Programmation des Systèmes

TP 4 : Pong

---

1 But

Le but de ce projet est de vous faire programmer une version simplifiée du jeu Pong. Il faudra programmer ce jeu entièrement en assembleur. Vous pourrez vous aider des procédures que vous avez programmées lors du TP 3.

Qu'est-ce que Pong ? Une petite « boule », se déplace à travers l'écran, rebondissant sur les rebords du haut et du bas, et les deux joueurs commandent chacun un « paddle », qui glisse de haut en bas à travers les extrémités de l'écran. Si la boule frappe la raquette, elle rebondit vers l'autre joueur. Si elle manque la raquette, l'autre joueur marque un point. La « boule » rebondit de différentes manières selon la façon dont elle touche la raquette.

2 Travail à faire

Dans ce TP, nous vous demanderons de programmer une variante de Pong :

- Le terrain de jeu est constitué de trois bords, en haut, à gauche et à droite (il s'agit de lignes).
- Il y a une boule qui rebondit sur les bords : La boule est un cercle vide (utilisez la fonction `drawCircle` du TP précédent). Elle doit être de couleur blanche.
- Une raquette est commandée par un joueur, elle se trouve en bas de l'écran et peut se déplacer à gauche et à droite. La raquette se présente sous la forme d'un rectangle.
- Si la raquette manque la balle, alors le jeu est terminé. Vous pouvez faire une animation si vous voulez. Pour recommencer la partie, il suffit d'appuyer sur `start` (Il s'agit de la touche `enter` sur l'émulateur).
- Le rebond consiste en une inversion du déplacement.
- Trouvez un algorithme qui n'efface pas tout l'écran entre deux affichages.

Ce TP devra être programmé **entièrement** en assembleur.

3 Indications

- Les valeurs des boutons se trouvent à l'adresse : `0x04000130`. Chaque bouton est codé sur un bit : 1 si le bouton est pressé, 0 sinon. Voici la position de chaque bouton :

Bouton	Bit
A	1
B	2
SELECT	3
START	4
RIGHT	5
LEFT	6
UP	7
DOWN	8
R	9
L	10

Exemple : Si le troisième bit vaut 3, alors le bouton `SELECT` est pressé.

– Pour créer un projet seulement en assembleur :

1. Créez un nouveau projet en C.
2. Supprimez le fichier `main.c` et ajoutez un fichier assembleur au projet.
3. Dans le fichier makefile, remplacez la ligne

   ```
   OFILES += main.o #nouveau_fichier.o#
   ```

   par

   ```
   OFILES += #nouveau_fichier.o#
   ```

   où `main.o` est supprimé.
4. Voici le code minimum pour créer un programme en assembleur.

   ```
           .ARM
           .align
           .globl  main
   main:
   ```