



SPACE RACER

L'ultime simulateur spatial

Frisson et aventure
pour toute la famille



50Fr

[HLIN202] Programmation impérative - CC no 2

Avril 2018

Durée : 1 heure 30 - examen machine - Documents : 1 feuille de notes

I Introduction

I.1 Consignes de l'examen

Téléchargez le fichier zip à l'adresse suivant

`https://www.lirmm.fr/~suro/examHLIN202/HLIN202-SujetCC2.zip`

Les fichiers à importants rendre seront SpaceRacer.cpp et SpaceRacer2.cpp pour la partie 2. Renvoyez le dossier complet avec votre code sous la forme d'un fichier zip nommé "HLIN202-votre nom.zip", à l'adresse suivante :

`suro@lirmm.fr`

Attendez que je vous confirme la bonne réception avant de quitter la salle.

Le contenu de votre dossier de travail doit être:

- SpaceRacer.cpp : le squelette du programme à compléter
- SuperVideoGameLibrary.a SuperVideoGameLibrary.h : librairie de fonctions à utiliser pour compléter le jeu
- Fonctions_Je_passe.a Fonctions_Je_passe.h : librairie de fonctions à utiliser si vous êtes bloqués
- SR : une version compilé du jeu que vous devez obtenir
- HLIN202-CC2.pdf : le sujet en pdf

Pour la première partie de l'examen (SpaceRacer.cpp), vous n'avez pas le droit de changer la signature des fonctions.

Si vous n'arrivez pas à compléter une fonction dont vous avez besoin plus tard (par exemple si vous n'arrivez pas à compléter *allouerEcran*, et que voulez tester *initialiserEcran*), vous pouvez utiliser les fonction de la librairie *Fonctions_Je_Passe.a* définies dans le fichier header. Par exemple vous pouvez remplacer *allouerEcran(...)* par *FJP_allouerEcran(...)*. Bien entendu, seul les fonctions que vous aurez définies vous même seront notés.

Pour compiler votre programme avec les librairies *Fonctions_Je_passe.a* et *SuperVideoGameLibrary.a*, la commande g++ est :

```
g++ SpaceRacer.cpp SuperVideoGameLibrary.a Fonctions_Je_Passe.a -o SpaceRacer
```

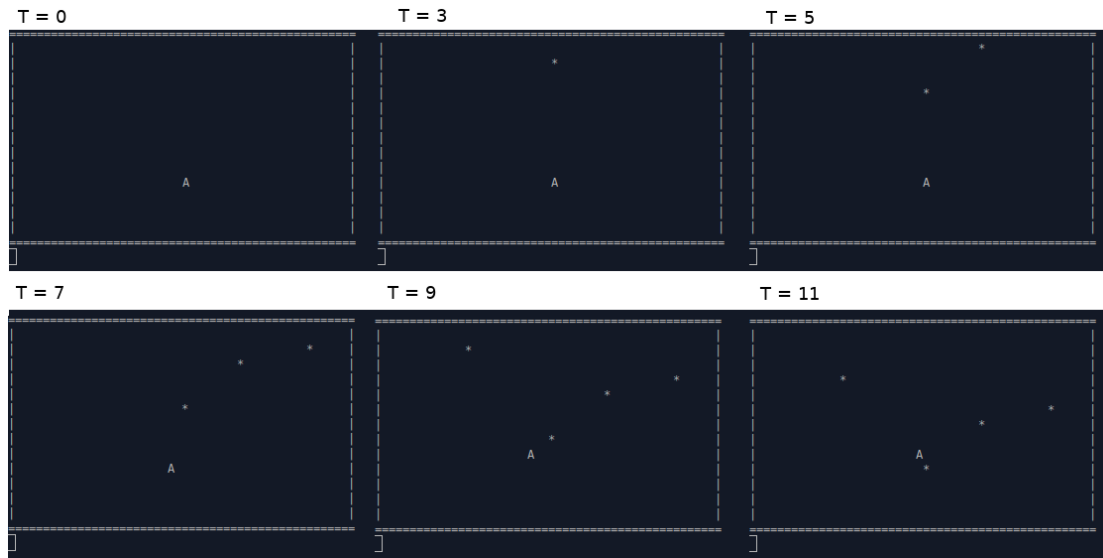
I.2 Situation

Suite à la sortie de l'épisode V de Star Wars, L'Empire contre-attaque, et à la nouvelle popularité des univers de science-fiction, la compagnie de jeux vidéo pour laquelle vous travaillez a décidé de publier un jeu de vaisseaux spatiaux.

Dans ce jeu , le joueur contrôle un vaisseau à l'aide du clavier et doit éviter les astéroïdes qui défilent verticalement.

L'écran de jeu est un tableau dynamique de caractères qui est affiché successivement sur la console.

Voici quelques images concept réalisé par le game designer.



T=0 montre l'état initial. chaque T successif montre le défilement. à T=7 le joueur commence à déplacer le vaisseau (vers la gauche).

Pour faciliter votre travail, votre compagnie à acheté une licence de la bibliothèque "Super-VideoGameLibrary" dont la documentation et donnée en annexe.

II Fonctions : 10 points

II.1 allouerEcran

Cette fonction retourne un tableau à deux dimensions de caractères, alloué dynamiquement, représentant l'écran de jeu défini par la résolution en X (horizontale) et Y (verticale) représentant respectivement le nombre de colonnes et le nombre de lignes de la console.

II.2 initialiserEcran

Cette fonction initialise l'écran en remplissant le tableau 2D de caractères : elle trace la bordure de l'écran (par exemple les caractères '=' et 'I') et prépare un environnement vide (par exemple le caractère ' ').

II.3 afficherEcran

Cette fonction affiche le contenu de l'écran sur la console ainsi que le vaisseau du joueur (par exemple, le caractère 'A') aux coordonnées passés en paramètre. Chaque ligne du tableau 2D est affiché comme une ligne sur la console.

ATTENTION : utilisez la bibliothèque "SuperVideoGameLibrary" pour effacer l'écran avant chaque affichage.

II.4 genererObstacles

Cette fonction renvoie une chaîne de caractères (un tableau 1D de char), dont la longueur est passée en paramètre, allouée dynamiquement, et qui commence et termine par le symbole de bordure. Entre ces bordures vous devez placer aléatoirement des caractères représentant des obstacle et des espace vides. (Cette chaîne représente une ligne qui sera placé en haut l'écran à chaque nouvelle étape du défilement)

ATTENTION : utilisez la bibliothèque "SuperVideoGameLibrary" pour générer aléatoirement des obstacles.

exemple de sortie :

```
|  *    **   *   *           * *   *           * |
```

II.5 defilerEcran

Cette fonction déplace chaque ligne de l'écran d'une ligne vers le bas et ajoute la nouvelle ligne en haut de l'écran. Attention à la bordure en haut et en bas de l'écran. Attention à bien libérer les lignes (allouées dynamiquement) que vous n'utilisez plus afin de ne pas causer de fuites de mémoire !

II.6 verifierGameOver

Cette fonction vérifie que le vaisseau ne s'est pas crashé dans un obstacle.

II.7 recupereCommandeJoueur

Cette fonction récupère les commandes du joueur saisies au clavier, modifie la position du vaisseau et renvoie True si l'utilisateur souhaite quitter la partie. la liste des commandes possible est :

- q : déplacer le vaisseau à gauche
- d : déplacer le vaisseau à droite
- z : déplacer le vaisseau en haut
- s : déplacer le vaisseau en bas
- t : quitter la partie

ATTENTION : utilisez la bibliothèque "SuperVideoGameLibrary", pour récupérer la touche du clavier que le joueur a appuyé, de manière non bloquante (c'est à dire que le programme renverra un caractère par défaut si aucune touche n'est appuyé au moment de l'appel, au lieu de se bloquer pour attendre)

II.8 recupereRecomencer

Cette fonction affiche le message de gameover et demande au joueur si il souhaite rejouer. le message est affiché tant que le joueur n'a pas entré une réponse valide (o/n , y/n, oui/non ...).
par exemple :

```
GAME OVER  
voulez vous rejouer ? (O/N)
```

III La boucle de jeu : 5 points

En utilisant les fonctions définies précédemment, complétez la fonction *demarrerSpaceRacer* et créez le jeu "Space Racer". (vous êtes libre de remplacer complètement le squelette de la fonction, veuillez cependant à bien réguler la vitesse du jeu ...)

IV Space Racer II : 5 points

Le jeu Space Racer a été très bien reçu par le public, et votre compagnie a décidé de financer une suite, Space Racer II, en vous laissant plus de liberté créative. Cependant votre compagnie a décroché un contrat avec un grand fabricant de machines d'arcades, contrat honoré à la condition que vous intégriez un système de score à votre jeu.

IV.1 Consigne

faites une copie du fichier SpaceRacer.cpp que vous renommerez SpaceRacer2.cpp. Vous apporterez les modifications suivantes dans le fichier SpaceRacer2.cpp (ne modifiez surtout pas SpaceRacer.cpp). Vous êtes libre de modifier tout ce que vous souhaitez, y compris les signatures des fonctions.

IV.2 Scoring

Ajoutez un système de score, qui comprend tout les affichages , la comptabilisation nécessaire. La règle de calcul du score est laissée à votre discrétion , à condition qu'une meilleure performance conduise à un meilleur score.

IV.3 GOTY

Ajoutez tout ce que vous souhaitez au jeu qui en fera à coup sûr le jeu de l'année 1981 !!

quelques exemples de fonctionnalités (par ordre de complexité):

- Des objets à ramasser augmentent le score.
- Des objets à ramasser donnent des bonus (vitesse, invulnérable...).
- En fonction du score/temps écoulé , la vitesse du jeu augmente, le nombre d'astéroïdes augmente ...
- Les astéroïdes ont des vitesses différentes.
- la génération d'astéroïdes suit certains motifs (groupés).
- Les astéroïdes peuvent se déplacer dans des trajectoires différentes.
- Le vaisseau peut tirer et détruire des astéroïdes.
- Des vaisseaux ennemis apparaissent, ils tirent des projectiles!!.
- moteur de rendu 3D, massivement multi-joueur, cinématiques avec des acteurs d'hollywood ...
- ...



Super Video game Library

REFERENCE MANUAL

Fonction:

`char nonBlockingKeyInput()`

Description:

Retourne le caractère correspondant à la touche appuyé au moment de l'appel. Fonction non bloquante qui retourne le caractère '0' si aucune touche n'est appuyé au moment de l'appel.

Paramètres:

aucun.

Fonction:

`void sleepFunction(int microsec)`

Description:

Interrompt l'exécution du programme pendant un temps donné.

Paramètres:

`int microsec` : la durée de l'interruption en micro secondes.

Fonction:

`bool randomBooleanGeneration(double probability)`

Description:

Retourne True aléatoirement, selon la probabilité passé en parametre.

Paramètres:

`double probability` : probabilité de retourner True, entre 0.0 et 1.0 .

Fonction:

`void clearScreen()`

Description:

Efface tout ce qui est affiché sur la console.

Paramètres:

aucun.

