



ÉCOLE
POLYTECHNIQUE
DE BRUXELLES

UNIVERSITÉ LIBRE DE BRUXELLES

RAPPORT FINAL

RPG - Mushrooms

INFO H-200

Auteurs :

Theo LISART

Neil LAMAS

06 mai 2016

Table des matières

1	Introduction	1
2	Explication de la structure orienté-objet	1
3	Fonctionnalités	1
3.1	Contenu	1
3.2	Affichage	2
3.3	Utilitaires	3

1 Introduction

Comme décrit dans les consignes, la dynamique principale du jeu consiste en l'exploration par le joueur de donjons infestés de créatures et de trésors. L'approche adoptée est la génération aléatoire de donjons.

Les sujets abordés seront :

- la structure du projet (orientée objet) ;
- le diagramme de classe (Model) et de séquences (cas particuliers)
- les fonctionnalités implémentées

2 Explication de la structure orienté-objet

Etant donné la diversité des tâches, le développement du jeu a été segmenté en trois aspects principaux : la modélisation de la dynamique (Model), l'affichage (View) et les interactions avec l'utilisateur (Controls).

Premièrement, la partie "Model" décrit les classes du jeu ainsi que les interactions entre elles. Les classes dynamiques sont le joueur (**Player**) et les monstres (**Mushrooms**) qui héritent de la classe personnage (**Character**).

La classe **Map** quant à elle sera le plateau de jeu, il s'agit d'une matrice de cellules, définies comme **Wall**, **Floor**, **Chest**, **AbstractItem** (un loot aléatoire d'un objet de n'importe quel type) et **Exit**. Les cellules dynamiques étant citée préalablement (toutes les classes héritant de Character). Deuxièmement, la partie "View" s'occupe de gérer la GUI du projet, offrant un rendu graphique du "Model". Il s'agit de KeyListeners notifiant les parties d'écoutes, les parties modèle notifiant les parties de View.

Troisièmement, la partie "Controls" s'occupe de gérer les inputs du joueur et de les mettre en interaction avec les deux autres parties.

Le diagramme de classe du modèle, et des diagrammes de séquences correspondant à des cas particuliers se trouvent en annexe dans l'email (plus de facilité de formattage) Le diagramme de séquence "1" correspond à l'attaque du player sur un monstre, et le "2" un déplacement sur une cas de type "Floor".

3 Fonctionnalités

3.1 Contenu

Génération procédurale de donjons aléatoires On a élaboré un algorithme de génération de donjons aléatoires mais cohérents, cet algorithme construit une grille d'entier uniforme dans laquelle on sculpte la structure du donjon. En ajoute ensuite les monstres et les coffres. (Une description plus précise est disponible dans la documentation du code source, package com.engine, classe Algo)

Difficulté croissante Le jeu est basé sur un système de cartes aléatoires se succédant. Une carte est terminée lorsque un totem est trouvé et activé le joueur passe au niveau suivant (stage). Le stage influence la quantité d'ennemis en présence et leur quantité de points de vie.

Evolution et progression du personnage A chaque ennemi vaincu, des points d'expériences sont gagnés, après un certain seuil le joueur monte d'un niveau. La montée de niveau augmente le nombre maximal de points de vie. Le joueur a également des points de compétences (Skills), de force (STRENGTH), vitesse (SPEED), magie de destruction (DESTRUCTION), et magie de soin (HEAL). Respectivement la force influence la capacité de l'inventaire, la vitesse la vitesse de déplacement du joueur, alors que les compétences de magie débloquent des sorts dans les styles respectifs (augmenter en destruction après déblocage des sorts augmente la portée des sorts d'attaque à distance, et les dégâts portés aux ennemis). Les compétences sont augmentées en utilisant des livres, trouvés dans les coffres ou laissé sur le sol par les ennemis vaincus.

Présence d'objets divers, équipements et "loot" Quatre types d'objets ont été créés, les armures et les armes représentent l'équipement, les armures diminuant les dégâts portés par les ennemis, et les armes augmentant les dégâts portés par le joueur. Les deux autres types sont les consommables (buffs) et les sorts (spells). Il y a trois types de consommables, les potions donnant un nombre précis de points de vie au joueur, les livres augmentant une compétence, et les plumes de phénix. Les plumes de phénix ressuscitant le joueur en cas de mort (il suffit de l'avoir dans son inventaire). Comme dit précédemment, les objets peuvent être ramassés après avoir tué un ennemi, ou trouvés dans les coffres (tous les loots sont aléatoires).

Système de sorts (attaques à distance) En jeu, si l'on équipe un sort, l'appui de la touche "S" le déclenche. Ce peut être soit un sort de régénération des points de vie, soit un sort d'attaque à distance. Leurs dynamiques ont été décrites plus haut.

3.2 Affichage

Animations et graphismes La plupart des actions du personnage sont animées, différentes animations pour des attaques différentes. L'affichage des déplacements de cellule en cellule est fluidifié (déplacements continus). Les coffres ont leur propre animation, ainsi que les ennemis. Le style des textures du donjon est modifiable via l'interface utilisateur (Deux versions, donjon "Classique", et donjon "Sephirot's Holyday").

Interface utilisateur Deux éléments d'aide visuelle ont été implémentés. La barre de vie, se situant en haut à droite de la fenêtre, et la console d'affichage de la progression (tracklog) permettant au jeu de communiquer l'évolution du joueur par messages concis. L'appui de la touche "O" ouvre la carte du joueur, affichant le stage (définissant la progression/difficulté) et l'arbre de compétences.

3.3 Utilitaires

Système de sauvegarde A tout moment, le joueur peut sauvegarder sa partie, sa position, ses points de vies et tous les autres paramètres sont enregistrés dans un fichier, et accessible depuis le menu principal. Le jeu supporte un nombre virtuellement illimité de sauvegardes, comme il prend le nom du joueur comme référence (à rentrer avant une nouvelle partie) .

Système de pause Quand le joueur ouvre son inventaire, le jeu se met en pause, lui permettant de prendre de décisions stratégiques en toute sérénité. (Les touches A et Z ont le même effet pour démonstration).

Intelligence artificielle Les ennemis cherchent le joueur, si il se trouve dans le champ de vision, se déplacent, et attaquent lorsqu'il est à portée.