Inter-Office Memorandum

To Mesa Users

Date October 24, 1977

From R. Johnsson

Location Palo Alto

Subject Simple Display Package

Organization SDD/SD

XEROX

Filed on: [MAXC1]<Johnsson>SimpleDisplay.bravo

This memo describes a simplified display package which may be of interest to some Mesa users. The package uses the DCB-per-line mode of operation in order to increase scroll speed and maximize the amount of text that can be displayed for a given amount of memory. There is no provision for other than Teletype style useage, i.e., there is no support for pointing, selecting, etc. This package can be used to replace the Display, Menus, Rectangles, and Windows modules in the standard Mesa system for applications which do not require the generality of those interfaces.

The modules described here may be used separately or may be used in the configuration SimpleDisplay which includes a standard control module.

Fonts

FontDefs defines the interface to Font objects.

```
FontDefs: DEFINITIONS =
    BEGIN

    BitmapState: TYPE = RECORD [
        origin: POINTER,
        wordsPerLine, x, y: CARDINAL];

    FontObject: TYPE = RECORD [
        paintChar: PROCEDURE [FontHandle, CHARACTER, POINTER TO BitmapState],
        clearChar: PROCEDURE [FontHandle, CHARACTER, POINTER TO BitmapState],
        charWidth: PROCEDURE [FontHandle, CHARACTER] RETURNS [CARDINAL],
        charHeight: PROCEDURE [FontHandle, CHARACTER] RETURNS [CARDINAL],
        close: PROCEDURE [FontHandle],
        destroy: PROCEDURE [FontHandle],
        lock: PROCEDURE [FontHandle] RETURNS [POINTER],
        unlock: PROCEDURE [FontHandle]];

    FontHandle: TYPE = POINTER TO FontObject;

    CharWidth: PROCEDURE [font: FontHandle, char: CHARACTER] RETURNS [CARDINAL];

    CharHeight: PROCEDURE [font: FontHandle, char: CHARACTER] RETURNS [CARDINAL];

    CreateFont: PROCEDURE [SegmentDefs.FileSegmentHandle] RETURNS
        [FontHandle];
```

```
END.
```

A FontObject contains the following operations:

paintChar: copies the specified character from the font to the bitmap position specified in the BitmapState; x is updated to point to the next character position.

clearChar: erases the rectangle containing the character (i.e. it does not just clear the bits of the character). The input state points just beyond the character and is modified to point to where the character used to be. paintChar[f, c, s] followed by clearChar[f, c, s] leaves s unchanged.

charWidth, charHeight: return the width and height of a character in bits and scan lines respectively.

close: swaps the font out of memory if it is not otherwise in use. The font will always be swapped in when needed, but may be swapped out at any time. It is not generally locked.

destroy: destroys the FontObject.

lock: locks the font in memory and returns a POINTER to the first word. This can be used to implement other operations on the bits in the font. Note that nothing in the FontObject dictates what font format is used.

unlock: undoes lock.

The module **AlFont** implements FontObjects for Al format fonts. Other modules for other font formats could be substituted easily. At this time no other modules have been written.

Display

The module **SystemDisplay** implements the following PROCEDURES from StreamDefs:

GetDefaultDisplayStream
ClearCurrentLine
ClearDisplayChar

DisplayDefs defines some additional interface procedures:

```
DisplayDefs: DEFINITIONS =
    BEGIN

    Background: TYPE = {white, black};

    SetSystemDisplaySize: PROCEDURE [nTextLines, nPages: CARDINAL];

    SetDummyDisplaySize: PROCEDURE [nScanLines: CARDINAL];

    InitDisplay: PROCEDURE [dummySize, textLines, nPages: CARDINAL, f: FontDefs.FontHandle];

    DisplayOff: PROCEDURE [color: Background];

    DisplayOn: PROCEDURE;
```

```
END.
```

The display consists of a dummy DCB for spacing followed by the display area itself. InitDisplay specifies the size of the dummy in scan lines (72 per inch), the size of the text area in text lines, and the number of pages of memory to allocate for bitmap space. A small display of 4-6 text lines works well with about 6 pages of bitmap space; a full screen of about 50 lines of program text (lots of white space) can be handled with about 40 pages of bitmap.

Once the display is initialized the size of either the dummy or the text area can be changed (changing the size to zero works correctly). DisplayOff and DisplayOn provide a simple means of reclaiming all of the display space. In addition DisplayOff calls font.close[font] to swap out the font if possible.

SystemDisplay also contains the following PUBLIC items which are not in the DisplayDefs interface (due to an oversight at the time the interface was frozen):

```
SetTypescript: PROCEDURE [StreamDefs.DiskHandle];
SetFont: PROCEDURE [FontDefs.FontHandle];
```

Call SetTypescript with a DiskHandle to enable the typescript facility; pass NIL to turn the typescript off. Call SetFont with a new FontHandle to change the font (use this procedure only when the display is off, i.e. size = 0).

Control Module

A standard control module, DisplayControl, is also provided. It will take care of initializing the display, font and typescript using the same algorithm as the standard Mesa system (use MesaFont.al or SysFont.al and Mesa.Typescript). It will also reestablish the font and typescript correctly after a MakeImage and flush the typescript buffer during OutLd-InId; i.e. it does the things the standard Mesa system does. The configuration to make a stand alone display package is:

```
SimpleDisplay: CONFIGURATION
    IMPORTS DirectoryDefs, ImageDefs, SegmentDefs, StreamDefs, StringDefs, SystemDefs
    EXPORTS DisplayDefs, FontDefs
    CONTROL DisplayControl =
    BEGIN
    AlFont;
    SystemDisplay;
    DisplayControl;
    END.
```

The source and object files for FontDefs, AlFont, DisplayDefs, SystemDisplay and SimpleDisplay are on the <MESA> directory.