

Projet Logiciel Transversal

Juliette COME - Charly BION

Table des matières

1	Présentation Générale	1
1.1	Archétype	1
1.2	Règles du jeu	1
1.3	Ressources	2
2	Description et conception des états	3
2.1	Description des états	3
2.2	Conception Logiciel	3
3	Rendu : Stratégie et Conception	4
3.1	Stratégie de rendu d'un état	4
3.2	Conception logiciel	4
4	Règles de changement d'états et moteur de jeu	5
4.1	Règles	5
4.2	Conception logiciel	6
5	Intelligence Artificielle	6
5.1	Stratégies	6
5.2	Conception logiciel	7
6	Modularisation	7
6.1	Organisation des modules	7
6.2	Conception logiciel	8

1 Présentation Générale

1.1 Archétype

Notre projet sera basé sur le jeu "slay the spire". C'est un jeu de deck building, rogue-like, pour le moment encore en early access.



FIGURE 1 – Jeu slay the spire

1.2 Règles du jeu

Dans ce jeu, le joueur commence avec un deck basique de départ, puis progresse de salle en salle jusqu'à arriver au boss final. Il peut compléter son deck de cartes en fonctions des salles qu'il croise.

- Les salles sont divisées en deux groupes majeurs : les salles d'ennemis, plus fréquentes, et les salles spéciales. Entrer dans une salle d'ennemis déclenche un combat entre le joueur et les ennemis présents dans la salle. Les salles spéciales peuvent avoir des effets positifs ou négatifs sur le joueur. Les effets peuvent être par exemple acquérir une carte puissante, ou obtenir un bonus ou un malus pour le combat suivant.
- Le joueur et les ennemis possèdent un nombre de point de vie et une force précise. La puissance des ennemis augmentera avec le nombre de salles parcourues. Vaincre un ennemi consiste à réduire ses points de vie à 0. De même, le joueur perd la partie si sa vie est réduite à 0.
- Le deck sera limité à 15 cartes, chaque carte ajoutée doit en remplacer une autre si cette limite est atteinte.
- A chaque tour, le joueur possède un taux précis d'énergie et pioche une nouvelle main de 5 cartes. Le joueur peut jouer ses cartes en dépensant de l'énergie, et finir son tour à tout moment. Toutes les cartes non jouées sont défaussées et sont mises dans la pile de défausse. Au tour suivant, toute l'énergie du joueur est régénérée. Il pioche une nouvelle main de 5 cartes issues de sa pioche. Lorsque sa pioche est vide, la défausse du joueur est mélangée et forme la nouvelle pile de pioche. Pendant le tour du joueur, il peut voir les actions qui seront réalisées par les ennemis. Les ennemis jouent toujours leurs actions après le joueur.
- Les cartes peuvent être piochées, jouées ou défaussées. Elles dépensent de l'énergie lorsqu'elles sont jouées. Elles peuvent être utilisées pour attaquer l'ennemi, ajouter du block au joueur, ou réaliser d'autres actions plus spécifiques.
- Les cartes et les ennemis possèdent un élément précis (air, eau, terre, feu). Certains éléments

seront plus puissants contre d'autres et inversement. Les cartes pourront également apporter des bonus et malus au joueur et aux ennemis.

- Les ennemis relâchent des cartes après qu'ils soient vaincus, que le joueur peut ajouter à son deck. Ces cartes augmentent en puissance en fonction du nombre de salles parcourues.
- Le jeu est découpé en étages et en salles. Chaque étage se termine par un combat de boss, un ennemi plus puissant que les autres, et représentant un élément spécifique. Chaque boss achevé permet la progression du joueur vers l'étage suivant, sauf pour le dernier boss du jeu. Quatre étages, représentant chacun des éléments définis, sont prévus.

1.3 Ressources

Ressources nécessaires :

- Images des ennemis et des joueurs
- Fonds pour les salles
- Images des cartes à jouer : fond de carte, illustration, éléments spécifiques comme l'énergie ou l'élément.
- Icones pour les bonus/malus, énergie, éléments et événements.
- Fond de la carte, images représentant les différents types de salles.

2 Description et conception des états

2.1 Description des états

2.2 Conception Logiciel

3 Rendu : Stratégie et Conception

3.1 Stratégie de rendu d'un état

3.2 Conception logiciel

4 Règles de changement d'états et moteur de jeu

4.1 Règles

4.2 Conception logiciel

5 Intelligence Artificielle

5.1 Stratégies

5.2 Conception logiciel

6 Modularisation

6.1 Organisation des modules

6.2 Conception logiciel