

ARNAOUT RICHARD

VECCHIO MATIAS

COMPTE RENDU PROJET C : STEALTH

Sommaire:

-Présentation du Jeu

-Exécution du Programme

-Modules implémentés

-Difficultés rencontrés

-Conclusion

Présentation du Jeu{

Stealth est un jeu d'infiltration jouant avec un joueur. Sur un terrain de jeu composé de plusieurs reliques, ce dernier doit les ramasser , mais attention il ne doit pas se faire voir par les gardes. Pour cela il dispose de murs ou il peut se cacher ainsi que des superpouvoirs comme être invisible ou encore se déplacer plus rapidement. Attention , ces super pouvoirs ont une utilisation restreinte , en effet le joueur dispose d'une jauge appelée jauge de Mana. il faut l'utiliser avec escient car si elle se vide, le joueur n'aura plus le droit à ses super pouvoirs. Lorsque le joueur ramasse une relique, elle devient grisée. Si un gardien passe dessus et constate son absence alors, le mode panique sera déclenché et la vitesse des gardes sera alors décuplée.

Attention à ne pas vous faire rattraper. Pour que la partie soit gagnée. Vous devez ramasser avec votre joueur toutes les reliques du terrain et lorsque vous les avez toutes ramassées. Votre dernière ligne droite est de revenir au point de départ ou vous avez spawn. Si vous réussissez , un écran de victoire apparaîtra, le nombre de secondes que vous avez joué à la partie s'affichera également ainsi que le taux de mana que vous avez utilisé durant le jeu. Le but du jeu est de ne pas se faire détecter . Si cela arrive, vous aurez donc perdu la partie et un écran d'affichage rouge s'affiche en disant que vous avez perdu. Vous devrez alors recommencer à nouveau.

}

Exécution du Programme{

Le but du projet à été de programmer ce Jeu à travers le langage C. En adaptant des modules optimisés, avoir du code propre et efficace, ainsi qu'un bon fonctionnement du Makefile.

Pour exécuter le programme, il va falloir se placer dans le répertoire courant ou est contenu le makefile et taper la commande suivante dans le terminal:

make

Ainsi, un exécutable sera généré, il faudra alors taper la commande suivante :

./stealth

L'écran de menu s'affiche alors et vous n'aurez plus qu'à faire vos choix pour jouer au jeu , quitter le jeu ou avoir de l'aide. Vous pouvez utiliser la souris.

Vous pouvez dès à présent "enjoy" votre Partie.

Les logs sont fournis dans la racine avec le makefile pour voir l'étendue du travail sur redmine ainsi que les doc avec Doxygen.

}

Modules implémentés{

Après implémentation du makefile, la première étape à été de constituer les modules afin de répartir correctement l'aspect du programme. Plusieurs modules ont été proposés au départ sauf que certains paraissent inutiles du coup nous avons réduit à 8 modules. Au fur et à mesure du projet on en a rajouté quand cela est nécessaire. Le travail à été réparti en diverses tâches. La toute première étape à été de faire le module **Object** qui va nous permettre d'initialiser de par sa structure, un objet dans le jeu afin de lui attribuer ses déplacements par la suite avec des fonctions qui font cela et une Structure **Engine_orientation** permettant la direction des objets. Ce module est très important car c'est le cœur de tous les personnages du jeu.

En effet, avec sa structure nous avons créé plusieurs structures dans différents modules comme Player qui sera constitué d'une structure **Engine_Player** attribuée au joueur. Une structure **Engine_Guard** permettant de stocker un objet contenant sa position dans le jeu utilisé bien évidemment dans le module **Guard**. Puis une structure **Engine_Relique** pour placer les reliques dans la partie avec son module **Relique**.

Engine_Object est quasiment utilisé partout dans le programme notamment dans les murs pour stocker la position des murs. bien entendu un module **Wall** et une structure **Engine_Wall**.

Input va permettre l'orientation des objets du jeu , **Graphisme** sera dédié à toute la partie graphique du jeu. En utilisant la bibliothèque MLV on aura , le dessin du menu avec ses boutons , dessin de l'image , des événements de clics de la souris , dessin du joueur, des gardes, des reliques et des murs. **Sound** sera un module également utilisé par la bibliothèque pour générer la musique du jeu et donner un aspect ambiant lors de la partie.

}

Difficultés rencontrés{

Commencer un Projet en partant de très peu d'éléments ne s'avère pas toujours facile, il a fallu coder correctement. Déplacer des valeurs numériques dans des constantes. Tester nos coordonnées par des fonctions tests, En effet, tout un travail qui a demandé de la rigueur et du temps. Une fois votre joueur créé et ses déplacements grâce aux touches directionnelles effectuées. Nous nous sommes attaquées à la partie garde en créant son curseur , puis en dessinant son champ de vision. Il a fallu ensuite gérer la collision entre le joueur et ce garde. Ce n'était pas évident de faire en sorte que le joueur puisse entrer en collision par tous les côtés du garde. Pour cela on a créé une fonction mathématique distance_between_objects permettant de voir la distance entre deux objets en paramètres.

Grâce à cette fonction, on va pouvoir entrer en collision avec notre garde. Cette fonction nous à été notamment très utile par la suite. elle nous a fait "une pierre 10 coups", en effet elle va servir pas que pour la collision des gardes mais aussi la collision du joueur avec les reliques pour récupérer les reliques avec un booléen ispickedup permettant de dire si la relique à été ramassée ou non. Autrement c'est lorsqu'elle est ramassée et que la collision du gardien avec cette relique ramassée déclenche le mode panique. Et enfin, la collision du joueur avec sa base de départ permettant la fin de jeu. Bien entendu, il a également ramassé toutes les reliques du terrain.

A savoir qu'il n'a pas été évident de sauvegarder les reliques lorsqu'elles ont été ramassées, nous avons fait au début un compteur de reliques qui va parcourir le tableau de reliques que l'on a généré dans son module respectif. Et quand le joueur se pose dessus pour la ramasser cela faisait un problème de compteur a chaque framerate au lieu de donner +1 cela donnait une boucle qui ne s'arrêtait pas car le joueur était encore sur les coordonnées de la relique en question. Nous avons corrigé cela en entrant les bonnes conditions. On avait fait une petite erreur dans les conditions, c'était la raison pour laquelle ça ne fonctionnait pas au début.

Nous avons également géré les déplacements des gardes en utilisant des déplacements aléatoires, le garde va prendre une direction à chaque framerate selon la Macro qu'on lui à attribué il changera aussi vite de direction. Si il rencontre un obstacle tel un mur par exemple, on lui a donné un enum REVERT lui permettant de revenir en arrière pour éviter qu'il reste au même endroit.

Nous avons ensuite géré l'écran de menu en chargeant les images et les boutons a notre préférence. puis la partie clique de souris à été aussi délicate car nous devons voir ou est ce que les coordonnées des souris devaient être lors d'un clic. Une fois celles-ci identifiées nous avons divisé nos valeurs par la taille des fenêtres et nous avons divisé nos constantes de fenêtre par des indices pour avoir une précision sur nos cliques lorsqu'on touche une box qui contient soit le bouton jouer , quitter ou help.

Nous n'avons pas réussi à adapter l'algorithme de l'énoncé à notre implémentation. Par manque de temps nous n'avons pas pu expérimenter efficacement les murs de façon récursive comme dans l'énoncé. Avec le peu de temps qu'il nous restait, nous avons généré des murs manuellement. Nous avons fini très juste dans les temps. On a eu le temps de vérifier que nous n'avions pas de fuites de mémoire et qu'on avait bien désalloué toutes nos variables alloués dynamiquement.

}

Conclusion{

Nous avons été très satisfait de la qualité de notre travail, Nous avons pu fournir du code propre à la lecture et efficace en respectant les consignes de l'énoncé. Ce Projet à été très intéressant et très enrichissant en termes d'expérience. Nous nous sommes mutuellement aidés entre nous en nous trouvant des solutions à nos difficultés rencontrées pour nous avoir permis un gain de temps pour les tâches suivantes. Le langage C est très pratique pour nous. On peut apprendre beaucoup en termes d'algorithme , la beauté de ce langage, c'est qu'on peut facilement évoluer en codant habituellement et par la suite nous permettre de faire de grandes choses en Informatique.

}