

HACK 'N' SLASH



Introduction

Un grand fan de jeux de type “Hack ‘n’ Slash” désire avoir sa propre version du jeu personnalisée. Il n’a cependant aucune compétence en développement informatique. C’est pourquoi il décide de faire appel à une aide extérieure. En tant que jeunes apprentis développeurs, vous êtes contactés pour réaliser cette mission !

Consignes générales

- Durée de développement: 3 semaines.
- La qualité d’expérience du jeu prime sur sa complexité: Un jeu avec moins de fonctionnalités mais qui tourne sans bugs sera préféré.
- Le code doit être clair et bien documenté (utilisation impérative du format javadoc).
- Dans un souci de droit logiciel, aucun projet ne sera accepté avec du code copié d’internet ou d’autres projets.
- Les spécifications ainsi qu’un code de base vous sont fournies comme point de départ.
- Date de rendu: **10 Décembre avant minuit**

GamePlay

Le jeu qu'il vous est demandé de réaliser est un Hack 'n' Slash. C'est un type de jeu dont le gameplay se focalise sur un personnage se déplaçant dans une map et affrontant des monstres dans le but d'obtenir des récompenses (items). L'objectif du joueur sera d'incarner un héros et de parcourir la carte du jeu en éliminant les ennemis à la recherche de **4 clés** ouvrant une porte magique.

Principes de base du jeu:

Votre jeu devra offrir les fonctionnalités suivantes :

- Le héros doit pouvoir se déplacer sur la map (*fonctionnalité de base fournie*).
- Le héros doit pouvoir attaquer les monstres.
- Le héros doit pouvoir ramasser les items lâchés par les ennemis à leur mort.
- Le héros doit pouvoir déposer les clés au niveau de la porte magique.
- Les monstres doivent pouvoir attaquer le héros (lorsqu'il est à portée).
- Il faut au moins 4 types de monstre différents dans la map en tout.
- Les clés sont libérées par les monstres lorsqu'ils sont éliminés, seulement 4 monstres possèdent une clef.

Dans le cadre du projet:

- La **map(GameWorld)** représente l'environnement de jeu:
 - Elle contient tous les éléments du jeu: les **Entités**:
 - Le joueur
 - Tous les monstres
 - Les items présents dans la map
 - A sa création, on définit les entités à placer, on les crée et à l'aide d'une fonction on les place aléatoirement dans la map.
 - Pensez aux types de collections à utiliser pour stocker les entités en fonction du cas, pour rester sur de bonnes performances logicielles et mémoire.
- Le **joueur**:
 - Le joueur peut se déplacer d'une case à l'autre grâce au clavier:
 - Touche **Q** pour une rotation à gauche de 45°
 - Touche **D** pour une rotation à droite de 45°
 - Touche **Z** pour avancer
 - Touche **S** pour reculer
 - Il possède un certain nombre de points de vie, il peut en perdre, en récupérer (consommer une potion touche **P**), lorsqu'il n'en a plus: Il meurt, game over.
 - Il doit stocker les items qu'il possède et récupère ceux "drippés" par les monstres ou déposés sur la map. (Touche **R** pour ramasser). Il les stocke dans

un inventaire qui contient tous les objets qu'il possède. L'affichage de l'inventaire se fait dans la console avec la touche **I**.

- Pour s'équiper d'un objet il faudra utiliser la touche **E** puis saisir le nom de l'objet à équiper dans la console.
- Le joueur peut s'équiper d'une épée, d'une armure, de plusieurs potions et de plusieurs bombes.
- Mécanisme de combats (Interaction au clavier)
 - (Touche **K**) Attaque directe: lorsqu'un monstre se trouve dans la case en face du joueur, le joueur peut l'attaquer.
 - (Touche **L**) Attaque indirecte: le joueur attaque le monstre à distance dans un périmètre de 4 cases en face du joueur (lancer de bombes)
- Lorsque la porte magique est en face de lui, le joueur peut déposer les clés qui sont dans son inventaire avec la touche **M**. Lorsque les 4 clés sont dans la porte, le jeu est gagné.
- Les **monstres**:
 - Deux types de monstres sont à prendre en compte:
 - Les **pièges**: Immobiles, si le joueur passe dessus, il perd des points de vie.
 - Les **chasseurs**, se déplaçant dans la map, peuvent attaquer le joueur, lui faisant perdre des points de vie.
 - Parmi les chasseurs on aura un type un peu spécial: le **Necromancien**: Ce chasseur a la possibilité de créer des enfants Necromanciens, et lorsque le Necromancien parent est tué, tous ses enfants meurent avec lui. Pensez à la structure de stockage judicieuse pour gérer les liens de parenté.
 - Vous devez créer au moins deux autres chasseurs.
 - Déplacement: les ennemis se déplacent de manière aléatoire sur la map.
 - Les monstres possèdent également des points de vie, des points d'attaque et des points de défense.

- Les **items**:

Nous distinguons 3 types d'items, les Consommables (potions, bombes) et les outils (armes, armures) et les clés:

- Les consommables peuvent être équipés de manière illimitée.
- Une seule Arme et une seule Armure peut être équipée à la fois.
- Les **potions** soignent le joueurs (augmentent ses points de vie d'une certaine quantité, dans la limite du maximum de points de vie).
- Les **bombes** infligent un certain nombre de points de dégât à distance.
- Les **armes** augmentent les points d'attaque du joueur d'un certain nombre.
- Les **armures** augmentent les points de défense du joueur d'un certain nombre.

- Les **clés** permettent d'ouvrir à elles quatre la porte magique.
- Mécanisme de combat :
 - Lorsque le joueur attaque un monstre, les dégâts infligés à ce dernier sont obtenus avec la formule suivante :
 - **degatsReçus = Atk(attaquant) - Def(défenseur).**
 - Si le joueur possède une épée, son attaque au moment du combat est augmentée du bonus de l'épée :
 - **Atk(attaquant) = Atk + bonusEpee**
 - Si le joueur se fait attaquer par un monstre dont les points d'attaque sont Atk, et que le joueur est équipé d'une armure, sa défense est augmentée au moment du combat.
 - **Def(défenseur) = Def + bonusArmure**
 - De plus, afin d'ajouter du réalisme au jeu, les attaques n'infligent pas toujours le même nombre de dégâts. Il peut arriver qu'elles soient moins efficaces. A vous d'ajouter un facteur aléatoire aux dégâts infligés.

Consignes

Vous avez un projet ambitieux à réaliser pour cette fin de semestre, mais qui devrait être assez amusant. Ce projet est à faire en binôme, et à rendre par mail à votre chargé de TP d'ici à dimanche 10 décembre à minuit.

La grille de notation prévue est la suivante (susceptible de changer sans préavis) :

- Projet réalisant toute la partie **Principes de base du jeu**, avec un code clair et commenté : 14/20
- IA améliorée pour les ennemis (poursuite du héros, coopération des monstres) : +1 pts
- IA pour l'attaque du monstre le plus proche sur la carte (missile autoguidé): +1 pts
- Ajout d'un mécanisme d'expérience du joueur + arbre de compétences : +3 pts
- Génération procédurale de la map : +1 pts
- Autres réalisations venant de votre imagination (à préciser au moment du rendu) : bonus de points à l'appréciation de votre chargé de TP.

On vous impose d'utiliser les Collections pour réaliser toutes vos structures de données. Donc pas de listes chaînées faites à la main : Pensez aux structures vues en TD (ArrayList, LinkedList...).

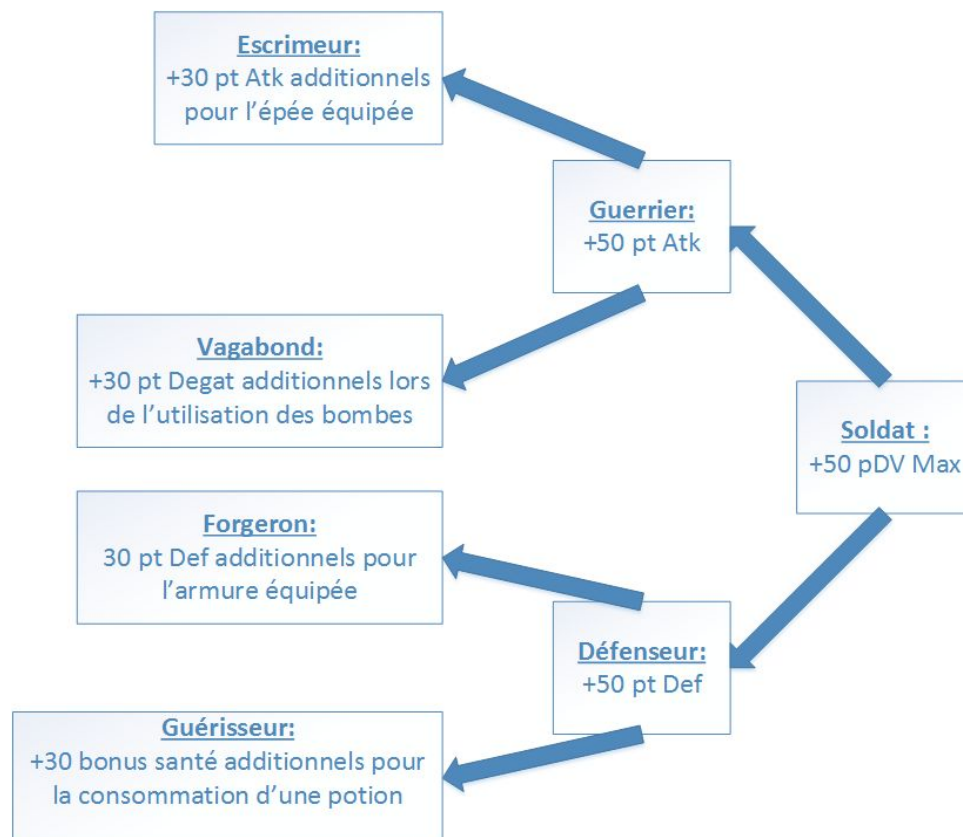
Comme d'habitude, copie de code sur un autre projet = 0 pour tous les participants.

Bon courage et surtout amusez-vous, c'est aussi le but de ce projet !

Annexe

--

Pour l'arbre de compétence: le joueur gagne de l'expérience en tuant des monstres. Lorsqu'il atteint un certain nombre de points d'exp, il passe au niveau suivant et dispose alors d'un point de compétence. Ce point de compétence est utilisé pour améliorer ses performances selon une structure d'arbre. Nous vous proposons l'arbre suivant :



Détails techniques et liens :

La librairie graphique utilisée est StdDraw, une petite librairie très simple utilisée dans un cours de Java de l'université de Princeton.

- Explications détaillées sur la librairie (en anglais, commence à partir du texte « **Standard Drawing** ») :

<http://introcs.cs.princeton.edu/java/15inout/>

- API StdDraw : <http://introcs.cs.princeton.edu/java/stdlib/javadoc/StdDraw.html>
- Liens vers des programmes exemple: <http://introcs.cs.princeton.edu/java/stdlib/>

Exemple d'un Hack 'n' Slash 2D: <https://www.youtube.com/watch?v=S7hA0PoHQVg>

Autre Exemple: Titan Quest, un excellent Hack n Slash 3D:

https://www.youtube.com/watch?v=7I0jN2h_TLA (à partir de la 18eme minute, illustration du principe de ramasser des objets dropés.)