

Manuel de développement du projet Rotofight – Java

LOPES MENDES Ailton – LIN Gérald

Structure du projet :

Notre projet est découpé de manière à faciliter les implantations des objets.

De manière globale :

- La classe « **Entity** » : une classe abstraite « entité » qui spécifie une forme (*coordonnées*), une direction par défaut (*la direction "regardée"*), et la couleur de l'entité.
Les murs, joueurs, et épées sont considérés comme des entités.
- La classe « **MobileEntity** » : une classe abstraite pour les entités mobiles, spécifiant leur vitesse.
Les joueurs et épées sont considérés comme des entités mobiles.
- La classe « **MultiEntity** » : instancie un objet avec une liste d'entités.
Cette liste ne peut contenir qu'un seul type d'entités, c'est-à-dire qu'une même liste ne peut pas contenir un joueur et une épée, ces deux types d'entités seront dans deux listes distinctes.
Elle n'accepte pas les doublons.

Les classes suivantes sont complémentaires aux entités et leurs permettront de gérer leur déplacement, leur direction, etc :

- La classe « **Commands** » : est un objet qui correspond à une touche du clavier.
- La classe « **Controls** » : est un objet qui correspond à une action possible d'une entité mobile.
Elle permet par exemple de changer la direction "regardée" par l'action de se déplacer dans une direction différente de celle par défaut.
- La classe « **SetCommands** » : est une liste qui associe chaque touche instanciée avec l'action qui lui correspond.
- La classe « **Direction** » : une classe énumération qui contient des coordonnées d'un déplacement en fonction d'une direction.
Par exemple pour la direction « *East* », les coordonnées sont $[x = 1; y = 0]$, ce qui permet, lors d'un déplacement d'une entité mobile (*dans une autre classe*) d'utiliser ces coordonnées pour le déplacer.
- La classe « **Move** » : cette classe gère les déplacements des entités mobiles, que ce soit en fonction de la direction de ceux-ci ou de la gravité du monde.

Par la suite, grâce aux classes d'entités, il est alors plus simple de créer les classes suivantes :

- La classe « **Player** » : cette classe instancie un objet *joueur (entité mobile)*, qui contient ses touches pour les actions, ses épées (*une liste d'épées, ce qui permet de limiter son nombre*), son nombre de points de vie.
- La classe « **Sword** » : cette classe instancie un objet *épée (entité mobile)*, qui spécifie ses dégâts et une direction dans laquelle il avancera jusqu'à un obstacle. Une épée est associée à un joueur, ainsi il est plus simple de différencier les épées tirées entre tous les joueurs.
- La classe « **Wall** » : instancie un objet *mur*.

Une fois que les entités sont présentes, il ne manque plus que les classes créant un monde pour contenir ces entités :

- La classe « **Gravity** » : cette classe permet le calcul des coordonnées d'une entité mobile en fonction de la gravité présente dans le monde.
- La classe « **MultiGravity** » : est une liste des gravités utilisées pour le monde, ce qui permet d'effectuer la rotation des gravités.
- La classe « **World** » : cette classe contient une liste des joueurs présents dans le monde, une liste des murs et la liste des gravités utilisées.

Améliorations/corrections apportées depuis la soutenance bêta :

Notre structure étant quasiment ce qui était attendu, nous n'avons pas eu grand chose à corriger dans notre projet.

Entre temps, nous avons ajouté les tirs d'épées, la gestion des dégâts, les différentes gravités.

Bugs connus :

- Le zoom n'est pas fonctionnel.