 A9 7A      LDA #7A
4222 8D 98 03   STA $0398
4225 A9 03      LDA #3
4227 8D 93 03   STA $0393
422A 8D 95 03   STA $0395
422D 8D 97 03   STA $0397
4230 8D 99 03   STA $0399
; set horizontal position for all
; sprites. these don't change
; during the course of the game.
4233 A9 1E      LDA #1E
4235 8D 00 D0   STA $D000
4238 A9 46      LDA #46
423A 8D 02 D0   STA $D002
423D A9 6E      LDA #6E
423F 8D 04 D0   STA $D004
4242 A9 96      LDA #96

```

Chapter 3

```

4244 8D 06 D0      STA  $D006
4247 A9 BE        LDA  #$BE
4249 8D 08 D0      STA  $D008
424C A9 E6        LDA  #$E6
424E 8D 0A D0      STA  $D00A
4251 A9 0F        LDA  #$0F
4253 8D 0C D0      STA  $D00C
4256 A9 37        LDA  #$37
4258 8D 0E D0      STA  $D00E
; msb for horizontals
425B A9 C0        LDA  #$C0
425D 8D 10 D0     STA  $D010
; no vert or horizontal expansion
4260 A9 00        LDA  #0
4262 8D 17 D0     STA  $D017
4265 8D 1D D0     STA  $D01D
; sid volume down low
4268 A9 01        LDA  #1
426A 8D 18 D4     STA  $D418
426D EA          NOP
; enable all sprites
426E A9 FF        LDA  #$FF
4270 8D 15 D0     STA  $D015
4273 60          RTS
;
; set up split screen effects
4274          INSNIT
; disable interrupts
4274 78          SEI
4275 A9 00        LDA  #0
4277 8D 0E DC     STA  $DC0E
; divert irq vector at $0314/$0315
; to point to our code at $4a8c.
427A A9 8C        LDA  #$8C
427C 8D 14 03     STA  $0314
427F A9 4A        LDA  #$4A
4281 8D 15 03     STA  $0315
; top-of-screen split value
4284 A9 0F        LDA  #$0F
4286 8D 88 4A     STA  MYRAS1
; set split at screen line 150
4289 A9 96        LDA  #$96
428B 8D 89 4A     STA  MYRAS2
428E A9 00        LDA  #0
4290 8D 8A 4A     STA  GATSCR
4293 A9 03        LDA  #3
4295 8D 8B 4A     STA  EAGSCR
; hi bit of raster compare reg off
4298 AD 11 D0     LDA  $D011
429B 29 7F        AND  #$7F
429D 8D 11 D0     STA  $D011
; set raster compare to 150
42A0 AD 89 4A     LDA  MYRAS2
42A3 8D 12 D0     STA  $D012
; enable raster compare irq
42A6 A9 01        LDA  #1
42A8 8D 1A D0     STA  $D01A
; enable interrupts
42AB 58          CLI
42AC 60          RTS
;
; text output
42AD A0 00        PRIT
42AF B1 FB        PTMOR
42B1 C9 7E        CMP  #$7E
42B3 F0 07        BEQ  PRTDUN
42B5 20 D2 FF     JSR  CHROUT
42B8 C8          INY
42B9 4C AF 42     JMP  PTMOR

42BC 60          PRTDUN
RTS
;
; move sprites for title screen
42BD EE 01 D0     MOVTT
42C0 CE 03 D0     DEC  $D003
42C3 CE 0D D0     DEC  $D00D
42C6 EE 0F D0     INC  $D00F
; do title sound effects
42C9 20 CA 44     JSR  SNDTIT
42CC 60          RTS
;
; set up sprites for title scrn
42CD A9 1E        SETTIT
42CF 8D 00 D0     STA  $D000
42D2 A9 46        LDA  #$46
42D4 8D 02 D0     STA  $D002
42D7 A9 C0        LDA  #$C0
42D9 8D 10 D0     STA  $D010
42DC A9 0F        LDA  #$0F
42DE 8D 0C D0     STA  $D00C
42E1 A9 37        LDA  #$37
42E3 8D 0E D0     STA  $D00E
42E6 A9 07        LDA  #7
42E8 8D 27 D0     STA  $D027
42EB 8D 2E D0     STA  $D02E
42EE A9 0D        LDA  $D0D
42F0 8D 28 D0     STA  $D028
42F3 8D 2D D0     STA  $D02D
42F6 A9 C3        LDA  #$C3
42F8 8D 15 D0     STA  $D015
42FB A9 FF        LDA  #$FF
42FD 8D FB 07     STA  $07FB
4300 8D FF 07     STA  $07FF
4303 A9 E4        LDA  #$E4
4305 8D F9 07     STA  $07F9
4308 A9 EA        LDA  #$EA
430A 8D FE 07     STA  $07FE
430D 60          RTS
430E EA          NOP
430F EA          NOP
;
; relocate a sprite shape
4310 A0 00        RELONE
4312 B1 FB        RELMOR
4314 91 FD        STA  ($FB),Y
4316 C8          INY
4317 C0 40        CPY  #$40
4319 F0 02        BEQ  RELDUN
431B D0 F5        BNE  RELMOR
431D 60          RELDUN
;
; relocate all sprite shape data
; and set up sid chip for sounds
431E A9 08        SHPREL
4320 85 FB        STA  $FB
4322 A9 47        LDA  #$47
4324 85 FC        STA  $FC
4326 A9 C0        LDA  #$C0
4328 85 FD        STA  $FD
432A A9 3A        LDA  #$3A
432C 85 FE        STA  $FE
432E 20 10 43     JSR  RELONE
4331 A9 80        LDA  #$80
4333 85 FD        STA  $FD

```

The Source Code

4335	20 10 43	JSR	RELONE	43D2	85 FD	STA	\$FD
4338	A9 48	LDA	#\$48	43D4	A9 3E	LDA	#\$3E
433A	85 FB	STA	\$FB	43D6	85 FE	STA	\$FE
433C	A9 00	LDA	#0	43D8	20 10 43	JSR	RELONE
433E	85 FD	STA	\$FD	43DB	A9 00	LDA	#0
4340	A9 3B	LDA	#\$3B	43DD	85 FD	STA	\$FD
4342	85 FE	STA	\$FE	43DF	20 10 43	JSR	RELONE
4344	20 10 43	JSR	RELONE	43E2	A9 08	LDA	#8
4347	A9 40	LDA	#\$40	43E4	85 FB	STA	\$FB
4349	85 FD	STA	\$FD	43E6	A9 49	LDA	#\$49
434B	A9 3A	LDA	#\$3A	43E8	85 FC	STA	\$FC
434D	85 FE	STA	\$FE	43EA	A9 80	LDA	#\$80
434F	20 10 43	JSR	RELONE	43EC	85 FD	STA	\$FD
4352	A9 88	LDA	#\$88	43EE	20 10 43	JSR	RELONE
4354	85 FB	STA	\$FB	43F1	A9 C0	LDA	#\$C0
4356	A9 00	LDA	#0	43F3	85 FD	STA	\$FD
4358	85 FD	STA	\$FD	43F5	A9 3D	LDA	#\$3D
435A	20 10 43	JSR	RELONE	43F7	85 FE	STA	\$FE
435D	A9 40	LDA	#\$40	43F9	20 10 43	JSR	RELONE
435F	85 FD	STA	\$FD	43FC	A9 48	LDA	#\$48
4361	A9 3B	LDA	#\$3B	43FE	85 FB	STA	\$FB
4363	85 FE	STA	\$FE	4400	A9 80	LDA	#\$80
4365	20 10 43	JSR	RELONE	4402	85 FD	STA	\$FD
4368	A9 C8	LDA	#\$C8	4404	20 10 43	JSR	RELONE
436A	85 FB	STA	\$FB	4407	A9 C0	LDA	#\$C0
436C	A9 80	LDA	#\$80	4409	85 FD	STA	\$FD
436E	85 FD	STA	\$FD	440B	A9 3E	LDA	#\$3E
4370	20 10 43	JSR	RELONE	440D	85 FE	STA	\$FE
4373	A9 C0	LDA	#\$C0	440F	20 10 43	JSR	RELONE
4375	85 FD	STA	\$FD	4412	A9 88	LDA	#\$88
4377	A9 39	LDA	#\$39	4414	85 FB	STA	\$FB
4379	85 FE	STA	\$FE	4416	A9 00	LDA	#0
437B	20 10 43	JSR	RELONE	4418	85 FD	STA	\$FD
437E	A9 08	LDA	#8	441A	A9 3F	LDA	#\$3F
4380	85 FB	STA	\$FB	441C	85 FE	STA	\$FE
4382	A9 48	LDA	#\$48	441E	20 10 43	JSR	RELONE
4384	85 FC	STA	\$FC	4421	A9 40	LDA	#\$40
4386	A9 80	LDA	#\$80	4423	85 FD	STA	\$FD
4388	85 FD	STA	\$FD	4425	A9 3D	LDA	#\$3D
438A	20 10 43	JSR	RELONE	4427	85 FE	STA	\$FE
438D	A9 C0	LDA	#\$C0	4429	20 10 43	JSR	RELONE
438F	85 FD	STA	\$FD	442C	A9 C8	LDA	#\$C8
4391	A9 3B	LDA	#\$3B	442E	85 FB	STA	\$FB
4393	85 FE	STA	\$FE	4430	A9 00	LDA	#0
4395	20 10 43	JSR	RELONE	4432	85 FD	STA	\$FD
4398	A9 48	LDA	#\$48	4434	20 10 43	JSR	RELONE
439A	85 FD	STA	\$FB	4437	A9 40	LDA	#\$40
439C	A9 40	LDA	#\$40	4439	85 FD	STA	\$FD
439E	85 FD	STA	\$FD	443B	A9 3F	LDA	#\$3F
43A0	A9 39	LDA	#\$39	443D	85 FE	STA	\$FE
43A2	85 FE	STA	\$FE	443F	20 10 43	JSR	RELONE
43A4	20 10 43	JSR	RELONE	4442	A9 08	LDA	#8
43A7	A9 00	LDA	#0	4444	85 FB	STA	\$FB
43A9	85 FD	STA	\$FD	4446	A9 4A	LDA	#\$4A
43AB	A9 3C	LDA	#\$3C	4448	85 FC	STA	\$FC
43AD	85 FE	STA	\$FE	444A	A9 C0	LDA	#\$C0
43AF	20 10 43	JSR	RELONE	444C	85 FD	STA	\$FD
43B2	A9 88	LDA	#\$88	444E	A9 3C	LDA	#\$3C
43B4	85 FB	STA	\$FB	4450	85 FE	STA	\$FE
43B6	A9 40	LDA	#\$40	4452	20 10 43	JSR	RELONE
43B8	85 FD	STA	\$FD	4455	A9 80	LDA	#\$80
43BA	20 10 43	JSR	RELONE	4457	85 FD	STA	\$FD
43BD	A9 00	LDA	#0	4459	A9 3F	LDA	#\$3F
43BF	85 FD	STA	\$FD	445B	85 FE	STA	\$FE
43C1	A9 39	LDA	#\$39	445D	20 10 43	JSR	RELONE
43C3	85 FE	STA	\$FE	4460	A9 48	LDA	#\$48
43C5	20 10 43	JSR	RELONE	4462	85 FB	STA	\$FB
43C8	A9 C8	LDA	#\$C8	4464	A9 C0	LDA	#\$C0
43CA	85 FB	STA	\$FB	4466	85 FD	STA	\$FD
43CC	A9 48	LDA	#\$48	4468	20 10 43	JSR	RELONE
43CE	85 FC	STA	\$FC	446B	A9 80	LDA	#\$80
43D0	A9 40	LDA	#\$40	446D	85 FD	STA	\$FD

Chapter 3

```

446F A9 3C          LDA #S3C          ; data for all the different sprite
4471 85 FE          STA $FE          ; shapes used in the game.
4473 20 10 43      JSR RELONE
; set up sid chip for sounds
4476 A9 0F          LDA #S0F
4478 8D 05 D4      STA $D405
447B 8D 0C D4      STA $D40C
447E 8D 13 D4      STA $D413
4481 A9 F0          LDA #S0F0
4483 8D 06 D4      STA $D406
4486 8D 0D D4      STA $D40D
4489 8D 14 D4      STA $D414
448C A9 3F          LDA #S3F
448E 8D 00 D4      STA $D400
4491 A9 18          LDA #S18
4493 8D 01 D4      STA $D401
4496 A9 42          LDA #S42
4498 8D 07 D4      STA $D407
449B A9 06          LDA #6
449D 8D 08 D4      STA $D408
44A0 A9 0C          LDA #S0C
44A2 8D 0E D4      STA $D40E
44A5 A9 06          LDA #6
44A7 8D 0F D4      STA $D40F
44AA A9 15          LDA #S15
44AC 8D 04 D4      STA $D404
44AF 8D 0B D4      STA $D40B
44B2 8D 12 D4      STA $D412
44B5 A9 3F          LDA #S3F
44B7 8D 18 D4      STA $D418
44BA A9 F7          LDA #SFF7
44BC 8D 17 D4      STA $D417
44BF A9 22          LDA #S22
44C1 8D 16 D4      STA $D416
44C4 A9 0F          LDA #S0F
44C6 8D 15 D4      STA $D415
44C9 60            RTS
;
; manage sound effects for title
44CA AD 3E 03 SNDDTT LDA $033E
44CD C9 2F          CMP #S2F
44CF F0 04          BEQ SNDDTT2
44D1 EE 3E 03      INC $033E
44D4 60            RTS
44D5 A9 00          LDA #0
44D7 8D 3E 03      STA $033E
44DA EE 3F 03      INC $033F
44DD AD 3F 03      LDA $033F
44E0 8D 0E D4      STA $D40E
44E3 CE 40 03      DEC $0340
44E6 AD 40 03      LDA $0340
44E9 8D 16 D4      STA $D416
44EC AD 3F 03      LDA $033F
44EF C9 35          CMP #S35
44F1 90 06          BCC SNDDTT3
44F3 A9 18          LDA #S18
44F5 8D 01 D4      STA $D401
44F8 60            RTS
44F9 AD 3F 03 SNDDTT3 LDA $033F
44FC 8D 01 D4      STA $D401
44FF 60            RTS
;
4500                * = $4500
;
; graphics data
; the first part of this section
; contains data for the title,
; instruction, and pick-skill-level
; screens. the second part contains

```


The Source Code

4676	11 11 20	.BYT	5,65,76,141,159 17,17,32,32,32,77,7 9,86,69,32,82	4802	18 00 00	.BYT	0,0 24,0,0,24,0,247,0,24 0,0,60
4681	49 47 48	.BYT	73,71,72,84,32,79,8 2,32,76,69,70	480D	00 00 3C	.BYT	0,0,60,0,4,126,32,12 90,48,6
468C	54 20 54	.BYT	84,32,84,79,32,67,7 9,78,84,82,79	4818	FF 60 03	.BYT	255,96,3,255,192,0, 255,0,0,255,0
4697	4C 20 41	.BYT	76,32,65,141,17,32, 32,32,32,32,32	4823	00 FF 00	.BYT	0,255,0,3,255,192,6, 255,96,0,126
46A2	20 20 20	.BYT	32,32,32,68,73,70,7 0,69,82,69,78	482E	00 00 3E	.BYT	0,0,62,0,0,62,0,0,60, 0,0
46AD	54 20 43	.BYT	84,32,67,82,69,65,8 4,85,82,69,141	4839	3C 00 00	.BYT	60,0,0,60,0,0,60,0,0, 120,0
46B8	11 11 20	.BYT	17,17,32,32,5,87,73, 78,32,66,89	4844	01 E0 00	.BYT	1,224,0,0,0,48,0,0,1 20,0,0
46C3	20 50 55	.BYT	32,80,85,83,72,73,7 8,71,32,79,80	484F	7C 00 00	.BYT	124,0,0,252,32,4,18 2,112,12,255,192
46CE	50 4F 4E	.BYT	80,79,78,69,78,84,8 3,32,79,70,70	485A	06 FF 00	.BYT	6,255,0,3,255,0,0,25 4,0,3,255
46D9	20 53 43	.BYT	32,83,67,82,69,69,7 8,141,17,17,158	4865	00 06 FF	.BYT	0,6,255,0,0,127,192, 0,62,96,0
46E4	20 20 20	.BYT	32,32,32,32,32,18,8 0,82,69,83,83	4870	3E 00 00	.BYT	62,0,0,60,0,0,60,0,0, 60,0
46EF	20 46 49	.BYT	32,70,73,82,69,32,6 6,85,84,84,79	487B	00 38 00	.BYT	0,56,0,0,120,0,7,240 0,1,192,0
46FA	4E 20 54	.BYT	78,32,84,79,32,66,6 9,71,73,78,146	4886	00 FF 01	.BYT	0,255,1,192,0,3,240, 0,1,248,64
4705	00 00 00	.BYT	0,0,0,0,3,128,0,15,1 92,2,31	4891	01 EC 60	.BYT	1,296,96,1,190,192, 5,255,128,12,255
4710	80 06 37	.BYT	128,6,55,128,3,125, 128,1,255,160,0	489C	00 07 FE	.BYT	0,7,254,0,0,254,0,7, 254,0,6
471B	FF 30 00	.BYT	255,48,0,127,224,0, 127,0,0,127,224	48A7	FF 00 00	.BYT	255,0,0,127,128,0,1 24,192,0,124,96
4726	00 FF 60	.BYT	0,255,96,1,254,0,3,6 2,0,6,62	48B2	00 78 20	.BYT	0,120,32,0,56,0,0,12 0,0,16,112
4731	00 04 1E	.BYT	0,4,30,0,0,28,0,0,30, 0,0	48BD	00 0C F0	.BYT	0,12,240,0,7,192,0,1 128,0,0
473C	0E 08 00	.BYT	14,8,0,15,48,0,3,224 0,1,128	48C8	00 00 00	.BYT	0,0,0,0,0,0,0,0,0,0, 0
4747	00 00 0C	.BYT	0,0,12,0,0,30,0,0,62, 0,4	48D3	00 0E 00	.BYT	0,14,0,112,31,60,24 8,127,255,254,126
4752	3F 00 0E	.BYT	63,0,14,109,32,3,25 5,48,0,255,96	48DE	7E 7E FC	.BYT	126,126,252,153,63, 239,24,247,143,60, 241
475D	00 FF C0	.BYT	0,255,192,0,127,0,0, 255,192,0,255	48E9	C7 E7 E3	.BYT	199,231,227,7,195,2 24,3,195,192,7,231
4768	60 03 FE	.BYT	96,3,254,0,6,124,0,0 124,0,0	48F4	E0 1E FF	.BYT	224,30,255,120,14,6 0,48,0,0,0,0
4773	3C 00 00	.BYT	60,0,0,60,0,0,60,0,0, 28,0	48FF	00 00 00	.BYT	0,0,0,0,0,0,0,255,0 0
477E	00 1E 00	.BYT	0,30,0,0,15,224,0,3, 128,255,0	490A	00 00 00	.BYT	0,0,0,0,0,0,0,0,0,3 0
4789	18 00 00	.BYT	24,0,0,60,0,0,60,0,4, 126,32	4915	00 78 7F	.BYT	0,120,127,189,254,2 55,255,255,222,126 123
4794	0C 5A 30	.BYT	12,90,48,6,255,96,3, 255,192,0,255	4920	8E DB 71	.BYT	142,219,113,15,24,2 40,3,24,192,7,255
479F	00 00 FF	.BYT	0,0,255,0,0,255,0,3, 255,192,6	492B	E0 07 81	.BYT	224,7,129,224,7,195 224,3,255,192,7
47AA	FF 60 00	.BYT	255,96,0,126,0,0,12 4,0,0,124,0	4936	FF E0 3E	.BYT	255,224,62,60,124,2 8,0,56,0,0,0
47B5	00 3C 00	.BYT	0,60,0,0,60,0,0,60,0, 0,60	4941	00 00 00	.BYT	0,0,0,0,0,0,255,0,0,0 0
47C0	00 00 1E	.BYT	0,0,30,0,0,7,128,0,0, 24,0	494C	00 00 00	.BYT	0,0,0,0,0,48,0,12,25 4,0,127
47CB	00 3C 00	.BYT	0,60,0,0,60,0,4,126, 32,12,90	4957	FF 99 FF	.BYT	255,153,255,223,25 5,251,30,60,120,14, 255
47D6	30 06 FF	.BYT	48,6,255,96,3,255,1 92,0,255,0,0	4962	70 07 18	.BYT	112,7,24,224,7,24,2 24,15,255,240,15
47E1	FF 00 00	.BYT	255,0,0,255,0,3,255, 192,6,255,96	496D	18 F0 07	.BYT	24,240,7,129,224,7, 255,224,7,255,224
47EC	00 7E 00	.BYT	0,126,0,0,126,0,0,12 6,0,0,60	4978	1E 7E 78	.BYT	30,126,120,62,0,124 32,0,4,0,0
47F7	00 00 3C	.BYT	0,0,60,0,0,60,0,0,60, 0,0				

Chapter 3

```

4983 00 00 00      .BYT  0,0,0,0,0,0,0,0,0,0,0
498E 00 00 00      .BYT  0,0,248,0,31,254,0
                    ; 127,127,129
4999 FE 3F DB      .BYT  254,63,219,252,31,6
                    ; 0,248,14,255,112,7
49A4 99 E0 0F      .BYT  153,224,15,24,240,1
                    ; 5,255,240,14,126,1
                    ; 12
49AF 07 18 E0      .BYT  7,24,224,3,195,192,
                    ; 14,255,224,14,126
49BA 70 3E 00      .BYT  112,62,0,124,34,0,6
                    ; 8,0,0,0,0
49C5 00 00 FF      .BYT  0,0,255,0,0,0,192,0,
                    ; 3,224,0
49D0 07 7C 00      .BYT  7,124,0,62,127,129,
                    ; 254,63,129,252,31
49DB DB F8 06      .BYT  219,248,6,255,96,3,
                    ; 60,192,7,153,224
49E6 0E 18 70      .BYT  14,24,112,15,255,24
                    ; 0,12,255,48,7,60
49F1 E0 03 81      .BYT  224,3,129,192,3,255
                    ; 192,15,126,240,24
49FC 00 18 3E      .BYT  0,24,62,0,124,50,0,7
                    ; 6,0,0,0
4A07 00 C0 00      .BYT  0,192,0,3,192,0,3,11
                    ; 2,0,14,126
4A12 00 7E 3F      .BYT  0,126,63,129,252,31
                    ; 129,248,31,195,24
                    ; 8
4A1D 06 FF 60      .BYT  6,255,96,3,126,192,
                    ; 7,153,224,14,24
4A28 70 0F DB      .BYT  112,15,219,240,12,2
                    ; 55,48,6,60,96,3
                    ;
4A33 81 C0 03      .BYT  129,192,3,255,192,1
                    ; 4,126,224,28,0,56
4A3E 3F 00 FC      .BYT  63,0,252,49,0,140,4
                    ; 8,0,12,255,28
4A49 00 38 30      .BYT  0,56,48,0,12,56,0,28
                    ; 62,0,124
4A54 3F 81 FC      .BYT  63,129,252,31,195,2
                    ; 48,15,195,240,6,25
                    ; 5
4A5F 60 03 7E      .BYT  96,3,126,192,7,189,
                    ; 224,14,24,112,11
4A6A DB D0 0C      .BYT  219,208,12,255,48,6
                    ; 60,96,3,129,192
4A75 03 FF C0      .BYT  3,255,192,7,126,224
                    ; 28,0,56,62,0
4A80 7C 33 00      .BYT  124,51,0,204,24,0,2
                    ; 4,0,15,150,0,3
                    ;
                    ;
                    ; we're above screen split.
                    ; transfer eagle shadow registers
                    ; into sprite control registers.
                    ;
4A8C      * = $A8C
                    ;
                    ; this section contains all of the
                    ; main game routines.
                    ;
                    ;
                    ; interrupt handler.
                    ; this code is initialized in the
                    ; routine insint, which sets up
                    ; the split-screen machinery.
                    ;
4A8C 00 00 00      LDY  RASTER
4A8F A2 02         LDX  #2
                    ; compare raster position to
                    ; screen split point
4A91 00 00 00      CPY  MYRAS1
4A94 F0 07         BEQ  RASTA
                    ; fudge a bit to minimize jitter
4A96 86           DEY
                    ;
4A97 00 00 00      CPY  MYRAS1
4A9A F0 01         BEQ  RASTA
4A9C CA           DEX
                    ; set border and background colors
                    ; accordingly, depending on which
                    ; zone you're in (upper or lower).
4A9D 00 FF FF RASTA LDA  GATSCR - 1,X
4AA0 00 00 00      STA  BACKGR
4AA3 00 00 00      STA  BORDER
                    ; set raster for next compare.
                    ; again, this depends on where you
                    ; are currently.
4AA6 00 FF FF      LDA  MYRAS1 - 1,X
4AA9 00 00 00      STA  RASTER
4AAC 00 00 00      LDA  VICIRQ
4AAF 00 00 00      STA  VICIRQ
                    ; when <<<1 you're in the gator zone
4AB2 E0 01         CPX  #1
4AB4 D0 06         BNE  GATRIR
                    ; hi-ho, hi-ho, to eaglesville we go...
4AB6 20 C8 4A      JSR  EGLSTUF
4AB9 4C C2 4A      JMP  ENDIRQ
                    ;
4ABC 20 59 4B GATRIR JSR  GTRSTUF
4ABF 00 00 00      JSR  UDTIM
                    ; exit this interrupt
4AC2 68           ENDIRQ
4AC3 A8           PLA
4AC4 68           PLA
4AC5 AA           TAX
4AC6 68           PLA
4AC7 40           RTI
                    ;
                    ;
                    ;
4AC8 AD 7A 03 EGLSTUF LDA  $037A
4ACB 8D 01 D0      STA  $D001
4ACE AD 7B 03      LDA  $037B
4AD1 8D 03 D0      STA  $D003
4AD4 AD 7C 03      LDA  $037C
4AD7 8D 05 D0      STA  $D005
4ADA AD 7D 03      LDA  $037D
4ADD 8D 07 D0      STA  $D007
4AE0 AD 7E 03      LDA  $037E
4AE3 8D 09 D0      STA  $D009
4AE6 AD 7F 03      LDA  $037F
4AE9 8D 0B D0      STA  $D00B
4AEC AD 80 03      LDA  $0380
4AEF 8D 0D D0      STA  $D00D
4AF2 AD 81 03      LDA  $0381
4AF5 8D 0F D0      STA  $D00F
4AF8 AD 41 03      LDA  $0341
4AFB 8D F8 07      STA  $0378
4AFE AD 42 03      LDA  $0342
4B01 8D F9 07      STA  $0379
4B04 AD 43 03      LDA  $0343
4B07 8D FA 07      STA  $037A
4B0A AD 44 03      LDA  $0344
4B0D 8D FB 07      STA  $037B
4B10 AD 45 03      LDA  $0345
4B13 8D FC 07      STA  $037C
4B16 AD 46 03      LDA  $0346
4B19 8D FD 07      STA  $037D
4B1C AD 47 03      LDA  $0347
4B1F 8D FE 07      STA  $037E
4B22 AD 48 03      LDA  $0348
4B25 8D FF 07      STA  $037F
4B28 AD 49 03      LDA  $0349

```

The Source Code

```

4B2B 8D 27 D0      STA  $D027
4B2E AD 4A 03      LDA  $034A
4B31 8D 2B D0      STA  $D028
4B34 AD 4B 03      LDA  $034B
4B37 8D 29 D0      STA  $D029
4B3A AD 4C 03      LDA  $034C
4B3D 8D 2A D0      STA  $D02A
4B40 AD 4D 03      LDA  $034D
4B43 8D 2B D0      STA  $D02B
4B46 AD 4E 03      LDA  $034E
4B49 8D 2C D0      STA  $D02C
4B4C AD 4F 03      LDA  $034F
4B4F 8D 2D D0      STA  $D02D
4B52 AD 50 03      LDA  $0350
4B55 8D 2E D0      STA  $D02E
4B58 60            RTS

;
;
; we're below screen split.
; transfer gator shadow registers
; into sprite control registers.
;
4B59 AD 72 03 GTRSTUF LDA  $0372
4B5C 8D 01 D0      STA  $D001
4B5F AD 73 03      LDA  $0373
4B62 8D 03 D0      STA  $D003
4B65 AD 74 03      LDA  $0374
4B68 8D 05 D0      STA  $D005
4B6B AD 75 03      LDA  $0375
4B6E 8D 07 D0      STA  $D007
4B71 AD 76 03      LDA  $0376
4B74 8D 09 D0      STA  $D009
4B77 AD 77 03      LDA  $0377
4B7A 8D 0B D0      STA  $D00B
4B7D AD 78 03      LDA  $0378
4B80 8D 0D D0      STA  $D00D
4B83 AD 79 03      LDA  $0379
4B86 8D 0F D0      STA  $D00F
4B89 AD 51 03      LDA  $0351
4B8C 8D 08 07      STA  $07F8
4B8F AD 52 03      LDA  $0352
4B92 8D 09 07      STA  $07F9
4B95 AD 53 03      LDA  $0353
4B98 8D 0A 07      STA  $07FA
4B9B AD 54 03      LDA  $0354
4B9E 8D 0B 07      STA  $07FB
4BA1 AD 55 03      LDA  $0355
4BA4 8D 0C 07      STA  $07FC
4BA7 AD 56 03      LDA  $0356
4BAA 8D 0D 07      STA  $07FD
4BAD AD 57 03      LDA  $0357
4BB0 8D 0E 07      STA  $07FE
4BB3 AD 58 03      LDA  $0358
4BB6 8D 0F 07      STA  $07FF
4BB9 AD 59 03      LDA  $0359
4BBC 8D 27 D0      STA  $D027
4BBF AD 5A 03      LDA  $035A
4BC2 8D 28 D0      STA  $D028
4BC5 AD 5B 03      LDA  $035B
4BC8 8D 29 D0      STA  $D029
4BCB AD 5C 03      LDA  $035C
4BCE 8D 2A D0      STA  $D02A
4BD1 AD 5D 03      LDA  $035D
4BD4 8D 2B D0      STA  $D02B
4BD7 AD 5E 03      LDA  $035E
4BDA 8D 2C D0      STA  $D02C
4BDD AD 5F 03      LDA  $035F
4BE0 8D 2D D0      STA  $D02D
4BE3 AD 60 03      LDA  $0360
4BE6 8D 2E D0      STA  $D02E
4BE9 60            RTS

;
;
; clear sid chip and set it up
; for main game sound effects.
;
4BEA                      SETSID      =      *
4BEA A0 00              LDY          #0
4BEC A9 00              LDA          #0
4BEE 85 FB              STX          $FB
4BF0 A9 D4              LDA          #$D4
4BF2 85 FC              STX          $FC
4BF4 A9 00              LDA          #0
4BF6 91 FB              STA          ($FB),Y
4BF8 C0 0F              STY          $0F
4BFA F0 04              BEQ          $IDDUN
4BFC C8                INY
4BFD 4C F4 4B          JMP          $IDZERO
4C00 A9 0F              LDA          #0
4C02 8D 18 D4          STA          $D418
4C05 A9 00              LDA          #0
4C07 8D 17 D4          STA          $D417
4C0A A9 60              LDA          #$60
4C0C 8D 05 D4          STA          $D405
4C0F A9 19              LDA          #$19
4C11 8D 13 D4          STA          $D413
4C14 A9 0F              LDA          #0
4C16 8D 00 D4          STA          $D400
4C19 8D 07 D4          STA          $D407
4C1C 8D 0E D4          STA          $D40E
4C1F A9 0F              LDA          #0
4C21 8D 01 D4          STA          $D401
4C24 A9 4F              LDA          #0
4C26 8D 08 D4          STA          $D408
4C29 A9 4F              LDA          #0
4C2B 8D 0F D4          STA          $D40F
4C2E A9 00              LDA          #0
4C30 8D 04 D4          STA          $D404
4C33 A9 00              LDA          #0
4C35 8D 0B D4          STA          $D40B
4C38 A9 00              LDA          #0
4C3A 8D 12 D4          STA          $D412
4C3D A9 00              LDA          #0
4C3F 8D 06 D4          STA          $D406
4C42 8D 0D D4          STA          $D40D
4C45 8D 14 D4          STA          $D414
4C48 A9 60              LDA          #$60
4C4A 8D 0C D4          STA          $D40C
4C4D 60                RTS

;
;
; sound effects. called at various
; points in joystick rtn scnjoy.
;
4C4E A9 60              SOUND1    LDA          #$60
4C50 8D 05 D4          STA          $D405
4C53 00 00 00          LDA          MYRAS2
4C56 4A                LSR          A
4C57 4A                LSR          A
4C58 4A                LSR          A
4C59 85 FB              STA          $FB
4C5B A9 24              LDA          #$24
4C5D E5 FB              SBC          $FB
4C5F 8D 01 D4          STA          $D401
4C62 A9 60              LDA          #$60
4C64 8D 05 D4          STA          $D405
4C67 A9 20              LDA          #0
4C69 8D 04 D4          STA          $D404
4C6C A9 21              LDA          #0
4C6E 8D 04 D4          STA          $D404
4C71 60                RTS

4C72 A9 60              SOUND2    LDA          #$60
4C74 8D 0C D4          STA          $D40C

```

Chapter 3

```

4C77 00 00 00      LDA  MYRAS2
4C7A 4A             LSR  A
4C7B 4A             LSR  A
4C7C 4A             LSR  A
4C7D 8D 08 D4      STA  $D408
4C80 A9 80          LDA  #$80
4C82 8D 08 D4      STA  $D408
4C85 A9 81          LDA  #$81
4C87 8D 08 D4      STA  $D408
4C8A 60             RTS

;
4C8B A9 2A SOUND3   LDA  #$2A
4C8D 8D 3C 03      STA  $033C
4C90 A9 10          LDA  #$10
4C92 8D 12 D4      STA  $D412
4C95 A9 15          LDA  #$15
4C97 8D 12 D4      STA  $D412
4C9A 60             RTS

;
4C9B A9 04 SOUND4   LDA  #4
4C9D 8D 3C 03      STA  $033C
4CA0 A9 10          LDA  #$10
4CA2 8D 12 D4      STA  $D412
4CA5 A9 15          LDA  #$15
4CA7 8D 12 D4      STA  $D412
4CAA 60             RTS

;
; an eagle has clamped on the border
4CAB             EQLCLAMP = *
; turn opponent's background
; orange, make a sound effect,
; make the eagles thrash with joy,
; turn gators' screen black again.
;
4CAB A9 09          LDA  #9
4CAD 00 00 00      STA  GATSCR
4CB0 A5 FB          LDA  $FB
4CB2 85 FD          STA  $FD
4CB4 A5 FC          LDA  $FC
4CB6 85 FE          STA  $FE
4CB8 98            TYA
4CB9 8D 3D 03      STA  $033D
4CBC A9 10          LDA  #$10
4CBE 8D 12 D4      STA  $D412
4CC1 A9 15          LDA  #$15
4CC3 8D 12 D4      STA  $D412
4CC6 A2 1F          LDX  #$1F
4CC8 A9 00          LDA  #0
4CCA 8D 0F D4      STA  $D40F
4CCD 20 97 4D EWIGMOR JSR  EGLWIGL
4CD0 8A            TXA
4CD1 8D 3E 03      STA  $033E
4CD4 0A            ASL  A
4CD5 0A            ASL  A
4CD6 0A            ASL  A
4CD7 0A            ASL  A
4CD8 8D 0F D4      STA  $D40F
4CDB A0 12          LDY  #$12
4CDD 20 5B 4D      JSR  DELAY
4CE0 AD 3E 03      LDA  $033E
4CE3 AA            TAX
4CE4 CA            DEX
4CE5 D0 E6          BNE  EWIGMOR
4CE7 AD 3D 03      LDA  $033D
4CEA A8            TRY
4CEB A5 FD          LDA  $FD
4CED 85 FB          STA  $FB
4CFE A5 FE          LDA  $FE
4CF1 85 FC          STA  $FC

4CF3 A9 E0          LDA  #E0
4CF5 8D 0F D4      STA  $D40F
4CF8 A9 85          LDA  #85
4CFA 8D 12 D4      STA  $D412
4CFD A9 00          LDA  #0
4CFF 00 00 00      STA  GATSCR
4D02 60             RTS

;
; a gator has clamped on the border.
4D03             GATCLAMP = *
; turn eagles' background
; orange, make a sound effect,
; make gators thrash with joy,
; turn eagles screen green again.
4D03 A9 09          LDA  #9
4D05 00 00 00      STA  GATSCR
4D08 A5 FB          LDA  $FB
4D0A 85 FD          STA  $FD
4D0C A5 FC          LDA  $FC
4D0E 85 FE          STA  $FE
4D10 98            TYA
4D11 8D 3D 03      STA  $033D
4D14 A9 10          LDA  #$10
4D16 8D 12 D4      STA  $D412
4D19 A9 15          LDA  #$15
4D1B 8D 12 D4      STA  $D412
4D1E A2 18          LDX  #$18
4D20 A9 08          LDA  #8
4D22 8D 0F D4      STA  $D40F
4D25 20 75 4D GWIGMOR JSR  GATWIGL
4D28 8A            TXA
4D29 8D 3E 03      STA  $033E
4D2C 0A            ASL  A
4D2D 0A            ASL  A
4D2E EA            NOP
4D2F EA            NOP
4D30 8D 08 D4      STA  $D408
4D33 A0 12          LDY  #$12
4D35 20 5B 4D      JSR  DELAY
4D38 AD 3E 03      LDA  $033E
4D3B AA            TAX
4D3C CA            DEX
4D3D D0 E6          BNE  GWIGMOR
4D3F AD 3D 03      LDA  $033D
4D42 A8            TRY
4D43 A5 FD          LDA  $FD
4D45 85 FB          STA  $FB
4D47 A5 FE          LDA  $FE
4D49 85 FC          STA  $FC
4D4B A9 E0          LDA  #E0
4D4D 8D 0F D4      STA  $D40F
4D50 A9 15          LDA  #$15
4D52 8D 12 D4      STA  $D412
4D55 A9 03          LDA  #3
4D57 00 00 00      STA  GATSCR
4D5A 60             RTS

;
; does what the name implies
4D5B C0 00          DELAY CPY  #0
4D5D F0 15          BEQ  DEL5
4D5F C0 01          CPY  #1
4D61 D0 03          BNE  DEL1
4D63 4C 6F 4D      JMP  DEL3
4D66 88            DEY
4D67 A2 C6          LDX  #$C6
4D69 CA            DEX
4D6A D0 FD          BNE  DEL2
4D6C 88            DEY

```

The Source Code

```

4D6D D0 EC      BNE DELAY
4D6F A2 C3     DEL3 LDX  #$C3
4D71 CA        DEL4 DEX
4D72 D0 FD     BNE DEL4
4D74 60        DEL5 RTS

;
; cycle all gator sprite shapes
4D75 A9 41     GATWIGL LDA  #$41
4D77 85 FB     STA  $FB
4D79 A9 03     LDA  #3
4D7B 85 FC     STA  $FC
4D7D A0 00     LDY  #0
4D7F B1 FB     GATWIG2 LDA  ($FB),Y
4D81 C9 F1     CMP  #$F1
4D83 F0 08     BEQ  GATWIG3
4D85 18        CLC
4D86 69 01     ADC  #1
4D88 91 FB     STA  ($FB),Y
4D8A 4C 91 4D JMP  GATWIG4
4D8D A9 E4     GATWIG3 LDA  #$E4
4D8F 91 FB     STA  ($FB),Y
4D91 C8        GATWIG4 INY
4D92 C0 08     CPY  #8
4D94 D0 E9     BNE  GATWIG2
4D96 60        RTS

;
; cycle all eagle sprite shapes
4D97 A9 51     EGLWIGL LDA  #$51
4D99 85 FB     STA  $FB
4D9B A9 03     LDA  #3
4D9D 85 FC     STA  $FC
4D9F A0 00     LDY  #0
4DA1 B1 FB     EGLWIG2 LDA  ($FB),Y
4DA3 C9 FF     CMP  #$FF
4DA5 F0 08     BEQ  EGLWIG3
4DA7 18        CLC
4DA8 69 01     ADC  #1
4DAA 91 FB     STA  ($FB),Y
4DAC 4C B3 4D JMP  EGLWIG4
4DAF A9 F2     EGLWIG3 LDA  #$F2
4DB1 91 FB     STA  ($FB),Y
4DB3 C8        EGLWIG4 INY
4DB4 C0 08     CPY  #8
4DB6 D0 E9     BNE  EGLWIG2
4DB8 A0 08     LDY  #8
4DBA EE 3C 03 EGLWIG5 INC  $033C
4DBD AD 3C 03 LDA  $033C
4DC0 8D 0F D4 STA  $D40F
4DC3 88        DEY
4DC4 D0 F4     BNE
4DC6 60        RTS

;
;
; eagle border check. come here
; when you press eagle fire button.
; check if you're in clamping distance.
; handle cases where you're already
; clamped or where there's a gator
; across the border from you.
4DC7          EBORCHEK = *
4DC7 AD 6C 03 LDA  $036C
4DCA 18        CLC
4DCB 69 29     ADC  #$29
4DCD 85 FB     STA  $FB
4DCF AD 6D 03 LDA  $036D
4DD2 85 FC     STA  $FC
4DD4 AD 67 03 LDA  $0367

4DD7 C9 00     CMP  #0
4DD9 D0 61     BNE  ECHKEXIT
4ddb A0 00     LDY  #0
4ddd B1 FB     LDA  ($FB),Y
4ddf C9 00     CMP  #0
4de1 F0 43     BEQ  EPUTBAK
4de3 A5 FB     LDA  $FB
4de5 38        SEC
4de6 E9 10     SBC  #$10
4de8 85 FB     STA  $FB
4dea A0 00     LDY  #0
4dec B1 FB     LDA  ($FB),Y
4dee 18        CLC
4def 69 19     ADC  #$19
4df1 00 00 00 CMP  MYRAS2
4df4 B0 03     BCS  ECLAMPIT
4df6 4C 31 4E JMP
4df9 20 AB 4C ECLAMPIT JSR
4dfc 00 00 00 LDA
4dff 38        SEC
4e00 E9 17     SBC  #$17
4e02 91 FB     STA  ($FB),Y
4e04 A5 FB     LDA  $FB
4e06 18        CLC
4e07 69 10     ADC  #$10
4e09 85 FB     STA  $FB
4e0b A9 00     LDA  #0
4e0d 91 FB     STA  ($FB),Y
4e0f A5 FB     LDA  $FB
4e11 18        CLC
4e12 69 08     ADC  #8
4e14 85 FB     STA  $FB
4e16 B1 FB     LDA  ($FB),Y
4e18 C9 00     CMP  #0
4e1a F0 03     BEQ  ESTAT
4e1c 4C 3C 4E JMP  ECHKEXIT
4e1f A9 FF     ESTAT CMP  #$FF
4e21 91 FB     LDA  ($FB),Y
4e23 4C 3C 4E JMP  ECHKEXIT
4e26 A5 FB     LDA  $FB
4e28 38        SEC
4e29 E9 10     SBC  #$10
4e2b 85 FB     STA  $FB
4e2d A9 25     LDA  #$25
4e2f 91 FB     STA  ($FB),Y
4e31 A5 FB     EREPUT LDA  $FB
4e33 18        CLC
4e34 69 10     ADC  #$10
4e36 85 FB     STA  $FB
4e38 A9 01     LDA  #1
4e3a 91 FB     STA  ($FB),Y
4e3c 60        ECHKEXIT RTS

;
4E3D          GBORCHEK = *
; handle gator border stuff. come here
; when you're pressing gator fire button.
; check if you're in clamping distance.
; handle cases where you're already
; clamped or there's an eagle across
; the border from you.
4E3D AD 6E 03 LDA  $036E
4E40 18        CLC
4E41 69 41     ADC  #$41
4E43 85 FB     STA  $FB
4E45 AD 6F 03 LDA  $036F
4E48 85 FC     STA  $FC
4E4A AD 6A 03 LDA  $036A
4E4D C9 00     CMP  #0
4E4F D0 61     BNE  GCHKEXIT
4E51 A0 00     LDY  #0

```

Chapter 3

```

4E53 B1 FB          LDA ($FB),Y
4E55 C9 00          CMP #0
4E57 F0 43          BEQ GPUBAK
4E59 A5 FB          LDA $FB
4E5B 38             SEC
4E5C E9 10          SBC # $10
4E5E 85 FB          STA $FB
4E60 A0 00          LDY #0
4E62 B1 FB          LDA ($FB),Y
4E64 38             SEC
4E65 E9 08          SBC #8
4E67 00 00 00      CMP MYRAS2
4E6A 90 03          BCC GCLAMPIT
4E6C 4C A7 4E      JMP GREPUT
4E6F 00 00 00      GCLAMPIT
4E72 18             CLC
4E73 69 05          ADC #5
4E75 91 FB          STA ($FB),Y
4E77 20 03 4D      JSR GATCLAMP
4E7A A5 FB          LDA $FB
4E7C 18             CLC
4E7D 69 10          ADC # $10
4E7F 85 FB          STA $FB
4E81 A9 00          LDA #0
4E83 91 FB          STA ($FB),Y
4E85 A5 FB          LDA $FB
4E87 38             SEC
4E88 E9 08          SBC #8
4E8A 85 FB          STA $FB
4E8C B1 FB          LDA ($FB),Y
4E8E C9 00          CMP #0
4E90 F0 03          BEQ GSTAT
4E92 4C B2 4E      JMP GCHKEXIT
4E95 A9 01          GSTAT
4E97 91 FB          STA ($FB),Y
4E99 4C B2 4E      JMP GCHKEXIT
4E9C A5 FB          GPUBAK
4E9E 38             SEC
4E9F E9 10          SBC # $10
4EA1 85 FB          STA $FB
4EA3 A9 F7          LDA # $F7
4EA5 91 FB          STA ($FB),Y
4EA7 A5 FB          GREPUT
4EA9 18             CLC
4EAA 69 10          ADC # $10
4EAC 85 FB          STA $FB
4EAE A9 FF          LDA # $FF
4EB0 91 FB          STA ($FB),Y
4EB2 60             GCHKEXIT
;
; control creature colors.
; make the creature currently
; under control of the joystick
; flash colors.
;
4EB3 FLASHIT =
; do eagles first
4EB3 AD 6C 03      LDA $036C
4EB6 85 FB          STA $FB
4EB8 AD 6D 03      LDA $036D
4EBB 85 FC          STA $FC
4EBD A0 00          LDY #0
4EBF AD 65 03      LDA $0365
4EC2 C9 FF          CMP # $FF
4EC4 F0 06          BEQ EFLA1
4EC6 C9 01          CMP #1
4EC8 F0 20          BEQ EFLA3
4ECA D0 3B          BNE EFLA5
4ECC A9 00          LDA #0
4ECE 91 FB          STA ($FB),Y
4ED0 A5 FB          LDA $FB

4ED2 C9 59          CMP # $59
4ED4 F0 0A          BEQ EFLA2
4ED6 C6 FB          DEC $FB
4ED8 A5 FB          LDA $FB
4EDA 8D 6C 03      STA $036C
4EDD 4C 07 4F      JMP EFLA5
4EE0 A9 60          LDA # $60
4EE2 85 FB          STA $FB
4EE4 8D 6C 03      STA $036C
4EE7 4C 07 4F      JMP EFLA5
4EEA A0 00          LDY #0
4EEC A9 00          LDA #0
4EEE 91 FB          STA ($FB),Y
4EF0 A5 FB          LDA $FB
4EF2 C9 60          CMP # $60
4EF4 F0 0A          BEQ EFLA4
4EF6 E6 FB          INC $FB
4EF8 A5 FB          LDA $FB
4EFA 8D 6C 03      STA $036C
4EFD 4C 07 4F      JMP EFLA5
4F00 A9 59          LDA # $59
4F02 85 FB          STA $FB
4F04 8D 6C 03      STA $036C
4F07 AD 70 03      LDA $0370
4F0A C9 11          CMP # $11
4F0C D0 05          BNE EFLA6
4F0E A9 0C          LDA # $0C
4F10 8D 70 03      STA $0370
4F13 EE 70 03      INC $0370
4F16 AD 70 03      LDA $0370
4F19 91 FB          STA ($FB),Y
; do same for gators
4F1B AD 6E 03      LDA $036E
4F1E 85 FB          STA $FB
4F20 AD 6F 03      LDA $036F
4F23 85 FC          STA $FC
4F25 AD 64 03      LDA $0364
4F28 C9 FF          CMP # $FF
4F2A F0 06          BEQ GFLA1
4F2C C9 01          CMP #1
4F2E F0 2D          BEQ GFLA3
4F30 D0 3B          BNE GFLA5
4F32 A0 00          LDY #0
4F34 A9 0D          LDA # $0D
4F36 91 FB          STA ($FB),Y
4F38 A5 FB          LDA $FB
4F3A C9 49          CMP # $49
4F3C F0 0A          BEQ GFLA2
4F3E C6 FB          DEC $FB
4F40 A5 FB          LDA $FB
4F42 8D 6E 03      STA $036E
4F45 4C 6F 4F      JMP GFLA5
4F48 A9 50          LDA # $50
4F4A 85 FB          STA $FB
4F4C 8D 6E 03      STA $036E
4F4F 4C 6F 4F      JMP GFLA5
4F52 A0 00          LDY #0
4F54 A9 0D          LDA # $0D
4F56 91 FB          STA ($FB),Y
4F58 A5 FB          LDA $FB
4F5A C9 50          CMP # $50
4F5C F0 0A          BEQ GFLA4
4F5E E6 FB          INC $FB
4F60 A5 FB          LDA $FB
4F62 8D 6E 03      STA $036E
4F65 4C 6F 4F      JMP GFLA5
4F68 A9 49          LDA # $49
4F6A 85 FB          STA $FB
4F6C 8D 6E 03      STA $036E
4F6F AD 71 03      LDA $0371
4F72 C9 09          CMP #9
4F74 D0 05          BNE GFLA6

```

The Source Code

```

4F76 A9 03          LDA #3
4F78 8D 71 03     STA $0371
4F7B EE 71 03 GFLA6 INC $0371
4F7E AD 71 03     LDA $0371
4F81 91 FB         STA ($FB),Y
4F83 AD A8 03     LDA $03A8
4F86 F0 04        BEQ ECOL
4F88 CE A8 03     DEC $03A8
4F8B 60           RTS

; color eagles black if moving
; or blue if clamped to border.
4F8C A9 03          LDA #3
4F8E 8D A8 03     STA $03A8
4F91 A9 82          LDA #$82
4F93 85 FB         STA $FB
4F95 A9 03          LDA #3
4F97 85 FC         STA $FC
4F99 A0 00          LDY #0
4F9B B1 FB         LDA ($FB),Y
4F9D C9 00          CMP #0
4F9F D0 1A         BNE ECOL3
4FA1 A5 FB         LDA $FB
4FA3 38           SEC
4FA4 E9 29         SBC #$29
4FA6 85 FB         STA $FB
4FA8 A9 06          LDA #6
4FAA 91 FB         STA ($FB),Y
4FAC A5 FB         LDA $FB
4FAE 18           CLC
4FAF 69 29         ADC #$29
4FB1 85 FB         STA $FB
4FB3 C0 07         ECOL2  CPY #7
4FB5 F0 19         BEQ GCOL
4FB7 C8           INY
4FB8 4C 9B 4F     JMP ECOL1
4FBB A5 FB         ECOL3  LDA $FB
4FBD 38           SEC
4FBE E9 29         SBC #$29
4FC0 85 FB         STA $FB
4FC2 A9 00          LDA #0
4FC4 91 FB         STA ($FB),Y
4FC6 A5 FB         LDA $FB
4FC8 18           CLC
4FC9 69 29         ADC #$29
4FCB 85 FB         STA $FB
4FCD 4C B3 4F     JMP ECOL2

; color gators light green if moving
; or white if clamped to border.
4FD0 A9 8A          GCOL  LDA #$8A
4FD2 85 FB         STA $FB
4FD4 A9 03          LDA #3
4FD6 85 FC         STA $FC
4FD8 A0 00          LDY #0
4FDA B1 FB         GCOL1  LDA ($FB),Y
4FDC C9 00          CMP #0
4FDE D0 1B         BNE GCOL3
4FE0 A5 FB         LDA $FB
4FE2 38           SEC
4FE3 E9 41         SBC #$41
4FE5 85 FB         STA $FB
4FE7 A9 01          LDA #1
4FE9 91 FB         STA ($FB),Y
4FEB A5 FB         LDA $FB
4FED 18           CLC
4FEE 69 41         ADC #$41
4FF0 85 FB         STA $FB
4FF2 C0 07         GCOL2  CPY #7
4FF4 F0 04         BEQ GCOLOUT
4FF6 C8           INY
4FF7 4C DA 4F     JMP GCOL1
4FFA 60           GCOLOUT RTS
4FFB A5 FB         GCOL3  LDA $FB

4FFD 38           SEC
4FFE E9 41         SBC #$41
5000 85 FB         STA $FB
5002 A9 0D          LDA #$0D
5004 91 FB         STA ($FB),Y
5006 A5 FB         LDA $FB
5008 18           CLC
5009 69 41         ADC #$41
500B 85 FB         STA $FB
500D 4C F2 4F     JMP GCOL2

;
; move eagles up or down between
; screen split and top of screen.
5010          EGLMOVE = *
5010 A9 82          LDA #$82
5012 85 FB         STA $FB
5014 A9 03          LDA #3
5016 85 FC         STA $FC
5018 A9 72          LDA #$72
501A 85 FD         STA $FD
501C A9 03          LDA #3
501E 85 FE         STA $FE
; do all 8 eagles
5020 A0 00          LDY #0
; check eagle's direction
5022 B1 FB         EUDCHK  LDA ($FB),Y
; is it moving down
5024 C9 FF         CMP #$FF
5026 F0 06         BEQ EMVDN
; is it moving up
5028 C9 01         CMP #1
502A F0 26         BEQ EMVUP
; guess it's clamped- next one
502C D0 1B         BNE EMVALL
502E B1 FD         EMVDN  LDA ($FD),Y
5030 18           CLC
5031 69 17         ADC #$17
; will eagle hit lower border of
; its zone if it moves down
5033 00 00 00     CMP MYRAS2
5036 B0 0A         BCS EDNHIT
; ok to move. add one to vertical
; position for this eagle
5038 B1 FD          LDA ($FD),Y
503A 18           CLC
503B 69 01         ADC #1
503D 91 FD         STA ($FD),Y
503F 4C 49 50     JMP EMVALL
; hit border. make sound effect
5042 20 C2 50     JSR EBRDSDN
; reverse direction flag. 1=up
5045 A9 01          LDA #1
5047 91 FB         STA ($FB),Y
5049 C0 07         EMVALL  CPY #7
504B F0 04         BEQ EMVOUT
504D C8           INY
504E 4C 22 50     JMP EUDCHK
5051 60           EMVOUT  RTS
; eagle is moving up. check upper
; boundary
5052 B1 FD         EMVUP  LDA ($FD),Y
5054 C9 1E         CMP #$1E
5056 90 0A         BCC EUPHIT
; ok to move up one pixel.
5058 B1 FD          LDA ($FD),Y
505A 38           SEC
505B E9 01         SBC #1
505D 91 FD         STA ($FD),Y
505F 4C 49 50     JMP EMVALL
; hit upper border. reverse direction

```

Chapter 3

```

5062 A9 FF EUPHIT LDA #$$F
5064 91 FB STA ($FB),Y
5066 4C 49 50 JMP EMVALL
;
; move gators up or down between
; screen split and bottom of screen
;
5069 GATMOVE = *
5069 A9 8A LDA #$$A
506B 85 FB STA $FB
506D A9 03 LDA #3
506F 85 FC STA $FC
5071 A9 7A LDA #$$A
5073 85 FD STA $FD
5075 A9 03 LDA #3
5077 85 FE STA $FE
; do all eight gators
5079 A0 00 LDY #0
; check gator's direction
507B B1 FB GUDCHK LDA ($FB),Y
; is it moving down
507D C9 FF CMP #$$F
507F FO 06 BEQ GMVDN
; is it moving up
5081 C9 01 CMP #1
5083 FO 1F BEQ GMVUP
; it's clamped
5085 D0 14 BNE GMVALL
; will gator hit bottom of screen
; if it moves down
5087 B1 FD GMVDN LDA ($FD),Y
5089 C9 FB CMP #$$B
508B B0 0A BCS GDNHIT
; ok to move down. add one to vertical
; position for this gator.
508D B1 FD LDA ($FD),Y
508F 18 CLC
5090 69 01 ADC #1
5092 91 FD STA ($FD),Y
5094 4C 9B 50 JMP GMVALL
; hit bottom. reverse direction.
5097 A9 01 GDNHIT LDA #1
5099 91 FB STA ($FB),Y
509B C0 07 GMVALL CPY #7
509D FO 04 BEQ GMVOUT
509F C8 INY
50A0 4C 7B 50 JMP GUDCHK
50A3 60 GMVOUT RTS
; gator is moving up. check if it
; has hit screen split yet.
50A4 B1 FD GMVUP LDA ($FD),Y
50A6 38 SEC
50A7 E9 06 SBC #6
50A9 00 00 00 MYRAS2 CMP MYRAS2
50AC 90 0A BCC GUPHIT
; ok to move up. subtract one from
; vertical position for this gator.
50AE B1 FD LDA ($FD),Y
50B0 38 SEC
50B1 E9 01 SBC #1
50B3 91 FD STA ($FD),Y
50B5 4C 9B 50 JMP GMVALL
; hit screen border. make sound effect.
50B8 20 D7 50 GLPHIT JSR GBRDSND
; reverse direction flag. 255==down.
50BB A9 FF LDA #$$F
50BD 91 FB STA ($FB),Y
50BF 4C 9B 50 JMP GMVALL
;
; sound of eagle hitting border

```

```

50C2 A9 10 EBRDSND LDA #$$10
50C4 8D 0C D4 STA $D40C
50C7 A9 45 LDA #$$45
50C9 8D 08 D4 STA $D408
50CC A9 80 LDA #$$80
50CE 8D 0B D4 STA $D40B
50D1 A9 81 LDA #$$81
50D3 8D 0B D4 STA $D40B
50D6 60 RTS
;
; sound of gator hitting border
;
50D7 A9 24 GBRDSND LDA #$$24
50D9 8D 05 D4 STA $D405
50DC A9 2F LDA #$$2F
50DE 8D 01 D4 STA $D401
50E1 A9 10 LDA #$$10
50E3 8D 04 D4 STA $D404
50E6 A9 11 LDA #$$11
50E8 8D 04 D4 STA $D404
50EB 60 RTS
;
; scan both joysticks, store
; direction and button values, and
; handle joystick-related sound effects.
50EC 00 00 00 SCNJOY LDA JOY1
50EF A0 00 LDY #0
50F1 A2 00 LDX #0
50F3 4A LSR A
50F4 B0 01 BCS SCNJO2
50F6 88 DEY
50F7 4A LSR A
50F8 B0 01 BCS SCNJO3
50FA C8 INY
50FB 4A LSR A
50FC B0 01 BCS SCNJO4
50FE CA DEX
50FF 4A LSR A
5100 B0 01 BCS SCNJO5
5102 E8 INX
5103 4A LSR A
5104 8E 66 03 STX $0366
5107 2A ROL A
5108 29 01 AND #1
510A 8D 68 03 STA $0368
510D 00 00 00 LDA JOY2
5110 A0 00 LDY #0
5112 A2 00 LDX #0
5114 4A LSR A
5115 B0 01 BCS SCNJO6
5117 88 DEY
5118 4A LSR A
5119 B0 01 BCS SCNJO7
511B C8 INY
511C 4A LSR A
511D B0 01 BCS SCNJO8
511F CA DEX
5120 4A LSR A
5121 B0 01 BCS SCNJO9
5123 E8 INX
5124 4A LSR A
5125 8E 69 03 STX $0369
5128 2A ROL A
5129 29 01 AND #1
512B 8D 68 03 STA $0368
512E AD 68 03 LDA $0368
5131 C9 00 CMP #0
5133 D0 11 BNE SCNJOA
5135 AD 67 03 LDA $0367

```


The Source Code

```

5138 C9 00          CMP #0
513A D0 12          BNE SCNJOB
513C AD AB 03      LDA $03AB
513F C9 FF          CMP #$FF
5141 FO 0B          BEQ SCNJOB
5143 EE AB 03      INC $03AB
5146 A9 01 SCNJOA  LDA #1
5148 8D 67 03      STA $0367
514B 4C 5C 51      JMP SCNJOC
514E 20 9B 4C SCNJOB JSR SOUND4
5151 A9 00          LDA #0
5153 8D AB 03      STA $03AB
5156 AD 68 03      LDA $0368
5159 8D 67 03      STA $0367
515C AD 6B 03 SCNJOC LDA $036B
515F C9 00          CMP #0
5161 D0 11          BNE SCNJOD
5163 AD 6A 03      LDA $036A
5166 C9 00          CMP #0
5168 D0 12          BNE SCNJOE
516A AD AC 03      LDA $03AC
516D C9 FF          CMP #$FF
516F FO 0B          BEQ SCNJOE
5171 EE AC 03      INC $03AC
5174 A9 01 SCNJOD  LDA #1
5176 8D 6A 03      STA $036A
5179 4C 6A 51      JMP SCNJOF
517C 20 8B 4C SCNJOE JSR SOUND3
517F A9 00          LDA #0
5181 8D AC 03      STA $03AC
5184 AD 6B 03      LDA $036B
5187 8D 6A 03      STA $036A
518A AD 66 03 SCNJOF LDA $0366
518D C9 00          CMP #0
518F FO 11          BEQ SCNJ10
5191 AD 65 03      LDA $0365
5194 C9 00          CMP #0
5196 FO 12          BEQ SCNJ11
5198 AD AD 03      LDA $03AD
519B C9 01          CMP #1
519D FO 0B          BEQ SCNJ11
519F EE AD 03      INC $03AD
51A2 A9 00 SCNJ10  LDA #0
51A4 8D 65 03      STA $0365
51A7 4C B8 51      JMP SCNJ12
51AA 20 72 4C SCNJ11 JSR SOUND2
51AD A9 00          LDA #0
51AF 8D AD 03      STA $03AD
51B2 AD 66 03      LDA $0366
51B5 8D 65 03      STA $0365
51B8 AD 69 03 SCNJ12 LDA $0369
51BB C9 00          CMP #0
51BD FO 11          BEQ SCNJ13
51BF AD 64 03      LDA $0364
51C2 C9 00          CMP #0
51C4 FO 12          BEQ SCNJ14
51C6 AD AE 03      LDA $03AE
51C9 C9 01          CMP #1
51CB FO 0B          BEQ SCNJ14
51CD EE AE 03      INC $03AE
51D0 A9 00 SCNJ13  LDA #0
51D2 8D 64 03      STA $0364
51D5 4C E6 51      JMP SCNJ15
51D8 20 4E 4C SCNJ14 JSR SOUND1
51DB A9 00          LDA #0
51DD 8D AE 03      STA $03AE
51E0 AD 69 03      LDA $0369
51E3 8D 64 03      STA $0364
51E6 60 SCNJ15     RTS
;
; check how many creatures are
; pushing border on each side.
; move border up or down accordingly,
; along with all clamped creatures.
; check if either side has won.
51E7 HUZAHD = *
51E7 A9 82      LDA #$82
51E9 85 FB      STA $FB
51EB A9 03      LDA #3
51ED 85 FC      STA $FC
51EF A9 00      LDA #0
51F1 8D 9A 03   STA $039A
51F4 A0 00      LDY #0
51F6 B1 FB      LDA ($FB),Y
51F8 C9 00      CMP #0
51FA D0 03      BNE ECNTALL
51FC EE 9A 03   INC $039A
51FF C0 07      ECNTALL CPY #7
5201 FO 04      BEQ GCNT
5203 C8          INY
5204 4C F6 51   JMP ECNT
5207 A9 8A      GCNT LDA #$8A
5209 85 FB      STA $FB
520B A9 03      LDA #3
520D 85 FC      STA $FC
520F A9 00      LDA #0
5211 8D 9B 03   STA $039B
5214 A0 00      LDY #0
5216 B1 FB      GCNT1 LDA ($FB),Y
5218 C9 00      CMP #0
521A D0 03      BNE GCNTALL
521C EE 9B 03   INC $039B
521F C0 07      GCNTALL CPY #7
5221 FO 04      BEQ HUMORE
5223 C8          INY
5224 4C 16 52   JMP GCNT1
;
; compare number of creatures
; pushing on both sides.
5227 AD 9B 03   HUMORE LDA $039B
522A CD 9A 03   CMP $039A
522D 90 6D      BCC EWINCHK
; if sides are equal, don't move border.
522F FO 6A      BEQ HUZDUN
;
; gators are pushing border upward.
; check if they've won yet.
5231 GWINCHK = *
5231 00 00 00   LDA MYRAS2
5234 C9 3E      CMP #$3E
5236 D0 03      BNE UPSPLIT
5238 4C 8B 53   JMP GATWON
; haven't won, move border up a line.
523B 38          UPSPLIT SEC
523C E9 01      SBC #1
523E 00 00 00   STA MYRAS2
; move clamped eagles upward
; along with border.
5241 A9 82      LDA #$82
5243 85 FB      STA $FB
5245 A9 03      LDA #3
5247 85 FC      STA $FC
5249 A0 00      LDY #0
524B B1 FB      UPEGL LDA ($FB),Y
524D C9 00      CMP #0
524F D0 15      BNE UPEGALL
5251 A5 FB      LDA $FB
5253 38          SEC
5254 E9 10      SBC #$10
5256 85 FB      STA $FB

```

Chapter 3

5258	B1 FB	LDA	(\$FB),Y	52CA	A5 FB	LDA	\$FB
525A	38	SEC		52CC	18	CLC	
525B	E9 01	SBC	#1	52CD	69 10	ADC	#\$10
525D	91 FB	STA	(\$FB),Y	52CF	85 FB	STA	\$FB
525F	A5 FB	LDA	\$FB	52D1	C0 07	CPY	#7
5261	18	CLC		52D3	F0 04	BEQ	DNGAT
5262	69 10	ADC	#\$10	52D5	C8	INY	
5264	85 FB	STA	\$FB	52D6	4C B6 52	JMP	DNEAG
5266	C0 07	CPY	#7	; move clamped gators downward			
5268	F0 04	BEQ	UPGAT	; along with the border.			
526A	C8	INY		52D9	A9 8A	DNGAT	LDA
526B	4C 4B 52	JMP	UPEGL	52DB	85 FB		STA
; move clamped gators upward				52DD	A9 03		LDA
; along with border				52DF	85 FC		STA
526E	A9 8A	UPGAT	LDA	52E1	A0 00		LDY
5270	85 FB		STA	52E3	B1 FB	DNGAT1	LDA
5272	A9 03		LDA	52E5	C9 00		CMF
5274	85 FC		STA	52E7	D0 15		BNE
5276	A0 00		LDY	52E9	A5 FB		LDA
5278	B1 FB	UPGAT1	LDA	52EB	38		SEC
527A	C9 00		CMF	52EC	E9 10		SBC
527C	D0 15		BNE	52EE	85 FB		STA
527E	A5 FB		LDA	52F0	B1 FB		LDA
5280	38		SEC	52F2	18		CLC
5281	E9 10		SBC	52F3	69 01		ADC
5283	85 FB		STA	52F5	91 FB		STA
5285	B1 FB		LDA	52F7	A5 FB		LDA
5287	38		SEC	52F9	18		CLC
5288	E9 01		SBC	52FA	69 10		ADC
528A	91 FB		STA	52FC	85 FB		STA
528C	A5 FB		LDA	52FE	C0 07	DNGATALL	CPY
528E	18		CLC	5300	F0 04		BEQ
528F	69 10		ADC	5302	C8		INY
5291	85 FB		STA	5303	4C E3 52		JMP
5293	C0 07	UPGTALL	CPY	5306	60	XDNSPL	RTS
5295	F0 04		BEQ	; eagles have won. clear screen,			
5297	C8		INY	; make expanded eagle move down			
5298	4C 78 52		JMP	; the screen, do sound effects.			
529B	60	HUZDUN	RTS	;			
; eagles are pushing the border							
; down. see if they've won yet.							
529C		EWINCHK	=	5307		EGLWON	=
529C	00 00 00		LDA	5307	A9 60		LDA
529F	C9 E8		CMF	5309	8D 0C D4		STA
52A1	D0 03		BNE	530C	8D 05 D4		STA
52A3	4C 07 53		JMP	530F	A9 00		LDA
; eagles haven't won yet. move				5311	8D 5C 03		STA
; the border down one line.				5314	A9 20		LDA
52A6	18	DNSPLIT	CLC	5316	8D 75 03		STA
52A7	69 01		ADC	5319	A9 FF		LDA
52A9	00 00 00		STA	531B	8D 85 03		STA
; move clamped eagles downward				531E	A9 FA		LDA
; along with border				5320	00 00 00		STA
52AC	A9 82		LDA	5323	A9 08		LDA
52AE	85 FB		STA	5325	00 00 00		STA
52B0	A9 03		LDA	5328	A9 08		LDA
52B2	85 FC		STA	532A	00 00 00		STA
52B4	A0 00		LDY	532D	00 00 00		STA
52B6	B1 FB	DNEAG	LDA	5330	A9 5C		LDA
52B8	C9 00		CMF	5332	8D AF 03		STA
52BA	D0 15		BNE	5335	A9 10	EGLWN1	LDA
52BC	A5 FB		LDA	5337	8D 04 D4		STA
52BE	38		SEC	533A	A9 15		LDA
52BF	E9 10		SBC	533C	8D 04 D4		STA
52C1	85 FB		STA	533F	AD 3D 03		LDA
52C3	B1 FB		LDA	5342	8D 0F D4		STA
52C5	18		CLC	5345	AD 75 03		LDA
52C6	69 01		ADC	5348	8D 01 D4		STA
52C8	91 FB		STA	534B	20 97 4D		JSR
				534E	20 10 50		JSR
				5351	20 10 50		JSR
				5354	A0 40		LDY

The Source Code

```

5356 20 5B 4D      JSR  DELAY
5359 AD AF 03      LDA  $03AF
536C C9 01        CMP  #1
535E F0 06        BEQ  EGLWIN2
5360 CE AF 03      DEC  $03AF
5363 4C 35 53      JMP  EGLWIN1
5366 A9 00          LDA  #0
5368 00 00 00      STA  ENABLE
536B A9 FF          LDA  #$FF
536D 8D 3D 03      STA  $033D
5370 A0 FF          LDY  #$FF
5372 20 5B 4D      JSR  DELAY
5375 AD 3D 03      LDA  $033D
5378 D0 06          BNE  EGLWIN4
537A CE 3D 03      DEC  $033D
537D 4C 70 53      JMP  EGLWIN3
; sprites back to normal size
5380 A9 00          LDA  #0
5382 00 00 00      STA  YEXPAN
5385 00 00 00      STA  XEXPAN
; game ended with eagle victory.
; pick skill level for next game.
5388 00 00 00      JMP  PIKSKL
;
;
; gators have won. make sound effect
; while expanded gator moves from
; bottom to top of screen.
538B          GATWON = *
538B A9 15          LDA  #$15
538D 8D 13 D4      STA  $D413
5390 8D 05 D4      STA  $D405
5393 A9 0D          LDA  #$0D
5395 8D 4C 03      STA  $034C
5398 A9 E0          LDA  #$E0
539A 8D 7D 03      STA  $037D
539D A9 01          LDA  #1
539F 8D 8D 03      STA  $038D
53A2 A9 1C          LDA  #$1C
53A4 00 00 00      STA  MYRAS2
53A7 A9 08          LDA  #8
53A9 00 00 00      STA  ENABLE
53AC A9 08          LDA  #8
53AE 00 00 00      STA  XEXPAN
53B1 00 00 00      STA  YEXPAN
53B4 A9 5C          LDA  #$5C
53B6 8D AF 03      STA  $03AF
53B9 A9 10          LDA  #$10
53BB 8D 12 D4      STA  $D412
53BE A9 15          LDA  #$15
53C0 8D 12 D4      STA  $D412
53C3 AD 3D 03      LDA  $033D
53C6 8D 01 D4      STA  $D401
53C9 AD 7D 03      LDA  $037D
53CC 8D 0F D4      STA  $D40F
53CF 20 75 4D      JSR  GATWIGL
53D2 20 69 50      JSR  GATMOVE
53D5 20 69 50      JSR  GATMOVE
53D8 A0 40          LDY  #$40
53DA 20 5B 4D      JSR  DELAY
53DD AD AF 03      LDA  $03AF
53E0 C9 01          CMP  #1
53E2 F0 06          BEQ  GATWON2
53E4 CE AF 03      DEC  $03AF
53E7 4C B9 53      JMP  GATWON1
53EA A9 00          LDA  #0
53EC 00 00 00      STA  ENABLE
53EF A9 FF          LDA  #$FF
53F1 8D 3D 03      STA  $033D
53F4 A0 FF          LDY  #$FF
53F6 20 5B 4D      JSR  DELAY
53F9 AD 3D 03      LDA  $033D
53FC D0 06          BNE  GATWON4
53FE CE 3D 03      DEC  $033D
5401 4C F4 53      JMP  GATWON3
5404 A9 00          LDA  #0
5406 00 00 00      STA  YEXPAN
5409 00 00 00      STA  XEXPAN
; game ended with gator victory.
; pick skill level for next game.
540C 00 00 00      JMP  PIKSKL
;
;
;-----
; main game loop
;-----
540F          MAINLOOP = *
540F AC 3F 03      LDA  $033F
5412 A9 0A          LDA  #$0A
5414 8D 3F 03      STA  $033F
5417 CE 3F 03      DEC  $033F
541A 88            DEY
541B D0 FA          BNE  LOOP1
541D AD A0 03      LDA  $03A0
5420 C9 3C          CMP  #$3C
5422 F0 06          BEQ  LOOP3
5424 EE A0 03      INC  $03A0
5427 4C 3B 54      JMP  LOOP4
542A 20 EC 50      JSR  SCNJY
542D 20 C7 4D      JSR  EBORCHEK
5430 20 3D 4E      JSR  GBORCHEK
5433 20 B3 4E      JSR  FLASHIT
5436 A9 00          LDA  #0
5438 8D A0 03      STA  $03A0
543B AD A1 03      LDA  $03A1
543E C9 0F          CMP  #$0F
5440 F0 06          BEQ  LOOP5
5442 EE A1 03      INC  $03A1
5445 4C 53 54      JMP  LOOP6
5448 20 10 50      JSR  EGLMOVE
544B 20 69 50      JSR  GATMOVE
544E A9 00          LDA  #0
5450 8D A1 03      STA  $03A1
5453 AD A2 03      LDA  $03A2
5456 C9 1E          CMP  #$1E
5458 F0 06          BEQ  LOOP7
545A EE A2 03      INC  $03A2
545D 4C 6B 54      JMP  LOOP8
5460 20 75 4D      JSR  GATWIGL
5463 20 97 4D      JSR  EGLWIGL
5466 A9 00          LDA  #0
5468 8D A2 03      STA  $03A2
546B AD A3 03      LDA  $03A3
546E C9 FF          CMP  #$FF
5470 F0 06          BEQ  LOOP9
5472 EE A3 03      INC  $03A3
5475 4C 80 54      JMP  LOOP10
5478 20 E7 51      JSR  HUZAHED
547B A9 00          LDA  #0
547D 8D A3 03      STA  $03A3
5480 AE 3F 03      LDA  $033F
5483 A0 18          LDY  $033F
5485 88            DEY
5486 D0 FD          BNE  LOOP12
5488 CA            DEX
5489 D0 F8          BNE  LOOP11
548B 4C 1D 54      JMP  LOOP2

```



Appendices



How to Type In Programs

In order to make it as easy as possible to type in the programs in this book, we've included two program-entry aids written in BASIC: "The Automatic Proofreader" and "MLX." To assist you in understanding how to enter these programs, COMPUTE! has established the following listing conventions.

Generally, BASIC program listings like the one for MLX contain words within braces which spell out any special characters: {DOWN} means to press the cursor-down key; {5 SPACES} means to press the space bar five times.

To indicate that a key should be *shifted* (pressed while you're holding down the SHIFT key), the key is underlined in our listings. For example, S means to type the S key while holding down the SHIFT key. This will appear on your screen as a heart symbol. If you find an underlined key enclosed in braces—for example, {10 N}—you should type the key as many times as indicated. In that case, you would enter ten shifted N's. If a key is enclosed in special brackets, [<>], you should hold down the *Commodore key* while pressing the key inside the special brackets. (The Commodore key is the key in the lower left corner of the keyboard.) Again, if the key is preceded by a number, you should press the key as many times as indicated; for example, [<9@>] means to type Commodore-@ nine times.

Refer to the following table when entering cursor- and color-control keys:

Appendix A

Keyboard Conventions

When You Read:

When You Read:	Press:	See:
{CLR}	SHIFT CLR/HOME	
{HOME}	CLR/HOME	
{UP}	SHIFT ↑ CRSR ↓	
{DOWN}	↑ CRSR ↓	
{LEFT}	SHIFT ← CRSR →	
{RIGHT}	← CRSR →	
{RVS}	CTRL 9	
{OFF}	CTRL 0	
{BLK}	CTRL 1	
{WHT}	CTRL 2	
{RED}	CTRL 3	
{CYN}	CTRL 4	
{PUR}	CTRL 5	
{GRN}	CTRL 6	
{BLU}	CTRL 7	
{YEL}	CTRL 8	

When You Read:

When You Read:	Press:	See:
{ 1 }	COMMODORE 1	
{ 2 }	COMMODORE 2	
{ 3 }	COMMODORE 3	
{ 4 }	COMMODORE 4	
{ 5 }	COMMODORE 5	
{ 6 }	COMMODORE 6	
{ 7 }	COMMODORE 7	
{ 8 }	COMMODORE 8	
{ F1 }	f1	
{ F2 }	SHIFT f1	
{ F3 }	f3	
{ F4 }	SHIFT f3	
{ F5 }	f5	
{ F6 }	SHIFT f5	
{ F7 }	f7	
{ F8 }	SHIFT f7	
←		
↑	SHIFT	

The Automatic Proofreader

Philip I. Nelson

“The Automatic Proofreader” helps you type in program listings for the Commodore 64 and 128, and prevents nearly every kind of typing mistake.

Type in the “Proofreader” *exactly* as listed. Since the program can’t check itself, type carefully to avoid mistakes. Don’t omit any lines, even if they contain unfamiliar commands. After you’ve finished, save a copy or two on disk or tape before running it. This is important because the Proofreader erases the BASIC portion of itself when you run it, leaving only the machine language portion in memory.

Next, type **RUN** and press RETURN. After announcing which computer it’s running on, the Proofreader displays the message *Proofreader Active*. Now you’re ready to type in a BASIC program.

Two Letters

Every time you finish typing a line and press RETURN, the Proofreader displays a two-letter checksum in the upper left corner of the screen. Compare this result with the two-letter checksum printed to the left of the line in the program listing.

The one program in this book that can be used in conjunction with the Proofreader is “MLX: Machine Language Entry Program,” which is found in Chapter 2. Make sure you use the Proofreader to type in MLX, since it is vital that you have the latter program entered correctly. If there is an error in MLX, you will not be able to type in any of the seven machine language games in this book.

If the letters match, it’s almost certain the line was typed correctly. If the letters don’t match, check for your mistake and correct the line.

Appendix B

What It Does, What It Doesn't

The Proofreader ignores spaces not enclosed in quotation marks, so you can omit or add spaces between keywords and still see a matching checksum. However, since spaces *inside* quotation marks are almost always significant, the Proofreader pays attention to them. For example, `10 PRINT"THIS IS BASIC"` generates a different checksum than does `10 PRINT"THIS ISBA SIC"`.

A common typing error is transposition—typing two successive characters in the wrong order, like `PIRNT` instead of `PRINT` or `64378` instead of `64738`. The Proofreader is sensitive to the *position* of each character within the line and thus catches transposition errors.

The Proofreader does *not* accept keyword abbreviations (for example, `?` instead of `PRINT`). If you prefer to use abbreviations, you can still check the line by `LISTing` it after typing it in, moving the cursor back to the line, and pressing `RETURN`. `LISTing` the line substitutes the full keyword for the abbreviation and allows the Proofreader to work properly. The same technique works for rechecking programs you've already typed in.

If you're using the Proofreader on the Commodore 128, *do not perform any GRAPHIC commands while the Proofreader is active*. When you perform a command like `GRAPHIC 1`, the computer moves everything at the start of BASIC program space—including the Proofreader—to another memory area, causing the Proofreader to crash. The same thing happens if you *run* any program with a `GRAPHIC` command while the Proofreader is in memory.

Though the Proofreader doesn't interfere with other BASIC operations, it's a good idea to disable it before running another program. However, the Proofreader is purposely difficult to dislodge: It's not affected by tape or disk operations, or by pressing `RUN/STOP-RESTORE`. The simplest way to disable it is to turn the computer off, then on. A gentler method is to `SYS` to the computer's built-in reset routine (`SYS 65341` for the 128; `64738` for the 64). These reset routines erase any program in memory, so be sure to save the program you're typing in *before* entering the `SYS` command.

If you own a Commodore 64, you may already have wondered whether the Proofreader works with other programming utilities like "MetaBASIC." The answer is, generally, yes, *if you're using a 64 and activate the Proofreader after installing the*

other utility. For example, first load and activate MetaBASIC; then load and run the Proofreader.

When using the Proofreader with another utility, you should disable *both* programs before running a BASIC program. While the Proofreader seems unaffected by most utilities, there's no way to promise that it will work with any and every combination of utilities you might want to use. The more utilities activated, the more fragile the system becomes.

The Automatic Proofreader

Read Appendix A before entering this program.

```

10 VEC=PEEK(772)+256*PEEK(773):LO=43:HI=44
20 PRINT"{CLR}{WHT}AUTOMATIC PROOFREADER FOR ";:IF
   VEC=42364 THEN PRINT "C-64"
30 IF VEC=50556 THEN PRINT "VIC-20{BLU}"
40 IF VEC=35158 THEN GRAPHIC CLR:PRINT "PLUS/4 & 1
   6"
50 IF VEC=17165 THEN LO=45:HI=46:GRAPHIC CLR:PRINT
   "128{WHT}"
60 SA=(PEEK(LO)+256*PEEK(HI))+6:ADR=SA
70 FOR J=0 TO 166:READ BYT:POKE ADR,BYT:ADR=ADR+1:
   CHK=CHK+BYT:NEXT
80 IF CHK<>20570 THEN PRINT "*ERROR* CHECK TYPING
   {SPACE}IN DATA STATEMENTS":END
90 FOR J=1 TO 5:READ RF,LF,HF:RS=SA+RF:HB=INT(RS/2
   56):LB=RS-(256*HB)
100 CHK=CHK+RF+LF+HF:POKE SA+LF,LB:POKE SA+HF,HB:N
   EXT
110 IF CHK<>22054 THEN PRINT "*ERROR* RELOAD PROGR
   AM AND CHECK FINAL LINE":END
120 POKE SA+149,PEEK(772):POKE SA+150,PEEK(773)
130 IF VEC=17165 THEN POKE SA+14,22:POKE SA+18,23:
   POKESA+29,224:POKESA+139,224
140 PRINT CHR$(147);CHR$(17);"PROOFREADER ACTIVE":
   SYS SA
150 POKE HI,PEEK(HI)+1:POKE (PEEK(LO)+256*PEEK(HI)
   )-1,0:NEW
160 DATA 120,169,73,141,4,3,169,3,141,5,3
170 DATA 88,96,165,20,133,167,165,21,133,168,169
180 DATA 0,141,0,255,162,31,181,199,157,227,3
190 DATA 202,16,248,169,19,32,210,255,169,18,32
200 DATA 210,255,160,0,132,180,132,176,136,230,180
210 DATA 200,185,0,2,240,46,201,34,208,8,72
220 DATA 165,176,73,255,133,176,104,72,201,32,208
230 DATA 7,165,176,208,3,104,208,226,104,166,180
240 DATA 24,165,167,121,0,2,133,167,165,168,105
250 DATA 0,133,168,202,208,239,240,202,165,167,69
260 DATA 168,72,41,15,168,185,211,3,32,210,255

```

Appendix B

270 DATA 104,74,74,74,74,168,185,211,3,32,210
280 DATA 255,162,31,189,227,3,149,199,202,16,248
290 DATA 169,146,32,210,255,76,86,137,65,66,67
300 DATA 68,69,70,71,72,74,75,77,80,81,82,83,88
310 DATA 13,2,7,167,31,32,151,116,117,151,128,129,
167,136,137

To order your copy of *COMPUTE!'s More Machine Language Games for the Commodore 64 Disk*, call our toll-free US order line: 1-800-346-6767 (in NY 212-887-8525) or send your prepaid order to:

*COMPUTE!'s More Machine Language Games
for the Commodore 64 Disk*

COMPUTE! Publications
P.O. Box 5038
F.D.R. Station
New York, NY 10150

All orders must be prepaid (check, charge, or money order). NC residents add 5% sales tax. NY residents add 8.25% sales tax.

Send _____ copies of *COMPUTE!'s More Machine Language Games for the Commodore 64 Disk* at \$12.95 per copy. (947BDSK)

Subtotal \$ _____

Shipping and Handling: \$2.00/disk \$ _____

Sales tax (if applicable) \$ _____

Total payment enclosed \$ _____

- Payment enclosed
 Charge Visa MasterCard American Express

Acct. No. _____ Exp. Date _____
(Required)

Name _____

Address _____

City _____ State _____ Zip _____

Please allow 4-5 weeks for delivery.





More Amazing Games

Creating computer arcade-style games isn't easy. That's especially true of games that have everything—speed, great play action, and a high entertainment value. Only with machine language (ML), the native language of your computer, can all this be achieved.

You'll find seven games in this book. All are written entirely in machine language.

- **Space Arena** puts you in a futuristic shoving match where the slightest mistake means your spaceship is twisted metal.
- **Saloon Shootout** lets you blast mugs, plink at mice, and plug playing cards in a game for shootists of all ages.
- **Prisonball** puts you in a world where bricks and balls come and go, furiously.
- **Q-Bird** makes you responsible for a defenseless baby bluebird who's the entree for a flock of hungry predators.
- **Bump-N-Run** may make friends into enemies as you and an opponent catch, steal, and shoot a ball.
- **Ringside Karate** has everything you need to kick, jab, punch, and block your way to a black belt.
- **Eagles and Gators** is a constant struggle between eagles and alligators with 16 sprites and clever animation.

All seven games are ready to type in with our machine language entry program, "MLX." You don't have to know anything about machine language.

But if you're interested in—or already know how to program in—ML, *More Machine Language Games* offers the complete source code for each game, just as it was written by the original programmer. You can learn a great deal by studying these exceptional programs.

We've even included a complete (and fast) machine language *assembler*—a program that helps you write in machine language.

Seven fast-action arcade games. Illuminating source code. An assembler. If you like games, like to program in ML, or both, *COMPUTE!'s More Machine Language Games for the Commodore 64* is the book for you.

All games run on the Commodore 64 and Commodore 128 (in 64 mode).

All the programs in this book are also available on a companion disk. See coupon in the back for details.

ISBN 0-87455-094-7