

Rapport TP1 LOG725

Repo Github :

Jeu choisi pour l'implémentation

J'ai choisi le jeu « Boing! » qui est une copie de Pong. Le jeu est codé en Python.

Entity-Component-System (ECS)

Pour l'ECS, j'ai mis en place plusieurs classes qui permettent de gérer le comportement du jeu selon ce système. Pour les entités, j'ai implémenté la classe « Entity » qui contient plusieurs fonctions permettant d'ajouter, de retirer et d'accéder à certains composants que l'entité possède. Parmi ces composants, j'ai créé les classes « Position », « Velocity » et « Sprite ». Le composant position permet de positionner l'entité dans le jeu, le composant Velocity permet de configurer la vitesse et la direction de l'entité et finalement le composant Sprite permet au système de « Rendering » d'afficher l'entité à l'écran.

J'ai appliqué l'ECS sur l'objet « Ball ». L'objet est mis à jour dans la classe « Game ». La classe « Game » est en quelque sorte le « main » de notre application, c'est là qu'on met à jour toutes les entités actives. Éventuellement, on pourrait appliquer l'ECS sur la classe « Bat » pour que cette classe devienne une entité ayant les composants Position, Velocity et Sprite.

Ensuite, on pourrait appliquer l'ECS sur la classe « Impact ». On pourrait transformer cette classe en système de collision. Ce système pourrait gérer les collisions actives. Pour que ce système puisse fonctionner, il faudrait ajouter une classe composant du style « Collider » pour que le système « ColliderSystem » permette de voir si une entité est entrée en collision avec une autre.

Autres patrons

J'ai pensé à implémenter le patron commande pour gérer les entrées utilisateurs. J'ai créé un « InputHandler » pour le joueur 1 et pour le joueur 2. Ce patron permet de gérer les commandes plus facilement. J'ai rencontré quelques difficultés avec l'implémentation de ce patron, j'en parlerai plus tard.

Un autre patron qui pourrait être intéressant serait le patron observateur. En effet, dans le jeu que j'ai choisi il y a un affichage des points des joueurs. Il serait intéressant de mettre en place ce patron pour simplifier la mise à jour des scores. En effet, on pourrait attacher l'observateur à la classe « Game » pour que celle-ci puisse le notifier lorsqu'un point est marqué. On pourrait avoir un observateur pour chaque joueur ainsi on peut notifier le bon observateur en regardant quel joueur a marqué le point. L'observateur pourrait ensuite mettre à jour l'affichage des scores. Il faudrait cependant avoir une classe « Score » qui s'occupe de garder les scores en mémoire. Cette classe pourrait être une entité ayant une position et un sprite.

Difficultés rencontrées

Pour ce travail pratique, j'ai eu de la difficulté à implémenter l'ECS et particulièrement le patron commande. Étant donné que c'était ma première expérience avec Python, j'ai eu de la difficulté à apprendre comment les classes fonctionnent dans ce langage. En plus d'être débutant en Python, le jeu que j'ai choisi a été programmé en utilisant un « package » de « pygame » qui se nomme « pgzero ». J'ai regardé le code de ce « package » et si j'ai bien compris, ce « package » agit en quelque sorte comme une « entity », mais qui contient directement les composants de position, vitesse et image comme attributs de classe.

Le jeu était complètement basé sur ce « Package », donc à chaque changement que je faisais au programme, beaucoup d'autres dépendances étaient affectées.

J'ai ensuite eu de la difficulté à mettre en place le patron « command ». En effet, la mise à jour est faite d'une certaine façon que les commandes ne sont pas détectées à chaque « frame » ou la détection n'est pas faite constamment lors du jeu. Je ne suis pas certain de la raison, mais j'ai passé quelques heures à essayer de comprendre. J'ai laissé le patron dans les fichiers sur Github, mais il n'est pas implémenté, car le jeu ne fonctionnait pas correctement sinon.