

# Projet final : Le Village

# Introduction

Nous souhaitons réaliser un jeu sur le thème de la Fantasy.

Le joueur contrôlera les actions d'un héros se trouvant dans un village composé de diverses structures.

Ce héros devra combattre des monstres pour gagner des pièces d'or, qu'il pourra utiliser pour se soigner ou pour améliorer son équipement, le rendant plus fort.

Le jeu s'arrête lorsque le héros a perdu l'intégralité de ses points de vie.

# Le héros

Le héros contrôlé par le joueur peut visiter les bâtiments du village. Il pourra incarner un guerrier, un magicien ou un paladin. Ce choix sera effectué au lancement du jeu.

Le guerrier inflige 50% de dégâts en plus et peut équiper des épées.

Le magicien peut équiper des bâtons et à 20% de chance d'esquiver les dégâts.

Le paladin peut porter des épées et des boucliers.

Les héros possèdent au minimum un nom, des points de vie et de l'or.

Ils ont aussi un inventaire pour stocker des potions de soin.

Le joueur pourra prendre des coups, se soigner, gagner et dépenser de l'or.

# Les bâtiments

Il y a au moins 3 bâtiments qui peuvent être implémenté dans le village :

- L'auberge, qui sert à soigner le héros contre des pièces d'or ;
- La mine qui permet au héros d'affronter des monstres pour gagner des pièces d'or ;
- Le marchand qui permet au héros d'acheter des armes, des boucliers ou des potions.

Les bâtiments sont **chargés dynamiquement** : certains bâtiments peuvent ne pas être présents, d'autre présents en double. Il devra donc être possible par exemple de rejoindre un village avec 3 auberges, 1 mine et 2 marchands.

# Système de combat

Le jeu se déroule en tour par tour, héros et monstre attaquant à tour de rôle.

Le héros possède des dégâts de base.

Les dégâts de son arme (s'il en possède une) doivent être additionnés à ces dégâts de base.

Le bouclier annule une partie des dégâts subis équivalent à sa valeur de protection.

Le héros peut aussi obtenir des potions de soin qu'il pourra utiliser pendant le combat.

# Les profondeurs de la mine

A chaque fois que le joueur sort vivant de la mine, celle-ci gagne un niveau de difficulté (elle commence au niveau 1).

Le nombre de monstres à affronter dans la mine devra suivre la suite de Fibonacci (niveau 1 = 1 monstre, niveau 2 = 1 monstre, niveau 3 = 2 monstres etc).

A vous de gérer le degré de difficulté en choisissant la vie et les dégâts des monstres de la mine, ainsi que sa potentielle diversité.

Le joueur ne peut quitter la mine tant qu'il n'a pas tué tous les monstres la composant.

# Rendu et notation

Le projet est à rendre en groupe de 2, avant le **1er novembre 2024 à 23h59**.

Vous devez rendre un diagramme de classe (format png) de votre projet.

Vous serez noté sur plusieurs points (par ordre d'importance) :

- Un diagramme de classe cohérent et bien pensé;
- La bonne utilisation de la POO (classes, héritage, polymorphisme...);
- La bonne compréhension du C++ (typage, boucles, appels de méthode...);
- La propreté du code (nom de classes/variables logique, code bien découpé...);
- Un bon algorithme de jeu;
- Les améliorations apportées au projet.

En clair : vous serez principalement noté sur la qualité du code écrit et non sur le produit final !

# Améliorations

Voici quelques idées pour améliorer notre petit jeu :

- Avoir plusieurs villages à débloquent (dont la liste des bâtiments sont chargés depuis un fichier ?)
- Baser les ressources sur le modèle de Donjons et Dragons (ou autre)
- Faire une interface graphique (via Qt, SFML, ncurses-like ou autre...)
- Écrire toute une campagne
- Faire du multijoueur en ligne 🤖
- ...
- Amusez-vous :)



# Conseils

- Dessinez un diagramme de classe AVANT de commencer à coder. Cela vous fera gagner du temps lorsque vous devrez créer les classes : la structure étant déjà faite, il ne restera plus qu'à remplir les méthodes !
- Décidez aussi de ce que vous allez devoir produire pour le rendu final avant d'attaquer le projet : strict minimum, quelques bonus, interface graphique.. ? Suivant les choix pris, la structure du projet pourrait être amené à évoluer.
- Répartissez-vous les tâches ! Vous êtes en binôme, profitez-en.
- N'oubliez pas que le plus important reste de faire un code bien structuré/découpé et propre. Un groupe ayant fait le strict minimum avec un code irréprochable aura une bien meilleure note qu'un groupe ayant une interface graphique mais un code illisible !
- N'hésitez pas à refaire certains TP (sans regarder la solution) si certaines notions sont encore floues.
- Même si vous n'êtes que 2 pour effectuer ce projet, ~40 autres étudiants le réalisent aussi : parlez-en entre vous pour vous donner des idées. :)