

Rapport de Projet: Programmation impérative

Marwan Ait Addi, Gustav Hubert

Décembre 2019

1 Présentation du jeu

Le jeu que nous avons choisi de réaliser s'appelle Joust, c'est un jeu d'arcade de 1982 où le joueur incarne un chevalier qui se bas a dos d'oiseau il doit jouter contre ses ennemis afin de les vaincre et d'engranger des points. Dans notre version on peut jouer seul ou a deux contre des vagues d'ennemis de plus en plus nombreux et forts, il existe deux type d'ennemis les passifs et les agressifs. Les ennemis passifs ne font que se balader sans but, tandis que les agressifs vont activement essayer de tuer les joueurs.

Afin de gagner une joute contre un ennemi il faut que l'on soit au dessus de lui quand on le touche sinon c'est lui qui gagne et qui nous tue, on peut également avoir une égalité dans ce cas les deux combattants se repoussent dans des directions opposées. Lorsque l'on tue un ennemi il se transforme en oeuf, si l'oeuf n'est pas ramassé il éclos au bout d'un certain temps faisant réapparaitre un ennemi.

On gagne des points en tuant les ennemis et en ramassant des oeufs. Lorsque le joueur meurt il peut inscrire son nom afin d'apparaitre dans la liste des meilleurs scores.

2 Choix d'implémentation

Nous utilisons quatre modules pour faire fonctionner notre jeu :

- Objects : fonction d'importation des objets depuis un fichier
- Screen : fonctions d'affichage du jeu, du menu et des scores
- Controls : Gère le déplacement des joueurs et la physique du jeu
- IA : Gère l'intelligence artificielle des deux types d'ennemis

Nous avons choisis d'utiliser autant de modules afin de faciliter le travail en équipe et d'améliorer la lisibilité du code. tout les modules ainsi que le main utilisent le module Object qui définit toutes les structures. Nous avons utilisé des structures afin de définir les différents objets du jeu, chaque objet appartient a un type d'objet auquel est attaché un sprite ainsi que d'autres valeurs générales puis, uniquement pour les oiseaux, un type d'oiseau qui définit si l'oiseau est un joueur, un ennemi, sa vitesse, sa direction etc. Chaque type d'objet (oiseaux, plateforme) est listé dans une liste (liste d'oiseaux, liste de plateforme, liste de Types d'oiseaux) afin de pouvoir être transmis et modifié aisément partout dans le programme.

3 Démarche et difficultés

Afin de travailler efficacement et d'avoir un historique des modifications effectuées nous utilisons git et gitHub. Ceci nous a permis plusieurs fois de récupérer notre travail et également de tester de nouvelles choses sans s'inquiéter du fait que cela pourrait détruire notre projet.

Au début il était compliqué de choisir par quoi commencer car il y avait beaucoup à faire, cependant une fois l'affichage du jeu au point le fait de créer et tester de nouvelles fonctionnalités est devenu beaucoup plus facile. La physique du jeu était l'une des parties les plus compliquées notamment les collisions, les bugs étaient nombreux et pendant longtemps il était dur d'empêcher les oiseaux de traverser les plateformes lorsqu'ils allaient trop vite. D'autres nombreux problèmes ont également eu lieu mais il serait trop long de les résumer dans ce court rapport.

Afin de se répartir les tâches nous avons mis en place une liste des fonctionnalités et bugs sur lesquels il fallait travailler, triés par ordre d'importance. Ainsi nous pouvions assez simplement choisir sur quoi travailler et nous répartir les tâches ainsi qu'ajouter d'autres tâches. Avoir une liste précise de ce qui doit être fait nous permet de mieux nous organiser et évite d'être démotivé devant la charge de travail.