

Projet Logiciel Transversal

Les Dee & Di

AYROLES Quentin, EL BÈZE Ilan, MAERTEN Eloïse
2021 - 2022

Table des matières

1	Présentation Générale	2
1.1	Archétype	2
1.2	Règles du jeu	2
1.3	Ressources	2
2	Description et conception des états	5
2.1	Description des états	5
2.1.1	Éléments fixes	5
2.1.2	Entités	5
2.2	Conception Logiciel	6
3	Rendu : Stratégie et Conception	7
3.1	Stratégie de rendu d'un état	7
3.2	Conception logiciel	7
4	Règles de changement d'états et moteur de jeu	8
4.1	Règles	8
4.2	Conception logiciel	8
5	Intelligence Artificielle	9
5.1	Stratégies	9
5.2	Conception logiciel	9
6	Modularisation	10
6.1	Organisation des modules	10
6.2	Conception logiciel	10

1 Présentation Générale

1.1 Archétype

L'objectif de ce projet est la réalisation d'un jeu de rôle (RPG) classique inspiré par Donjons & Dragons.

1.2 Règles du jeu

Le joueur déplace ses personnages sur une carte (une grille). Son but est de réaliser sa quête (définie au début de la partie). Sur la carte, le joueur peut être confronté à des ennemis, événements et réaliser diverses actions. Ces actions auront une part d'aléatoire pour imiter les lancer de dés du jeu original. À terme, l'idée est d'aussi permettre au joueur de collecter des objets et d'améliorer ses personnages. Le mode multijoueur sera un mode coopératif où les personnages seront répartis entre les joueurs.

1.3 Ressources

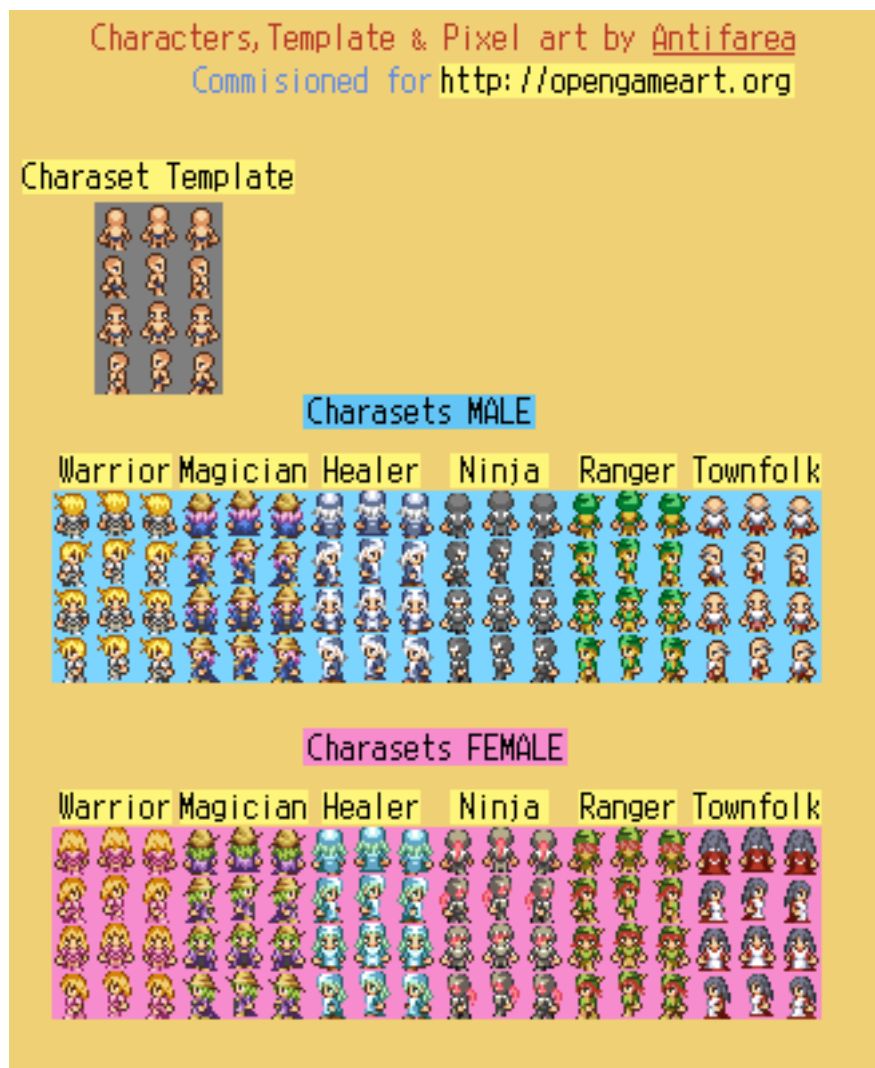


FIGURE 1 – Textures des personnages



FIGURE 2 – Police d’écriture



FIGURE 3 – Textures des ennemies



FIGURE 4 – Textures des décors

2 Description et conception des états

2.1 Description des états

Un état de jeu est constitué d'un tableau d'éléments représentant la grille du jeu (donc les murs, le sol...) et un autre tableau avec les personnages (appelé entité, qui peuvent à la fois être des héros ou des ennemis). On note aussi dans l'état de jeu le nombre de tour déjà joué, ainsi que l'ordre de jeu des entités et l'entité dont c'est le tour.

2.1.1 Éléments fixes

La grille du plateau de jeu est constitué d'un tableau d'éléments de la classe "Décor". Un décor peut avoir un seul type parmi les suivants :

- None : état passif, la case "n'existe pas" pour le joueur
- Mur : élément infranchissable et un obstacle visuel (on ne peut ni voir ni tirer au travers par exemple)
- Sol : éléments de base sur lequel le joueur peut se déplacer
- Porte : élément infranchissable comme le mur nécessitant une action pour s'ouvrir et se transformer en sol
- Secret : case porte ayant une texture de mur pouvant se révéler après une certaine action
- Piège : case sol infligeant des dégâts à une entité se déplaçant dessus
- Trésor : case sol comportant un objet (ou équipement) que le joueur peut ramasser, elle peut donc devenir une case de type sol après la récupération de l'objet

Ces éléments de décor ont donc besoin d'une méthode leur permettant de passer de leur état de "cases spéciales" (trésor, piège...) à un état de case normale tout en effectuant l'action qui leur est associé (donner un équipement pour un trésor ou infliger des dégâts pour un piège).

Chaque case est associé à un objet de la classe environnement qui définit à la fois la luminosité ambiante et la météo (neige, pluie, orage...).

Un autre argument nous permet de choisir la texture parmi une liste de textures prédéfinie nous permettant de faire varier les décors (forêt, intérieur du donjon...).

2.1.2 Entités

Les entités sont les personnages du jeu, qu'ils soient héros (personnages interprétés par le ou les joueurs) ou ennemis (personnages contrôlés par l'IA). Une entité possède dans sa classe toutes ses caractéristiques, comme :

- niveau
- points de vie restants
- points de magie restants
- ses différentes statistiques comme les points d'attaque, de défense, de vitesse
- l'équipement qu'il possède comme des armes
- les statuts qu'il subit (gelé, confus...)
- les actions supplémentaires qu'il peut effectuer en plus de ses actions de base comme les sorts

L'ordre de jeu des entités est définie selon leur statistique d'initiative. Elle peut, durant son tour, se déplacer (selon sa statistique de déplacement) ou attaquer (selon sa statistique d'attaque). Une action supplémentaire remplace l'attaque dans son tour. Lorsque l'entité est attaquée elle se défend (selon sa statistique de défense) puis subit des dégâts et obtient possiblement un statut. Nous avons également une méthode correspondant à la mort de l'entité.

Du côté des différences entre ennemis et héros, les héros ont, en plus de ces caractéristiques, une classe (qui définit leurs statistiques de base et leur texture) et la possibilité de récupérer de l'équipement ou apprendre des actions supplémentaires comme des sorts. Les ennemis ont une race, équivalente à la classe,

qui définit donc leur statistiques de base et leur textures. Pour éviter que, pendant un tour, un ennemi soit pris en compte alors qu'il n'est pas dans la même salle que les joueurs et donc pas accessible, on lui rajoute un attribut "actif" qui définit si les ennemis sont en jeu ou non. Cet attribut passe de inactif à actif lorsque le ou les joueurs sont suffisamment proche de lui.

2.2 Conception Logiciel

Le diagramme des classes pour les états est présenté en Figure 5, dont nous pouvons mettre en évidence les groupes de classes suivants :

Classes élément : les classes Entité, Décor, Héros, Ennemi et Équipement représentent les différentes catégories et types d'élément. Les éléments peuvent être séparés en trois catégories. Les éléments mobiles (Héros et Ennemi), les éléments fixes (Décor) et les éléments invisibles (équipement). Ces derniers dépendent forcément d'une Entité et ne peuvent pas exister par eux même.

Conteneurs d'élément : les classes State et ElementTab permettent de contenir des ensembles d'éléments. ElementTab est un tableau en deux dimension d'éléments, par exemple pour contenir la grille des éléments du niveau. Il peut également être considéré comme un tableau à une dimension dans le cas où la hauteur (height) est égale à 1 (utile pour la liste des personnages). Enfin, la classe State est le conteneur principal, à partir duquel on peut accéder à toutes les données de l'état.

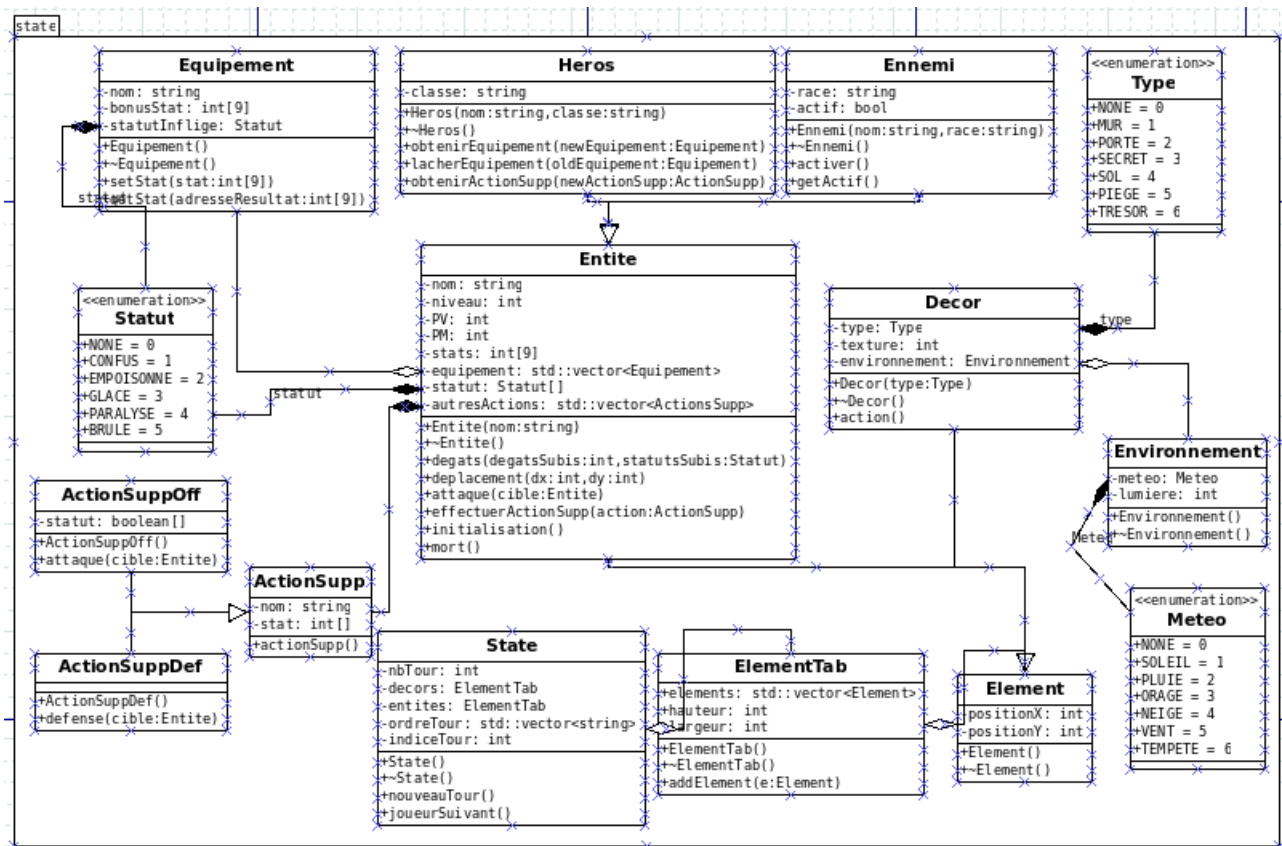


FIGURE 5 – Diagramme des classes d'état

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