

Jimmy Labrecque

Rapport projet synthèse

« Hollow Grotto »

Présenté à

Jean-Christophe Demers

Projet synthèse C61

Groupe Intensif

Cégep du Vieux Montréal

25 mai 2021

Présentation

Hollow Grotto est jeu de survie dans des cavernes souterraines dont l'objectif est de retrouver des robots brisés et de les réparer. Pour y arriver, on possède une combinaison mécanique qui nous donne l'habilité de creuser des tunnels dans les parois de la caverne.

Résumé

La production s'est assez bien déroulée sachant que j'ai peu suivi le plan de base. La préproduction m'a aidée sur certains points mais j'ai tellement divagué du plan à cause des nombreux aspects qui m'étaient inconnus. J'avais utilisé Unity auparavant mais je ne m'étais jamais aventuré aussi loin dans ses fonctionnalités. Aussi, j'apprenais le logiciel en même temps dans le cours de Jeux Vidéo avec Maël donc c'était dur de créer un plan sans savoir ce qui est possible, ou devrais-je dire, facilement accessible grâce à Unity.

Mais en fin de compte, j'ai été surpris de la facilité avec laquelle j'ai pu avancer. J'étais sceptique sur le projet car il y avait beaucoup de concepts dont je n'avais aucune idée comment coder mais avec un peu de recherche et de réflexion, je suis arrivé à de bons résultats.

Fonctionnalités

- Système de Chunks : Parfaitement fonctionnelle
- Object Pooling : Parfaitement fonctionnelle
- Génération de caverne : Parfaitement fonctionnelle
- Mouvement : Parfaitement fonctionnelle
- Creuser : Parfaitement fonctionnelle
- Objectif : Parfaitement fonctionnelle
- Sauvegarde : Parfaitement fonctionnelle
- Vie : Parfaitement fonctionnelle
- Ver de terre géant : Parfaitement fonctionnelle
- Menu principal: Parfaitement fonctionnelle
- Grappin : Semi-fonctionnelle
- Fusée éclairante : Abandonnée

- Inventaire : Abandonnée
- Animation : Semi-fonctionnelle
- Son : Semi-fonctionnelle

Améliorations

Une des améliorations que j'ai en tête est la façon de trouver un endroit valide pour faire apparaître le joueur et les robots. Je pense que, comme la génération de mesh, je pourrais écrire le code dans un *Compute Shader*, ce qui accélérerait le calcul énormément et ça empêcherait le petit lag lorsqu'on répare un robot. Aussi, je modifierais les variables que j'ai mis *public* seulement pour avoir accès à la variable dans l'inspecteur de Unity. Parce que dans Unity, si une variable est *public*, elle peut être modifiée à l'extérieur du code dans la section *Inspector* du UI. Mais il existe d'autres façons d'afficher une variable sans la mettre pour autant *public*, il faudrait juste repasser pour faire le changement dans tout le code.

Auto-Évaluation

Même si dans le cours de jeux vidéo j'ai appris les bases de Unity, j'ai beaucoup recherché sur internet à propos du logiciel pour en apprendre d'avantage. Cependant, la formation m'a beaucoup aidé à développer de bonnes techniques de programmation et même à savoir ce qu'est un design pattern (même si j'aurais voulu avoir un cours spécifique sur les designs patterns). Aussi, j'avais déjà essayé de faire des jeux sur Unity avant le DEC mais je n'ai jamais fini de projets et ils n'avaient aucunement l'envergure de celui que je viens de compléter. Donc, ce n'est pas seulement un accomplissement académique, c'est aussi une victoire pour moi parce que je suis enfin capable de réaliser un rêve d'enfance, qui est de faire mes propres jeux vidéo.

Auto-évaluation note finale : 90%

J'aurais vraiment voulu ajouter un peu plus de vie dans les cavernes. J'ai même des prototypes pour des roches sur le sol, des stalactites qui peuvent être creusés. De base, le code utilisé pour trouver un endroit valide à spawner le joueur et les robots était supposé servir à peupler le sol de petites roches. Mais bon, finalement, le jeu fonctionne et je suis satisfait avec cela.