

Master de Science, Mention Scientifique
Sciences et Ingénierie du Réseaux, de l'Internet et des Systèmes

Programmation Avancée

Projet - Arkanoid

Rapport

De :

Govindaraj VETRIVEL

Binôme :

Constantin DIVRIOTIS

1. Présentation

Pour le projet de l'UE Programmation Avancée, nous avons implémenté une version du jeu "Arkanoid". J'ai fait ce projet avec Constantin DIVRIOTIS, aussi en Master SIRIS.

Nous avons utilisé l'outil git, ainsi que l'outil cmake comme dans l'exemple donnée par l'enseignant de l'UE.

Pour utiliser le programme, il faut exécuter les commandes suivantes dans un terminal :

- cmake CMakeList.txt
- make
- ./Arkanoid.

Je vous ai envoyé une invitation pour avoir accès au projet sur le git de l'unistra.

2. Travail effectué

Je me suis d'abord occupé de l'affichage de tous les éléments du jeu, tous les objets visibles dans le jeu ont une méthode view() qui utilise SDL_blitSurface.

Je me suis ensuite occupé des collisions des balles avec d'une part les vaisseaux (1 ou 2 selon le mode choisi) et les rebords de la fenêtre de jeu, ainsi que la collision entre vau et bonus. Toutes les classes d'objets visibles héritent de la classe MObject, qui implémente une fonction de collision entre l'objet lui même et un autre MObject.

J'ai ajouté les fonctionnalités liés aux mouvements contrôlés des vaisseaux ainsi que certains raccourcis pour les tests (retour au menu ou passage au niveau suivant).

J'ai aussi implémenté certains bonus dont le bonus C "Catch and Fire".

J'ai dû utiliser l'objet "Vector" avec des shared_ptr pour les objets qui peuvent être en plusieurs exemplaires (balles, vau, bonus). Ceci m'a permis de ne pas m'occuper de la mémoire et gagner du temps.

J'ai implémenté un objet pour l'affichage de mots et/ou chiffres, en général je n'utilise pas de "#define" pour chaque image que je veux utiliser. Si les images sont adjacentes et de même nature (briques, lettres, vau), j'incrémente seulement les valeurs des coordonnées pour me placer au bon endroit sur la surface du sprite.

Enfin, j'ai ajouté un mode multijoueur, qui est un mode de coopération, j'aurai voulu en ajouter plus mais par manque de temps, je n'ai pas pu y arriver.

3. Difficultés et améliorations

J'ai conscience que nous aurions pu rendre le programme plus modulaire et extensible en créant plus de classes et en ajoutant certaines fonctionnalités. Par exemple, la gestion des événements SDL se fait à travers une seule fonction, nous aurions dû créer une classe pour les événements : un "events manager".

Il en va de même pour l'affichage SDL, j'ai dû créer un sprite manager, et bien encapsuler les fonctions SDL. Nous avons privilégié l'implémentation du jeu au reste.

Il y a donc beaucoup de chose à améliorer. Par exemple, laisser à l'utilisateur le choix du fond d'écran, du niveau ou de créer un niveau, etc.