

Projet de POO

Réalisation d'un jeu vidéo 2D : UBomb.

MOUCHRIF Dounia
ABARKAN Suhaila

Ce qui ne marche pas :

GESTION DES MONDES

Lorsque nous ouvrons la porte avec la clé et que le player va sur la porte, nous n'avons pas réussi à le faire passer au level supérieur. Nous avons pensé à charger une nouvelle Map associée au level correspondant lorsque le player se trouve sur une porte ouverte mais nous ne sommes pas parvenues à mettre notre idée en œuvre.

Nous n'avons pas réussi à créer une game et charger un monde depuis la lecture d'un fichier. Dans la classe GameLauncher nous avons mis en commentaire la methode public *Game load(File file)* qui ne fonctionne pas. Nous voulions l'appeler dans la classe GameLaucherView dans la partie "*loadItem.setOnAction*" mais la méthode ne fonctionnant pas nous avons mis cette partie en commentaire.

Nous avons implémenté dans la classe GameLaucher les méthodes suivantes qui fonctionnent : *MapLevel load(String)* (permet de renvoyer une Map à partir d'une String compressé) et *String export(MapLevel mapLevel)* (permet d'exporter une Map en une String compressé). Nous avons pensé à les appeler dans la méthode *private MapLevel StringProperty(Properties config, String name, MapLevel defaultValue)* afin de récupérer la Map pour chaque level mais cela n'a pas fonctionné nous avons du mal à trouver comment les appeler et si c'était bien le bon endroit.

Ce qui marche :

MONSTRES

Les monstres (au nombre de 3 pour la Map par défaut) sont ajoutés avec des trajectoires et déplacements aléatoires. Une collision du joueur avec ces derniers entraînent la perte d'une vie du joueur. Ils ne peuvent pas ramasser les bonus qui se trouvent sur le sol. Ils ne peuvent pas marcher sur des portes. Ils ne peuvent pas déplacer les caisses.

PANNEAU D'INFORMATIONS

Toutes les informations sont correctement renseignées et se mettent à jour au fil du jeu.

GESTION DES PORTES

Du fait qu'on situe au level 1 pour notre jeu, il est affiché qu'une porte fermée.

Lorsque le joueur se situe à côté de la porte fermée ET qu'il la regarde, il appuie sur la touche [ENTER] pour pouvoir l'ouvrir, s'il possède au moins une clé dans son inventaire.

Une fois utilisée, elle disparaît de l'inventaire.

GESTION DES BONUS ET MALUS

Le joueur ramasse automatiquement un bonus lorsqu'il marche sur la case qui le contient. Les monstres peuvent marcher sur les cases des bonus, mais ne peuvent pas les ramasser.

DÉPLACEMENT DES CAISSES

Les caisses peuvent être déplacées par le joueur. Le joueur ne peut déplacer qu'une seule caisse à la fois. Si un bonus ou un monstre se trouve dans la direction de déplacement d'une caisse, la caisse reste bloquée.

GESTION DES BOMBES

La bombe est déposée quand le joueur presse la touche [ESPACE] déclenchant une explosion au bout de 4 secondes. La mèche de la bombe diminue chaque seconde. La portée de la bombe ne peut être de moins d'une case.

Les éléments de décor stoppent bien la propagation de l'explosion dans le sens qu'ils obstruent. Une explosion ne peut détruire qu'une seule caisse dans une même direction. Si un bonus (ou un malus) se trouve sur le chemin de l'explosion, il disparaît.

Enfin, si le joueur est sur une cellule touchée par une explosion, il perd une vie. Le monstre meurt (car il n'a qu'une vie).

Les explosions n'ont aucun effet sur les portes et les clés. Lorsqu'une bombe explose, une nouvelle bombe est ajoutée à l'inventaire du joueur.

GESTION DES VIES

Le joueur peut perdre une vie s'il se trouve sur une case à portée de l'explosion d'une bombe ou s'il croise un monstre. Le joueur bénéficie alors d'une temporisation de `playerInvicilityTime` (de 4sec) pendant laquelle il est invulnérable.

FIN DE PARTIE

La partie est finie lorsque le joueur arrive sur la case de la princesse, ou si le joueur n'a plus de vie.

Spécificités du code :

- Les positions des monstres au démarrage du jeu sont renseignées dans "game.Configuration" sous forme d'un tableau de positions.
- Le joueur ne prend pas le malus qui diminue la portée des bombes lorsque la portée vaut 1.
- Les caisses sont représentées comme des éléments de décor. Lorsque le player déplace une caisse, celle-ci est en fait supprimée et une nouvelle caisse est créée aux bonnes coordonnées.
- Pour coder la méthode "engine.GameEngine.createNewBombs", nous nous sommes inspirées du code de la méthode "engine.GameEngine.animateExplosion".
- Pour coder la méthode "engine.GameEngine.checkExplosions", lors de l'explosion, on vérifie que les éléments de Décor bloquent la portée de la bombe, pour chaque trajectoire de la portée de la bombe.