

# Projet Logiciel Transversal

Vincent AYME-Valentin LAURENT

# Table des matières

## **1. Objectif**

- 1.1 Présentation générale
- 1.2 Règles du jeu
- 1.3 Conception Logiciel

## **2. Description et conception des états**

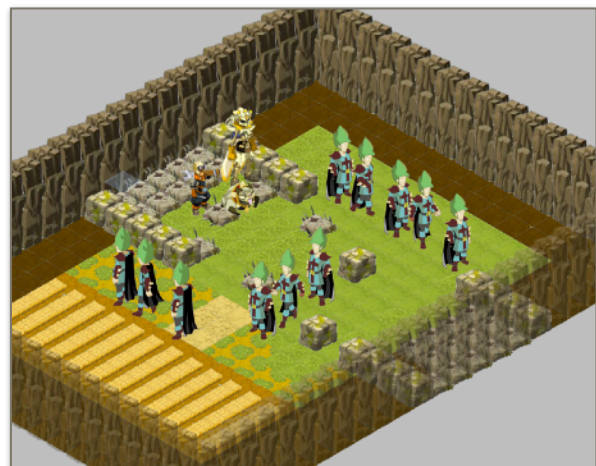
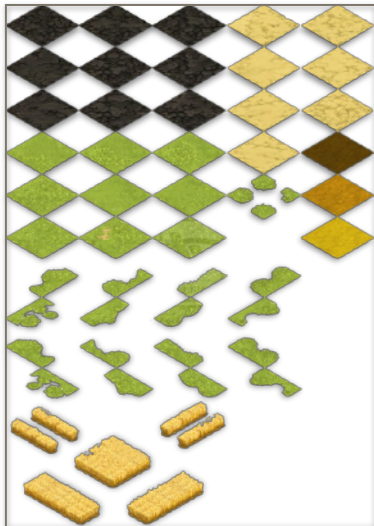
- 2.1 Description des états

# 1 Objectif

## 1.1 Présentation générale

Notre projet se base sur le système de combat tour par tour du jeu vidéo Dofus. Le joueur aura la possibilité de choisir entre plusieurs classes de personnages avec des aptitudes différentes telles que Iop, Sadida ou Sram. Le jeu débutera sur une carte proposant différents donjons. Chaque donjon entraînera une série de combats (le joueur versus une IA). Durant chaque affrontement, le joueur possèdera, à chaque tour, des points de déplacements ainsi que des points d'action lui permettant de se déplacer dans l'arène et d'invoquer ses différentes attaques.

Voici un aperçu des textures graphiques que nous allons utiliser.



## 1.2 Règles du jeu

- Le joueur, au début du jeu, choisit sa classe qui lui confèrera certains avantages en fonction des éléments de chaque donjon.
- Le joueur doit triompher de l'ensemble des donjons afin de terminer le jeu.
- Le joueur doit finir un donjon pour débloquent le suivant.
- A chaque arène terminée, le joueur recevra de l'expérience qui lui permettra de monter différents niveaux tout au long du jeu.
- Une fois qu'une arène est terminée, le joueur aura une probabilité de recevoir de l'équipement additionnel.
- Le joueur dispose à chaque tour de points de déplacement et des points d'action qui sont réinitialisés à chaque début de tour (sauf malus) lui permettant de se déplacer dans l'arène et d'utiliser des compétences.
- Si le joueur est vaincu lors d'une arène d'un donjon, celui-ci devra recommencer le donjon depuis la première arène pour le terminer.
- (Optionnel) Possibilité de terminer un donjon à plusieurs joueurs, en jouant en réseau.
- (Optionnel) Possibilité d'affronter un autre joueur en réseau.

## 1.3 Ressources

## 2. Description et conception des états

### 2.1 Description des états

Un état du jeu est formée par un ensemble d'éléments fixes (arène) et un ensemble d'éléments mobiles (personnage joueur et adversaire). Tous les éléments possèdent les propriétés suivantes:

- Coordonnées (x,y) sur l'arène
- Statistiques (Liste[PV (Points de Vie), PA (Points d'Action), PM (Points de Mouvement)])
- Compétences (propres à la classe du personnage): ensemble des actions que peut faire l'élément
- Equipement: Ensemble des objets dont le personnage est équipé
- Identifiant de type d'élément (nombre indiquant la nature de l'élément: IA ou joueur)

#### 2.1.1 Etat éléments fixes

Le labyrinthe est formé par une grille d'éléments nommé « cases ». La taille de cette grille est fixée au démarrage du niveau. Les types de cases sont :

**Cases "Obstacle"** Les cases "obstacle" sont des éléments infranchissables pour les éléments mobiles, elles correspondent à la fois aux frontières d'une arène mais également à certaines cases au sein de celle ci. Le choix de la texture aura une influence sur l'évolution du jeu car certaines peuvent être traversées par des compétences lors d'un combat. On considère les types de cases "obstacle" suivants:

- Les obstacles "mur", qui définissent les contours d'une arène ou un obstacle qui ne peut être traversés par une attaque.
- Les obstacles "décors", qui peuvent être traversés par une attaque d'un joueur.

**Cases "Espace"** Les cases "espace" sont les éléments franchissables par les éléments mobiles. On considère les types de cases "espace" suivants:

- Les espaces "vides"
- Les espaces "départ", qui définissent les positions initiales pour le joueur et l'adversaire.
- Les espaces "piège", qui déclenchent une action lorsqu'un personnage marche dessus.

### 2.1.2 Etat éléments mobiles

*Les éléments possèdent une direction (gauche, droite, haut ou bas) et une position.*

**Element mobile “Personnage”** Cet élément est soit dirigé par le joueur soit par l'IA qui, en fonction du nombre de points de déplacement, commande la propriété de direction. On utilise une propriété que l'on nommera “statut”, et qui peut prendre les valeurs suivantes:

- Statut “normal”: cas le plus courant, le personnage peut se déplacer. lors de son tour de jeu.
- Statut “empoisonné”: cas où le personnage a subi une certaine attaque. Ses points de vie diminueront légèrement à chaque début de tour pendant un nombre de tours définis avant de repasser au statut “normal”.
- Statut “mort”: Les points de vie du personnage sont inférieurs ou égale à 0, le personnage ne peut plus bouger ni réaliser aucune action.

### 2.1.3 Etat général

*A l'ensemble des éléments statiques et mobiles, nous rajoutons les propriétés suivantes:*

- Tour** : Permet à un personnage d'effectuer ses actions et déplacements.
- Compteur d'ennemis en vie** : le nombre d'ennemis qu'il reste à vaincre avant de terminer une arène.
- Timer** (Optionnel): Délimite le temps passé pour un tour de jeu lorsque deux joueurs humains s'affrontent.

## 2.2 Conception générale

Le diagramme des classes pour les états est présenté sur la figure suivante, nous pouvons mettre en évidence les groupes de classes suivants:

**Classe Elements** Toute la hiérarchie des classes filles de “Element” (en jaune), permet de représenter aussi bien les personnages (joueur ou IA) que les éléments présents sur une map, tels que les cases où un personnage peut se déplacer, où il commence le combat ou les cases occupées par un décors.

**Conteneurs d'Elements** On peut s'apercevoir que les classes “State” ainsi que “ElementTab” viennent contenir l'ensemble des éléments sur une carte donnée. Une instance du type “ElementTab” va donc être une liste de l'ensemble des objets présents sur la map tels que les éléments de décors et les personnages.

