

Programmation Jeux vidéo – IFT 2103

Présentation de l'exercice :

Dans le cadre du module IFT 2103, certains exercices sont à effectuer dans le but de parfaire nos connaissances du secteur du jeu vidéo. Parmi ceux disponibles, cette fiche présente le très célèbre jeu « Pong ». Pong est considéré à juste titre comme le premier jeu vidéo ayant vu le jour. D'après Wikipédia, Pong est un jeu vidéo inspiré du tennis de table développé par Ralph Baer et son équipe à Sanders Associates en 1967. On retrouve ainsi deux raquettes de chaque côté de l'écran avec une balle qui parcourt le terrain et qui se renvoie du côté opposé à chaque raquette quand celles-ci touchent la balle. Le premier qui ne réussit pas cela octroie un point à son adversaire. Au bout de sept points la partie se termine.

L'exercice est donc de reproduire ce jeu avec la technologie de notre choix.

Grâce aux avancées technologies d'aujourd'hui, un tel jeu n'est pas très compliqué à faire dans le fond. Mais on peut se servir de ce prétexte qu'est l'exercice pour s'exercer sur les différentes possibilités d'une technologie. C'est ce qui a motivé mon choix dans l'utilisation de Unity (moteur de jeu) car je travaille sur Unity dans le cadre d'un autre projet, mais étant plus porté sur la documentation je n'ai que peu travaillé sur celui-ci.

Aperçu du jeu :

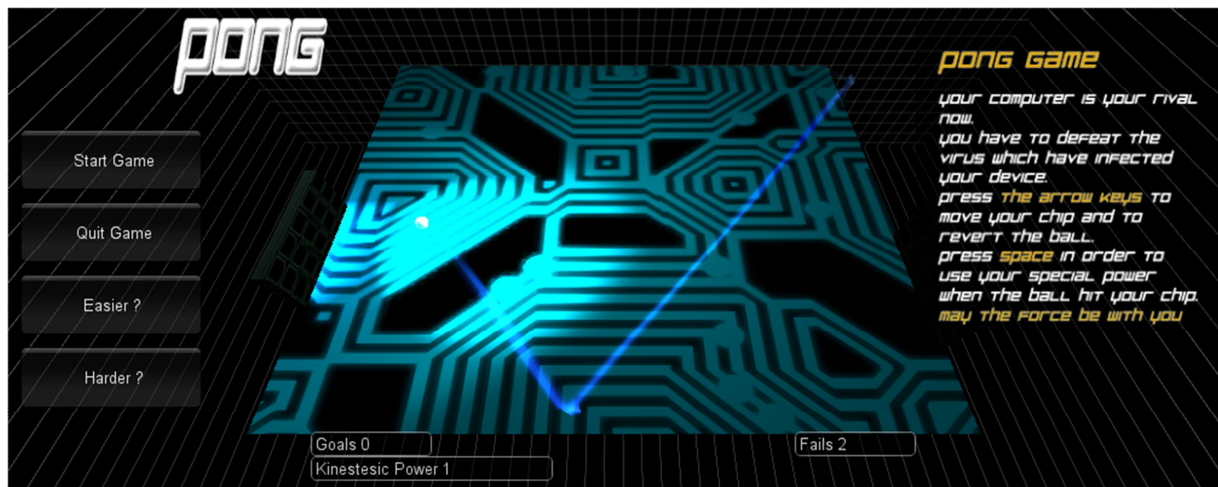


Figure 1 Aperçu du projet "Pong"

On peut voir ci-dessus un aperçu du jeu vidéo. Le visuel est assez particulier et rappelle l'univers de « Tron ». Sur le côté gauche, on aperçoit un succinct menu permettant de lancer le jeu et de le quitter (option non disponible pour la version Web) ainsi que de diminuer ou d'augmenter la difficulté du jeu en temps réel. La G.U.I (Graphic User Interface) est aussi agrémentée du compteur de points de chaque camp en bas de l'écran, ainsi que d'un compteur de pouvoir pour le joueur (Voir règles du Jeu). Et enfin, sur le bord droit de l'écran se trouve un petit résumé des règles du Jeu pour les néophytes et pour pouvoir expliquer le petit ajout comparé à la version originale.

Règles du jeu :

Le but du jeu est de marquer des points en faisant en sorte que l'adversaire n'arrive pas à renvoyer à temps la balle dans le camp opposé. Pour se faire, chaque joueur ou ordinateur peut déplacer sa raquette de manière verticale selon une vitesse déterminée par les touches « Haut » et « Bas ». Chaque collision avec le décor renvoie la balle avec le même angle d'incidence mais pas sur les raquettes car celles-ci peuvent modifier la trajectoire pour surprendre l'adversaire. Le joueur humain étant au-dessus de l'intelligence artificielle et possédant le statut de « Créateur » peut user d'un pouvoir d'accélération sur la balle. En appuyant sur « Espace » au moment même où la balle entre en contact avec la raquette du joueur, celui-ci insufflé une force à la balle pendant un court instant ce qui permet de surprendre l'adversaire sur le déplacement de la balle. Le premier des joueurs à atteindre le score de sept gagne la partie et la partie ainsi s'achève.

Architecture du jeu :

- Pong Scene
- Préfab
- Scripts
- Sounds
- Textures

La scène représente le plateau de jeu ainsi que tous les éléments affairant avec.

Les préfabs sont au nombre de deux : Explosion et PongBall. L'explosion est un objet qui est instancié lors de la destruction d'une balle (But), c'est une représentation d'une petite explosion à travers l'outil de particules de Unity. La PongBall est la balle du jeu avec toutes ses caractéristiques telles que les scripts associés ou bien sa position absolue.

Les scripts sont les morceaux de code qui régissent la scène. Ils sont écrits en C# et il en existe cinq :

1. AIScript (Gestion de la raquette adverse)
2. BallScript (Gestion des mouvements de la balle et des changements d'angle opérés)
3. GameEngine (Gestion de l'interface graphique complète, du score et du pouvoir spécial)
4. InputManager (Gestion de la raquette du joueur)
5. LookAtCameraScript (Gestion de la lumière qui suit la balle)

Les sons sont au nombre de deux. Il y a le son GameOver qui signale la fin du jeu et le son Rebound qui s'effectue lors du rebond de la balle.

Les textures sont nombreuses, elles sont toutes visibles à l'écran comme celle de la raquette, du terrain ou du logo du jeu. Elles ont été travaillées sur Photoshop.

Caractéristiques du jeu :

Ce jeu a été réalisé en environ dix heures, car plusieurs pistes ont été abordées dans sa réalisation notamment pour tout ce qui a trait à la gestion des collisions et des rebonds. Le jeu intègre tout de même une légère intelligence artificielle et quelques effets sonores et visuels intéressants facilités par Unity (Jeu de lumière, Trail, particules, AudioClip, ...).

Le déplacement des raquettes ainsi que celui de la balle est effectué par une translation de l'objet selon la vitesse paramétrée, le temps écoulé entre chaque « Frame » (Classe Time) et une valeur de déplacement arbitraire.

Les rebonds de la balle sont calculés dans lors d'une rencontre avec un trigger (zone événementielle), ainsi en prenant le vecteur directeur (forward) des deux éléments de la collision, on est à même de déterminer avec quelle angle touche la surface en question. Selon la direction et la zone rencontrée, la balle est renvoyée de façon à imiter un angle d'incidence.

Les rebonds effectués sur les raquettes sont par contre différents car ils sont coefficientés par rapport à l'endroit de collision sur la raquette. En d'autres termes, plus l'endroit où la balle touche la raquette est éloigné du centre de celle-ci, plus l'angle sera obtus. Ce choix a particulièrement été motivé pour agrémenter le Game Play d'une notion de stratégie de placement et casser le rythme imposé par le simple rebond.

L'intelligence artificielle du jeu est très basique car celle-ci s'évertue à suivre la balle dès que celle-ci rentre dans son camp. Cet artifice est réalisé par l'intermédiaire d'une translation de la raquette par rapport à la position de la balle.

Toute l'interface du jeu a été placée dans le GameEngine qui est un script accolé à la caméra. Ce script gère le score, le son, le pouvoir du joueur ainsi que les différents boutons utilisables.

Conclusion :

La réalisation de cet exercice m'a permis de me familiariser avec la manipulation de vecteurs car en parallèle, j'effectuais les calculs à la main selon les éléments de cours. Mais l'avantage est d'avoir pu toucher à plusieurs aspects de Unity comme la gestion des particules, du son ou bien encore de la physique à travers les calculs de vecteur.