TP6 Splashmem

Le code du tp de splashmem se trouve dans le fichier tp6.tar.gz

Le répertoire contient un dépôt git, pour le rendu faire un commit
```
$ git add *
$ git commit -m “rendu tp6”
```
Puis zipper le répertoire avec les fichiers. Attention à ce que le répertoire .git soit bien présent dans l’archive.
Envoyer le zip par mail à herve.chaminaud@univ-tours.fr

Si necessaire Installer la library SDL 2 pour compiler splashmem
```
$ sudo apt install libsdl2-dev
```

Exercice 1 (6 points)

Réaliser un programme qui charge une bibliothèque dynamique.

http://tldp.org/HOWTO/Program-Library-HOWTO/dl-libraries.html

Ecrire dans un fichier `ex1.c` une fonction `main` qui charge un fichier bibliothèque dynamique.
Vous devez utilisez les fonctions `dlopen` qui va ouvrir la bibliothèque, la fonction `dlsym` qui permettra de récupérer un pointeur de fonction sur la fonction qui nous intéresse, ici `get_actions.`

Reprendre l’exemple dans le chapitre 4.5 du lien précédent et l'adapter pour charger un des fichiers player (.so) du dernier TP. (des fichiers sont fournis avec le code du TP6)

Ces fonctions se trouve dans la libdl.so déjà installée sur le sytème linux, vous devez prendre cela en compte lors de la compilation/linkage

---

Exercice 2 (6 points)

Modification du moteur de jeu

Reprendre le l’exercice précédent pour implémenter la fonction load_players dans le moteur de jeu (fichier loader_player.c)

Le prototype:
`void load_players(int argc, char *argv[])`

La fonction `load_player()` doit charger pour chaque joueur le champ `so_handle` et remplir le champ pointeur de fonction de `get_action` du joueur correspondant.

Les fichiers .so sont passés en paramètre et donc doivent être récupéré dans la variable argv.

Exercice 3 (8 points)

Coder les actions du moteur de jeux
Fichier actions.c.
Compléter toutes les fonctions vides, plus la fonction actions_init() qui initialise le tableau de pointeur de fonction.

Annexe

Création d'un jeu "multijoueur", 4 joueurs max. Chaque joueur est un programme dont l'objectif est de remplir des cases mémoires. Le programme/joueur qui aura rempli le plus de case gagne la partie.

Chaque programme-joueur a un credit d'action de 9000.

Code	**Cout**	**Action**
ACTION_MOVE_L, ACTION_MOVE_R, ACTION_MOVE_U, ACTION_MOVE_D,	1	Déplace le programme joueur d'une case
ACTION_DASH_L, ACTION_DASH_R, ACTION_DASH_U, ACTION_DASH_D,	10	Déplace le programme joueur de 8 case dans une direction donnée
ACTION_TELEPORT_L, ACTION_TELEPORT_R, ACTION_TELEPORT_U, ACTION_TELEPORT_D,	4	Téléporte le joueur de 8 case dans une direction
ACTION_SPLASH	8	Marque toute les cases autour du joueur
ACTION_BOMB	9	Place une bombe qui

		marque 9 cases et qui ce déclenche après 5 tours
ACTION_STILL	1	Pas d'action

Dès qu'un joueur arrive sur une case, celle-ci est automatiquement marquée.

La taille du plateau de jeux est de 100 cases par 100 cases.
Chaque joueur commence à la même distance
Lorsque qu'un joueur dépasse un bord il est renvoyé sur le côté opposé.

Chaque programme-joueur a des coordonnées x,y, la coordonnée (0,0) et en haut à gauche de la zone de jeu.