Journal de bord
Test Enodo

Jour 0 :

A la réception du test, j'ai d'abord été surpris parce que je m'attendais à devoir créer un jeu plus traditionnel. Ensuite j'ai surtout été intrigué parce que je n'avais aucune idée à première vue de la méthode à aborder pour obtenir le résultat voulu.

J'ai donc consacré un moment à bien comprendre et visualiser le résultat final, en prévoyant les contraintes et fonctionnalités propres à chaque étape. Je ne voulais pas me coucher sans avoir tout prévu pour le lendemain.

Dès le départ, j'ai su que ce que l'étape 1 me prendrait bien plus de temps que les trois autres. Les deux premiers points importants étaient de reconnaitre une intersection entre deux segments (pour les interdire) et de déterminer si un point est à l'intérieur d'un polygone ou non.

➔ Utilisation de fonctions et outils existants ? création dynamique de meshs/colliders ?

Les créer « from scratch » ?

J'ai finalement décidé que la meilleure méthode était d'utiliser une simple formule géométrique pour détecter les intersections, puis d'utiliser cette reconnaissance d'intersections pour savoir si un polygone contient un point.

Jour 1 :

J'ai complété l'étape 1 à l'aide de l'algorithme préparé sur papier le soir précédent.
Tout s'est plus ou moins passé comme prévu : arrivé au soir j'avais mon outil de dessin, la reconnaissance des intersections et la reconnaissance d'un clique à l'intérieur d'un polygone.

J'ai décidé d'affiché les points en tant qu'objets pour une meilleure visualisation, j'ai donc aussi ajouté un script que j'ai écrit il y a quelques années pour avoir un système de Pool. J'en ai profité pour mettre ce script à jour pour qu'il gère les types génériques. Le peu de temps que ça m'a pris a été regagné par la suite.

Le principal souci de cette journée a été le stress de ne travailler que sur l'étape 1, en ayant tout juste préparé l'étape 2. Si je savais que les dernières étapes me prendraient moins de temps, je n'étais globalement pas rassuré.

Jour 2 :

La création des boites, les modifications de *transform* et les autres fonctionnalités demandées dans les 3 dernières étapes étaient plus classiques, j'ai donc pu avancer beaucoup plus vite.

J'ai au départ pensé à un système d'héritage pour différencier bâtiments simples et ennemis. J'ai au final considérer que, en imaginant que le projet aille plus loin, il n'y avait pas de fonctionnalité de l'un qui n'était pas intéressante chez l'autre.
Si un bâtiment ennemi peut faire des dégâts, il n'y a pas de raison que ce ne soit le cas pour aucun bâtiment simple. Aussi, cela permettrait si besoin, de faire basculer un bâtiment d'allié à ennemi ou vice versa.

J'ai commencé avec un système de création simple, avec uniquement le *material* qui changeait en fonction du type et un changement de booléen pour l'ennemi, mais au fur et à mesure, différents points m'ont poussé à recréer le système.
Au final j'utilise donc des *scripted objects* pour créer des *templates* de bâtiments tous différents, et instanciés selon le mode de construction utilisé, tout en utilisant la même Pool de bâtiments pour tous.

Pour le placement des bâtiments, j'aurais aimé qu'ils soient plus centrés dans leur zone, mais je n'ai pas trouvé de solution pour ça. Calculer le centre de gravité des polygons aurait été une idée, mais dans certains cas, le batiment se serait retrouvé à l'extérieur de la zone.

Ce dont je suis content :

- Le système de *snap* est plus pratique et utile que prévu
- Une fois l'étape 1 passée, le reste du projet a suivi naturellement
- Le début du projet m'a complètement sorti de la zone de confort mais je suis satisfait du résultat final
- Le projet a été très intéressant à réaliser!

Ce dont je ne suis pas satisfait :

- Il y a une surabondance de *Foreach*, dans un plus gros projet, ça impacterait trop les performances.
- Le système pour empêcher de tracer des diagonales empêche aussi la création de certaines formes concaves.
- Les éléments sont potentiellement mal placés si le plane ne se situe pas à z = 0
- Les polygones doivent être dessinés entièrement, le système ne détecte pas si le polygone est fermé par un autre.
- Je continue de passer trop de temps sur des choses peu importantes
- Il y a eu une coupure d'électricité pendant l'écriture de ce document et je n'avais pas sauvegardé depuis 10min

Commandes supplémentaires :

Clic droit ou Left Control →Changer l'orientation de la caméra

Flèches directionnelles → Déplacer la caméra

Space →Reset le dessin en cours