

RAPPORT TP1

git url : <https://github.com/M3K0PZ/LOG725-TP>

Etape 1 : Choisir un jeu

Pour la sélection du jeu de départ je me suis basé sur le code de boing! du livre du livre code the classics : <https://github.com/Wireframe-Magazine/Code-the-Classics>.

Etape 2 : - Implémenter un ECS

Pour l'implémentation, j'ai repris le code de l'ia du pong, et j'ai entièrement refactorisé le code du jeu pour le faire fonctionner uniquement sur les ECS (Entity component system).

Après avoir enlevé les parties sources de bugs, dues à d'autres librairies et des fonctionnalités pas abordées en cours (exemple les animations, ou pgzero), j'ai fini avec un jeu pong totalement différent, et entièrement fonctionnel avec des ECS.

fonctionnement : w et s pour monter/descendre, on se bat contre une ia, le premier à 7 points gagne, la balle accélère au fur et à mesure du jeu.

Chaque entité possède ses propres composants, qui sont mis à jour dans le système de rendu (nommé drawSystem)

Mon jeu possède un système pour :

les mouvements des entités (la balle)

les contrôles des raquettes (IA + joueur) :

- l'ia tremble, c'est normal, chaque tick on force une erreur entre -30 et 30 pixels, pour que le joueur aie une chance de gagner, c'est un problème qu'on aurait pu fix en ajoutant des composants pour initialiser du temps et faire un lissage de la trajectoire dans le système. (si on enlève l'aléatoire alors l'ia ne tremble plus mais vous ne gagnerez pas) (ligne 97 dans systems.py)

la gestion du score

l'affichage graphique :

- il y a 3 "updates" différents, j'avais commencé à implémenter un gestionnaire d'états pour faire un écran d'accueil, de fin et de jeu. Cependant, on m'a rappelé que le but de ce TP est de montrer la compréhension des ECS, donc j'ai arrêté le développement de cette solution.

Comme j'ai déjà passé beaucoup de temps sur créer un jeu entier fonctionnant sur l'ECS, j'ai mis de coté le développement de patrons que j'aurais pu implémenter :

- un observer pour nous aider à gérer les événements sonores de mon jeu
- un inputhandler comme vu en labo pour les touches
- Le système de state a été supprimé par manque de temps mais il m'aurait aidé pour gérer différents états du jeu (par exemple si on veut jouer 1v1 c'est possible il suffit de mettre un composant controller a la place d'ia et le ECS s'adapte automatiquement pour rajouter les flèches du clavier et on peut jouer a 2..