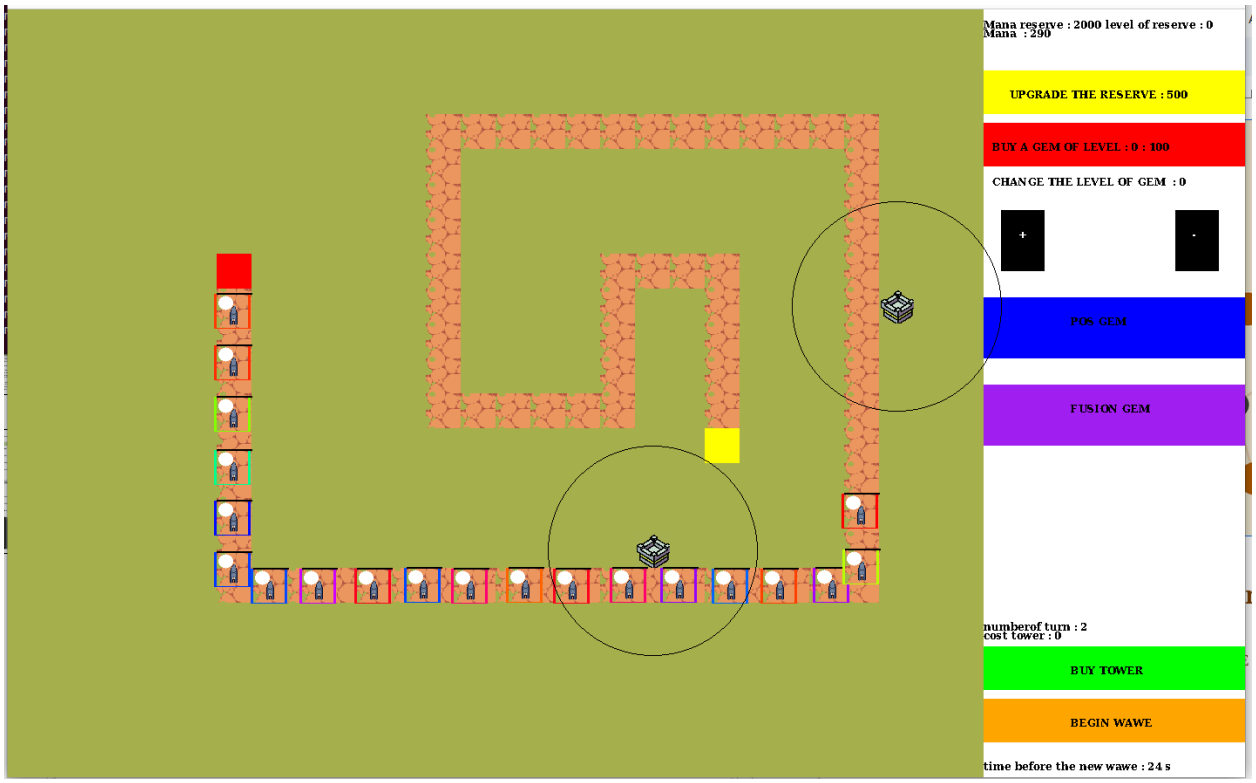
PROJET FINALE - PROG C AVANCEE : RAPPORT

Projet en langage C édité par : SIVAYNAMA Prashath et DORLEAN Julien - Groupe 3



TABLES DES MATIERES :

I – Présentation et objectifs du projet

II – Moyen mis en œuvre pour réponse aux exigences du projet

III – Difficulté rencontrée et fonction non incrémentées

IV – Conclusion

V – Annexe 1 : Instruction de compilation et information du jeu

I – Présentation et objectifs du projet

Le projet ‘Tower Defense " est jeu programmé en langage C, vous devez arrêter la vague de monstre avec les tourelles et les gemmes .

Le joueur possède du mana s'il tombe à 0 le joueur à perdu et il en gagne en tuant des monstres et il en perd si les monstres atteignent son camp..

Nous utilisons la bibliothèque graphique ‘MLV’ pour représenter graphiquement les informations du jeu tel que les tourelles, les gemmes, les monstres et le plateau de jeu .

L’objectif de ce projet est donc de réaliser une interface graphique proposant un contrôle par des clicks sur la fenêtre de jeu et que le joueur tienne le plus lomgtemps possible .

II – Moyen mis en œuvre pour réponse aux exigences du projet

Nous avons découpé notre projet en 8 modules en tout sans compter la main.

Le premier module est le module nommée “game engine” il creé le plateau de jeu et la route pour les monstres il inclut aussi plusieurs énumérations importantes pour la suite du jeu tel que la Direction Case qui représente la direction d’un objet.

Le second module est le module nommé “gem” il permet de gérer les gemmes (les ajouter, les supprimer et les fusionner) il inclut aussi plusieurs énumérations importantes pour les gems .

Le troisième module “monstre” permet de gérer les monstres (il permet de créer les vagues de monstre avec le type choisit aléatoirement et de supprimer des monstres) il'inclut aussi plusieurs énumérations importantes pour les monstres ce module contient aussi la plus grande structure car les monstres ont plusieurs caractéristiques à prendre en compte (leur vie , leur teinte , le temp de début d’un effet de résidue ..).

Le quatrième module est “tower” il permet de créer les tourelles du joueur chaque tourelle contiennent une gemme et une position .

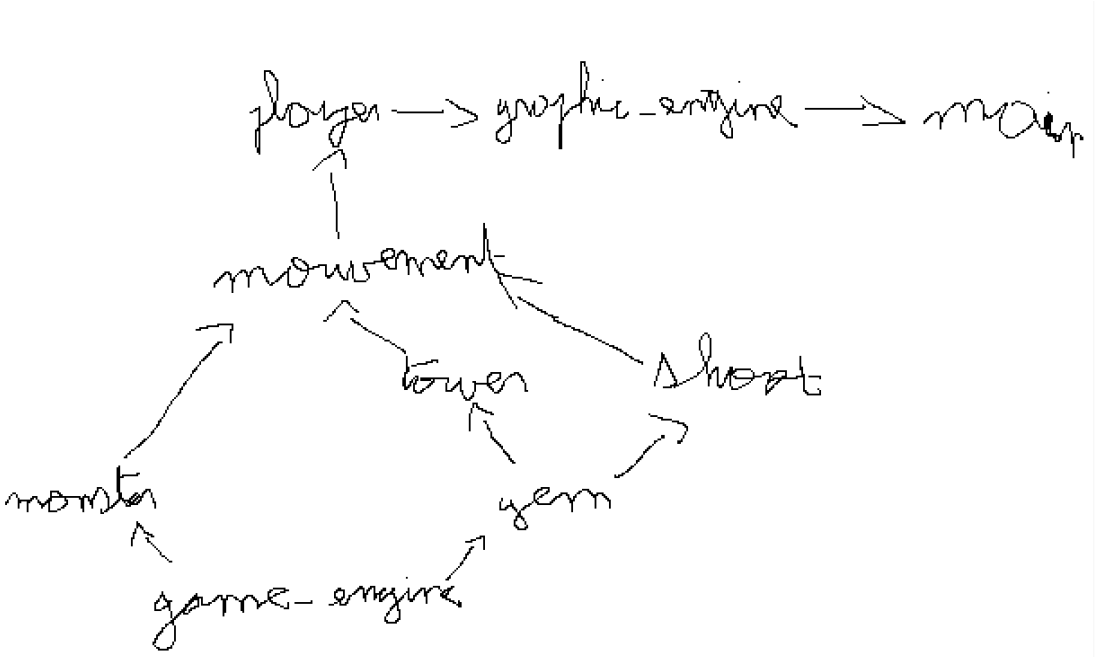
Le cinquième module “shoot” gère les tirs des tourelles (ajoute des tirs à la pile de tir du joueur et les supprime aussi) .

Le sixième module “mouvement” gère les déplacements des monstres, des tirs et des effets de certains résidu tels que les ralentissements et les degats doublé ou tripler sur les monstres .

Le septième module est “player” elle contient notamment la structure Player qui contient la pile des tourelles, tour et tir c’est le module qui va utiliser d’autre fonction intégrer dans le module tower, gem et shoot pour gérer les achats de gemme, de fusion de gemme, achat de tour et supprimer du mana si les monstres on atteint son camp et d’en gagner si un monstre meurt .

Le dernier module est “graphic engine” qui va afficher les menus (les boutons d’action et les informations nécessaires pour le joueur) et détecter les cliques du joueur.

Voici un graphe d’inclusion de notre projet ;



Nous ne sommes pas répartis les tâches ainsi un a fait la création du plateau et la route des monstres du jeu pendant que l’autre à faire les monstres et leurs mouvements et puis nous ne sommes répartis les autres taches restantes .

III - Difficulté rencontrée et fonction non incrémentées et changement de fonction

Les difficultés rencontrées ont été les mouvements des objets notamment les monstres car les mouvements des shoots sont similaires au monstre .

Les fonctions non incrémentées sont le fait de laisser 2 seconde la gem quand on vient de la poser sur la tour avant qu'il tire dans notre projet la gem tire directement .

L'effet de dendro ne fonctionne pas correctement il n'inflige pas de dégât sur la durée nécessaire .

Pour la création de la gemme du joueur nous avons changer pour que le joueur puisse seulement choisir le niveau de la gemme souhaiter l'élément et la teinte seront prises aléatoirement (la teinte sera comprise entre la fourchette de teinte possible pour l'élément obtenu)..

IV – Conclusion

Comme dit plus haut en ce qui concerne la répartition des tâches pour ce projet nous avons fait un travail quasi équivalent du début à la fin ce celui-ci .

Ce projet nous a permis de mieux maitriser certaines subtilités du langage C, et des mieux comprendre la bibliothèque MLV .

Les difficultés surmontées durant ce projet pour nous ont été la générations de la route des monstre et les mouvement des différents objects (monstre et les tirs) .

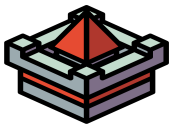
V – Annexe 1 : Instruction de compilation et information du jeu

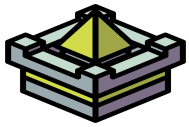
Le projet se compile avec la commande suivante :

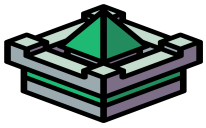
```
make
```

Une fois l'exécutable générer, celui-ci se trouve dans le dossier bin entrer la commande suivante :

```
./bin/exe
```

Les tours Pyro sont affiché ainsi : 

Les tours Hydro sont affiché ainsi : 

Les tours Dendro sont affiché ainsi : 

Les monstre de la vague NORMAL et CROWD sont affiché ainsi : 

Les monstre de la vague AGILE sont affiché ainsi : 

Les monstres de la vague BOSS sont affiché ainsi : 

Les teintes des monstres sont affichées graphiquement avec un carré autour de l'image du monstre la couleur de la teinte et les effets de résidu sont représenté avec un cercle dans l'image :

Les couleurs des cercles de résidu sur le monstre sur l'image :

- Résidu Dendro + Hyro : cercle jaune
- Résidu Pyro + Hyro : cercle violet
- Résidu Pyro : cercle rouge
- Résidu Hydro : cercle bleu

- Résidu Dendro : cercle vert
- sinon cercle blanc

Les tirs seront affiché en cercle de la couleur de leur teinte