

Projet de programmation

LEVEL ONE

Le projet LEVEL ONE consiste à développer le premier niveau (*i.e.*, le début) d'un *jeu vidéo* en Java dans lequel le *joueur* contrôle un personnage dans un *monde* où il peut combattre des *monstres*, interagir avec des *personnages non-joueurs (PNJ)*, trouver et utiliser différents *items*. Les sections suivantes détaillent les différents éléments du jeu.

Le jeu

- L'interface graphique du jeu doit être réalisée avec la bibliothèque graphique JavaFX (possiblement complétée avec d'autres bibliothèques spécifiques).
- Les conditions de victoire (*e.g.*, vaincre un boss ou tous les ennemis, atteindre un lieu spécifique, trouver un PNJ ou un item particulier) et d'échecs (*e.g.*, perte de tous ses points de vie, temps écoulé) sont laissées libres.

Les mondes

- Un monde est constitué d'éléments de décor qui sont traversables (possiblement sous conditions, comme une porte à ouvrir avec une clé) ou non (*e.g.*, murs, pièges) par le joueur.
- Un monde peut comporter des monstres, des PNJ et des items.
- Un monde comportent aussi des lieux permettant au joueur d'accéder à d'autres mondes.
- La manière dont un monde est représenté (grille 2D ou structure plus complexe) et affiché (en totalité ou en partie avec du *scrolling*), s'il est fermé ou ouvert, fini ou infini, prédéfini ou généré aléatoirement, est laissée libre.

Le joueur

- Le joueur se déplace librement dans un monde.
- Le joueur a des points de vie (et possiblement d'autres caractéristiques : nom, argent...).
- Le joueur possède un inventaire pour stocker les items obtenus.
- La manière dont le joueur contrôle son personnage (clavier et/ou souris, au tour par tour ou en pseudo temps réel) est laissée libre.

Les monstres

- Les monstres ont des points de vie (et possiblement d'autres caractéristiques : nom, force...).
- Les monstres possèdent un inventaire d'items qu'ils peuvent utiliser en combat ou laisser tomber une fois vaincu par le joueur.
- Il doit exister différents types de monstres.

- Le comportement des monstres (statiques ou en mouvement, avec un champs de vision ou non, stratégie d'attaque) et le fonctionnement du système de combat (attaque/défense, automatique en fonction des données du joueur et du monstre ou choix d'actions, au tour par tour ou en pseudo temps réel) sont laissés libres.

Les Personnage Non-Joueur (PNJ)

- Les PNJ possèdent un inventaire d'items.
- Il doit exister différents types de PNJ.
- Les interactions des PNJ avec le joueur (*e.g.*, donner/marchander des informations ou des items, déclencher une quête) sont laissées libres.

Les items

- Les items sont ramassés par le joueur et stockés dans son inventaire.
- Il doit exister différents types d'items.
- Les items peuvent avoir des effets sur le joueur, les PNJ, les monstres et les éléments de décor des mondes.
- Les effets d'un item (*e.g.*, redonner de la vie au joueur, diminuer les dégâts causés par un monstre, ouvrir une porte) et son activation (automatique ou manuelle, limitée ou illimitée) sont laissés libres.

Modalités

- Travail individuel ou en groupes (de 2 à toute la classe)
- 4 séances de 3h (=12h) + 12h hors emploi du temps = 24h minimum/élève
- Projet à rendre sur Teams (**Devoirs**) avant la semaine d'examens :
 - Une archive JAR exécutable,
 - Le code source (fichiers `.java`) correctement documenté,
 - Les ressources nécessaires (*e.g.*, images, bibliothèques),
 - Des diagrammes de classes et de séquences (UML),
 - Un mode d'emploi du jeu.
- Examen (individuel) = note finale :
 - 3h, sur votre machine avec votre projet
 - Exemples de ce qui peut être demandé :
 - Ajout de nouvelles conditions de victoire et d'échecs du jeu,
 - Modification du contenu d'un monde,
 - Ajout de caractéristiques au joueur, monstres, PNJ,
 - Création de nouveaux éléments de décor (traversables ou non par le joueur),
 - Création de nouveaux monstres qui combattent de telle façon,
 - Création de nouveaux PNJ qui interagissent de telle manière avec le joueur,
 - Création de nouveaux items qui ont tels effets sur le joueur, les monstres, les PNJ ou les éléments de décor.
 - Rendu : exécutable et code source Java sur Teams (**Devoirs**)
 - **Tout code erroné ou non testable sera noté 0.**