Rapport de développement

Rotofight

par LAMBERT--DELAVAQUERIE Fabien
et CORSEAUX Axel

Travail effectué

Le programme initial demandé a été réalisé. Les points suivants sont fonctionnels :
- L'importation de niveaux dans des fichiers externes.
- Le rendu graphique du jeu (joueurs, murs, sourcils, épées).
- Les déplacements de chaque joueur.
- Le déplacement des épées, ainsi que le maximum d'épées lançables.
- La gestion des collisions entre épées, murs et joueurs.
- Les différentes gravités demandées.
- Le paramétrage du jeu.
- Le zoom.

Nous avons également réalisé une amélioration : le mode gravité 360.

Architecture

Description des paquets

Le programme comporte plusieurs paquets. En voici les noms et une courte description pour chacun d'entre-eux :
- Main : ne contient que la classe Main, soit le point d'entrée et la boucle principale du jeu.
- Read : contient les classes de gestion des entrées, soit la gestion des fichiers, des arguments du jeu et du clavier.
- World.Element : contient les classes des éléments affichés dans le jeu, soit le joueur, les murs et les épées.
- World.Environment : contient les entités invisibles du jeu, soit la gravité, les constantes et méthodes déterminant le déroulement du jeu.
- World.Event : contient les classes d'événements du jeu, soit une classe pour les collisions et une classe de gestion du temps écoulé entre chaque rendu.
- World.View : contient la classe de rendu graphique (gérant également le zoom).

Description des classes

fr.umlv.rotofight.Main :

Main.java : classe non-instanciable. Entrée du programme. Initialise toutes les variables et constantes du jeu, et contient la boucle principale du jeu.

fr.umlv.rotofight.Read :

Command.java : classe non-instanciable. Ensemble de méthodes permettant de gérer les paramètres du programme.

Input.java : classe non-instanciable. Méthodes de gestion du clavier et des actions en découlant.

Map.java : classe instanciable. Représente un niveau de jeu, préalablement chargé par la classe ReadMap.

ReadMap.java : classe non-instanciable. Ensemble de méthodes permettant le chargement d'une instance de Map par le biais de fichiers (chargement d'un niveau du jeu).

fr.umlv.rotofight.World.Element :

Element.java : classe instanciable. Représente un objet possédant une Position.

Player.java : classe instanciable. Représente un joueur.

Position.java : classe instanciable. Représente une position (x, y) dans la fenêtre de jeu.

Sword.java : classe instanciable. Représente une épée.

Swords.java : classe instanciable. Représente plusieurs épées (plusieurs instances de Sword).

Wall.java : classe instanciable. Représente un mur.

Walls.java: classe instanciable. Représente l'intégralité des murs d'un jeu.

fr.umlv.rotofight.World.Environment :

Constants.java : classe instanciable. Contient toutes les constantes du jeu (taille des joueurs, taille des murs, largeur de fenêtre, hauteur de fenêtre…).

Gravity.java : classe instanciable . Représente la gravité du jeu.

World.java : classe non-instanciable. Contient les méthodes permettant de faire bouger tous les objets du jeu (joueurs et épées), de vérifier les vies de chaque joueur et de terminer le jeu.

fr.umlv.rotofight.World.Event :

Collision : classe non-instanciable. Méthodes de gestion des collisions entre chaque entité du jeu (joueurs, murs, épées).

RenderTimer : classe non-instanciable. Comporte une méthode permettant de stopper le jeu un certain nombre de millisecondes (pour limiter les images par seconde).

fr.umlv.rotofight.View :

Draw.java : Interface. Les classes implémentant cette interface doivent contenir une méthode drawElement permettant de les afficher dans la fenêtre. Contient une méthode permettant l'affichage de tous les éléments du jeu.

Héritage, interfaces et précisions complémentaires

Les classes précédemment décrites comme « non-instanciables » possèdent un constructeur privé. Elles ne sont constituées que de méthodes statiques.

Les classes Player, Walls et Swords implémentent Draw, puisque ce sont les éléments affichables du jeu.

Les classes Player, Wall et Sword héritent de Element, qui représente n'importe quel élément du jeu ayant une position en x et en y.

Améliorations depuis la bêta

Nous avons ajouté les différentes gravités, l'animation des sourcils ainsi que le mode gravité 360. Le zoom a été ajouté au rendu.

Nous avons pour cela pris compte des remarques de la soutenance de bêta : la gravité s'opère donc sur l'axe x et y, contrairement à la bêta où la force de la gravité n'était appliquée que verticalement. Nous avons également veillé à ce que chaque getter émette une copie défensive lorsque l'entité demandée est un objet. Nous n'avons cependant pas réussi à délier les touches de l'objet Player : les touches du joueur sont donc stockées directement dans la classe Player et non dans la classe Input.