POLYTECH SORBONNE

Sorbonne Université
2021-2022

Cours de Programmation en C – EI-SE

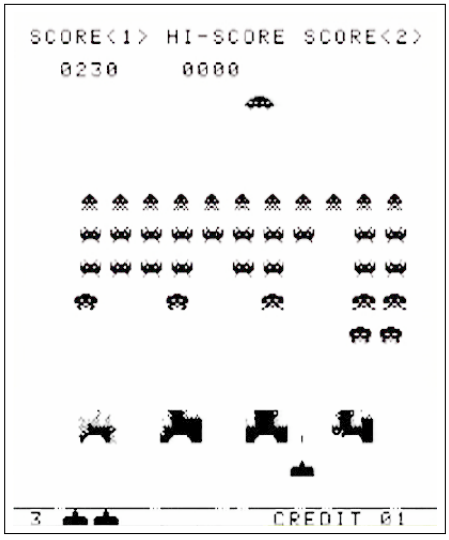
Travaux Pratiques

Objectif(s)

- ★ Définition de nouveaux types
- ★ Structures de données
- ★ Unions et enum

Objectif(s)

- ★ Structures
- ★ Tableaux statiques et fonctions
- ★ Pointeurs



Le but de ce TP est de réaliser un petit clone du jeu *Space Invaders*, sorti sur borne arcade en 1978[1]. Une petite bibliothèque graphique simplifiée va vous permettre d'afficher différents objets (appelé *sprites*) et de les animer. On ne pourra pas réaliser un jeu complet, mais déjà les 1ers éléments.

Tout fichier source doit **obligatoirement** comporter comme entête un long commentaire avec le nom de l'auteur, son groupe, la date et le numéro de TP ainsi que le sujet du TP et une explication sur le contenu du fichier.

Vous devrez utiliser le principe de la compilation séparée. Il vous faudra indiquer en commentaire de chacun de vos fichiers .c la ligne de commande utilisée pour la compilation, ainsi que la ligne de commande utilisée pour l'édition de liens.

Question 1 – Affichage du décor et du vaisseau

Dans , récupérez les fichiers suivants :

- `affichage.h` : fichier en-tête fournissant les prototypes des fonctions d'affichage
- `affichage.o` : fichier objet (déjà compilé) pour les fonctions d'affichage

Il est *indispensable* de lire attentivement les fonctions dans le fichier en-tête (fichiers `.h`), car les commentaires constituent la référence pour leur utilisation.

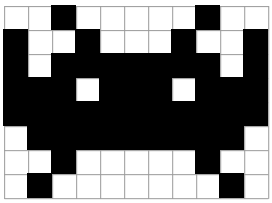
L'affichage concerne tous les motifs graphiques (vaisseau spatial, ennemis, ...) qui seront appelés des *sprites*[2]. Ici, on n'affichera que des sprites en une seule couleur, et on les représente par un tableau (le type `t_sprite`) de taille définie (11×8 ici, pour reprendre les sprites du jeu originel). Ce tableau ne comporte que des 0 et des 1.

Par exemple, un alien sera représenté par le tableau suivant

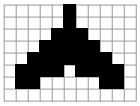
[1] Ce jeu est le 1er *shoot them up* de l'histoire des jeux vidéos : `http://fr.wikipedia.org/wiki/Space_Invaders`

[2] Un sprite (en français lutin) est dans le jeu vidéo un élément graphique qui peut se déplacer sur l'écran. En principe, un sprite est en partie transparent, et il peut être animé (c'est-à-dire qu'il est formé de plusieurs bitmaps qui s'affichent les uns après les autres). Le fond de l'écran constitue généralement le décor et les sprites sont les personnages et les objets qui se superposent au fond d'écran et qui se déplacent.

```
t_sprite alien = {
{0,0,1,0,0,0,0,0,1,0,0},
{1,0,0,1,0,0,0,1,0,0,1},
{1,0,1,1,1,1,1,1,1,0,1},
{1,1,1,0,1,1,1,0,1,1,1},
{1,1,1,1,1,1,1,1,1,1,1},
{0,1,1,1,1,1,1,1,1,1,0},
{0,0,1,0,0,0,0,0,1,0,0},
{0,1,0,0,0,0,0,0,0,1,0}};
```



On se limitera aux sprites d'une seule couleur (comme la version originelle) à choisir parmi les constantes définies dans `affichage.h`.

1. Créer un sprite correspondant au vaisseau. Par exemple, cela pourra être quelque chose qui ressemble à 
2. Écrire un fichier `main.c` permettant d'afficher une fenêtre graphique (par exemple avec une largeur de 400 pixels et une hauteur de 600 pixels)
3. Afficher un vaisseau rouge (par exemple au milieu en bas)

Pour cela, l'affichage se fera :

- en appelant la fonction `initAffichage`
- en répétant dans une boucle (dans cet ordre!)
  - calculs des positions
  - affichage des différents sprites (avec)
  - mise à jour de l'affichage, avec la fonction `miseAJourAffichage` (on prendra une temporisation entre 50 et 200μs)

On veillera à ce que le programme quitte lorsque l'utilisateur a pressé la touche ESC.

Question 2 – Déplacement du vaisseau

Modifier le programme précédent pour que le vaisseau se déplace de gauche à droite quand on appuie sur les flèches. Attention, le vaisseau ne doit pas sortir de la fenêtre.

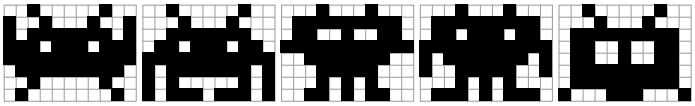
Question 3 – Affichage et déplacement des aliens

On considère maintenant les aliens. Les fonctions associées seront placées dans un fichier `alien.c` (et leurs prototypes dans `alien.h`).
On veut créer 3 rangées de 5 aliens (chaque rangée à une couleur différente), qui se déplacent horizontalement (de gauche à droite puis de droite à gauche), et verticalement (quelques pixels vers le bas) à chaque fois qu'ils ont atteint le bord droit. Pour cela, on créé une structure `t_alien` contenant une position et une couleur.

1. Créer la structure dans le fichier `alien.h`, et créer un tableau de 15 aliens dans `main.c`
2. Initialiser les positions et couleurs de ces aliens.
3. Créer une fonction `deplaceAlien` dans `alien.c`. Cette fonction devra modifier un alien passé en paramètre pour le **déplacer**.
   Puisque les aliens doivent se déplacer de gauche à droite puis de droite à gauche et aussi verticalement, rajouter un champ `compteur` à la structure. À chaque déplacement, on incrémentera cette valeur, que l'on rebouclera à 0 lorsqu'elle aura atteint 50 (pensez au modulo!). Lorsque `compteur` est inférieur à 25, on déplace les aliens vers la gauche de quelques pixels, et sinon on les déplace vers la droite. Lorsque `compteur` atteint 50 uniquement, on déplace les aliens vers le bas (de quelques pixels).
   Pour chaque alien, appelez cette fonction dans la boucle principale du `main`.
4. Puis afficher chaque alien (dans la boucle du `main`).

Question 4 – Animation des aliens

1. Créer de nouveaux sprites pour les aliens, en vous inspirant des modèles suivants, et utilisez-les pour varier les aliens (un type par ligne, par exemple).



2. Modifier la structure `t_alien` pour y stocker un sprite. Attention, on ne stocke pas un sprite **mais un pointeur vers un `t_sprite` déjà créé**.

3. Écrire une fonction `afficheAlien` dans `alien.c`. Faire appel à cette fonction dans `main.c` pour l'affichage.

4. (Question bonus) En utilisant des sprites légèrement différents que l'on alterne à chaque itération, créez une animation des aliens.

Question 5 – Tir

Il faut désormais pouvoir abattre les aliens.

1. Créer une structure (dans un fichier `vaisseau.h`) pour décrire un vaisseau (uniquement la position pour l'instant). Créer une fonction `deplaceVaisseau` (dans `vaisseau.c`) qui reprend le code utilisé pour **déplacer** le vaisseau en fonction des touches, et qui l'**affiche**

2. Rajouter les coordonnées d'un missile dans cette structure, et éventuellement un booléen qui indique si un missile a été tiré ou non. Complétez la fonction `deplaceVaisseau` pour qu'elle déplace aussi un missile s'il a été tiré (appui sur la touche espace).
   Attention, un seul missile à la fois ne peut être tiré.

3. Écriver une fonction `testMissile` qui teste si le missile touche un alien ou non. Cette fonction renvoie le numéro de l'alien, ou -1 si il n'y a pas de collision.

4. Utiliser cette fonction dans le `main` et modifier les fonctions `deplaceAlien` et `afficheAlien` ainsi que la structure `t_alien` pour qu'un alien touché ne s'affiche plus (et produise un éclair durant l'impact)

Question 6 – Bonus

1. Modifier encore les sources précédents pour permettre aux aliens d'envoyer des missiles (un alien envoie un missile tous les 150 itérations, par exemple). Gérez le déplacement de ce missile, ainsi que le contact possible avec le vaisseau. La partie doit s'arrêter si le vaisseau est touché, ou si les aliens sont descendus trop bas sans être tous abbatus.

2. Lorsque chaque vague de 15 aliens est abattue, une nouvelle vague d'aliens doit arriver, en se déplaçant plus vite.

3. Dans le jeu originel, le vaisseau peut se cacher derrière des obstacles. Installer ces obstacles. Les tirs des aliens (mais aussi du vaisseau) peuvent percer ces obstacles.