

Projet logiciel TRANSVERSAL



29 septembre 2016

Encadré par : GOSSELIN Philippe

GRANIER Bernard

Auteurs : Fady ZAMBA-ZAMBA

Ramzi Mouad

Table des matières

[1 Objectif 3](#RefHeading__1316_824097905)

[1.1 Présentation générale 3](#RefHeading__1318_824097905)

[1.2 Règles du jeu 3](#RefHeading__1320_824097905)

[1.3 Nos ressources 3](#RefHeading__3406_1053578290)

2 Description et conceptions des états 3

2.1 Description des états 3

2.1.1 Etat éléments fixes 4

2.1.2 Etat élément mobile 4

2.2 Conception logiciel 4

2.2.1 Diagramme des classes 4

2.2.2 Description des classes 5

# **1. Objectif**

## **Présentation générale**

Ce projet consistera à réaliser un jeu de combat de type JRPG (voir figure d’illustration).

## **1.2 Règles du jeu**

Tout d'abord, le joueur commence son tour en choisissant une attaque. Une fois l'attaque choisie, le personnage la mettra en œuvre. L'adversaire (IA), suite à cela, va riposter à son tour en lançant une attaque. Il faut noter que le joueur pourra également utiliser des attaques spéciales (qui correspondent à des dommages plus importants) ; ces dernières ne pourront être réutilisées qu’après plusieurs tours. Les barres de vies détermineront ensuite le vainqueur (le personnage ayant une barre de vie vide sera considéré comme le perdant).

## **1.3 Nos ressources**

Nous avons tout d’abord eu besoin de choisir des « Spritesheets » qui permettront par la suite de réaliser des animations sur les personnages.

Ensuite, il a fallu déterminer une image de fond, afin de représenter la zone environnante des deux combattants. Nous avons retravaillé l’image de fond afin que celle-ci s’anime lors des combats.

Enfin, nous avons rajouté quelques fichiers sonores afin de rendre le jeu plus vivant (musique d’introduction, bruits des coups donnés, etc…).

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

## **2.1. Description des états**

Un état du jeu est formé par un ensemble d’éléments fixes (Background, Gauge, Time) et un ensemble d’éléments mobiles (Player).

## **2.1.1. Etat éléments fixes**

**Element fixe « Background ».** Le background constituera l’environnement dans laquelle les deux combattants s’affrontent. C’est au joueur de choisir le décor Ainsi, il aura le choix entre plusieurs textures (un combat se déroulant dans une pièce, ou un combat se déroulant en plein air par exemple).

**Element fixe « Gauge».** Par dessus le Background se trouveront deux jauges:

* Jauge de vie : une zone rectangulaire initialisée par une couleur verte. A chaque coup subit, le rectangle diminuera en longueur(diminution des point des vie).
* Jauge d’expérience : une zone rectangulaire, mais cette fois-ci qui se charge en fonction du nombre de coups émis. Une fois la jauge chargée, le joueur pour activer une attaque surpuissante. Une fois l’attaque lancée, la jauge se videra. (le joueur devra réitérer les mêmes actions pour recharger la jauge).

**Element fixe «Time».** Il y’aura ici un emplacement dédié au chronomètre afin de limiter la partie. Il faut savoir que deux combattants peuvent ne pas attaquer et ainsi se mettre en position défensive : le jeu durerait trop longtemps s’il n’y avait que des défenses lancées de chaque côté. Ainsi, à la fin du temps réglementaire, celui ayant le plus de point de vie sera considéré comme gagnant.

## **2.1.2 Etat élément mobile**

**Elément mobile « Player ».** Cet élément est dirigé par le joueur. Il sera caractérisé par une énumération. En effet, on lui attribuera plusieurs « statuts » (**NORMAL, SUPER, DEAD,** etc…) permettant de déterminer l’état dans lequel se trouve le joueur et ainsi réaliser les méthodes adéquates.

## **2.2 Conception logiciel**

## **2.2.1 Diagramme des classes**

Le diagramme des classes pour les états est représenté ci-dessous.

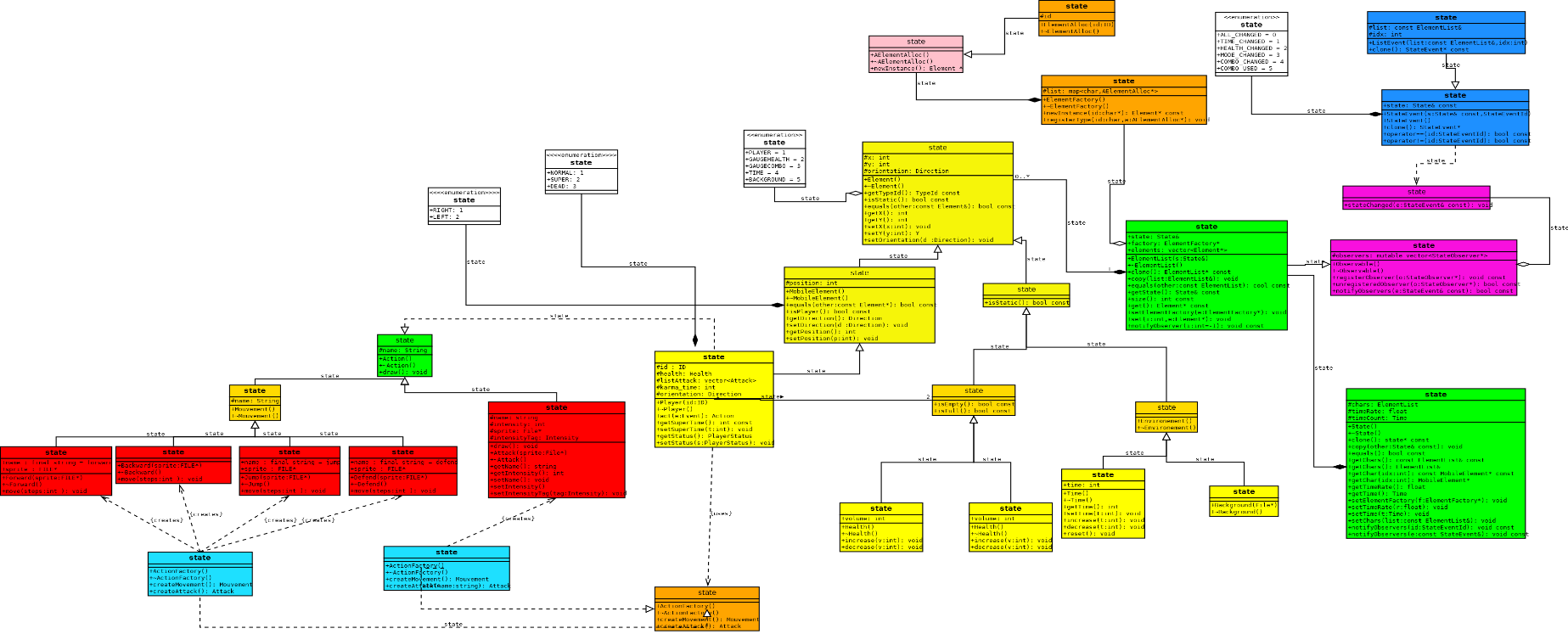


Diagramme des classes d’état

## **2.2.2 Description des classes**

Voici le tableau récapitulatif :

|  |  |
| --- | --- |
| **Element** | Plusieurs évènements interviennent en même temps, d’où la nécessité de constituer chaque élément. On distingue les éléments mobiles et statics. |
| **MobileElement** | Notre seul élément mobile est le personnage. Cependant, nous pourront par la suite ajouter d’autres éléments mobiles. |
| **StaticElement** | Nous y trouverons le décors, le chronomètre, et les barres de points de vie. |
| **Player** | Gestion du personnage. |
| **Action** | A travers cet élément, nous pouvons nous diriger vers deux évènements. Le premier qui serait la possibilité de réaliser un mouvement (Jump, Forward, Backward, Defend), et le deuxième, celui de réaliser une attaque (en terme de dégâts, nous avons les coups physiques et les coups magiques). |
| **Mouvement** | Représente tout types de déplacement (en avant, en arrière, saut, défense). |
| **Attack** | Représente tout types d’attaques (Corps à corps, à distance). |
| **ActionFactory** | Permet de fabriquer nos « actions ». |