

# Rapport SAé Java

Pour la 2eme partie de la SAé en Java, nous avons décidé de rajouter les fonctionnalités suivantes :

- Système de meilleur score
- Armes différentes avec une durabilité, des dégâts et des bonus selon l'arme équipé
- Monstre de type différents avec des avantages de types + héro de type aléatoire
- Passage d'un plateau 3x3 à un plateau 5x5
- Ajouter des illustrations aux différentes cases que nous pouvons rencontrer

## Utilisation :

Le jeu se déroule toujours de la même manière, aller le plus loin possible en faisant le meilleur score. Le passage au plateau en 5x5 augmente le nombre choix que le joueur peut effectuer et est plus agréable. Au début de chaque nouvelle partie, notre héro tout comme les monstres se voient attribuer un type parmi les suivants : Feu, Vent, Foudre, Terre, Eau, Lumière et Ténèbres.

Selon le type de notre héro et le type du monstre rencontré, nous infligerons ou subiront des dégâts supplémentaires qui peuvent nous aider à aller plus loin ou alors que la partie se termine rapidement.

Les monstres sont reconnaissables par leur couleur ainsi que les armes grâce aux images les définissant.

## Technicité :

Pour faire le système de meilleur score, nous avons créé une nouvelle classe MeilleurScore dans lequel nous allons gérer la lecture ainsi que l'écriture du meilleur score dans un fichier et qui va comparer le score de la partie actuelle avec le meilleur score dans le fichier pour savoir si le meilleur score change.

Pour afficher un plateau en 5x5, nous avons modifier la taille du GridLayout dans les classe VueCase et VuePlateau, et la taille du plateau dans le Controleur.

Pour afficher les images des monstres, nous avons mis des images dans un fichier et que l'on récupère grâce à un Toolkit.getImage et nous devons gérer quelle image allait se mettre pour quelle case (ex : image d'un monstre si c'est une case monstre etc).

Dans la classe monstre, nous avons créé un tableau type, dans le constructeur de monstre nous avons un random par rapport au nombre de type, puis nous avons un getType qui renvoie un this.nom qui contient le string du type que l'on va appliquer au monstre. Le fonctionnement est le même pour Héros.

Pour la durabilité de l'arme, de la hache et de la Carotte nous avons fait comme cela

```
this.dura = v - (rand.nextInt(v-2));
```

Il y a une particularité avec la hache, lorsqu'on tue un monstre avec, nous regagnons 2 PV, puis avec la Carotte, celle-ci est plus puissante mais si l'on tue d'un coup un monstre elle perd 3 de durabilité et on perd 1 de durabilité si le nombre de PV du monstre est supérieur aux dégâts infligés par la Carotte.

## Diagrammes :

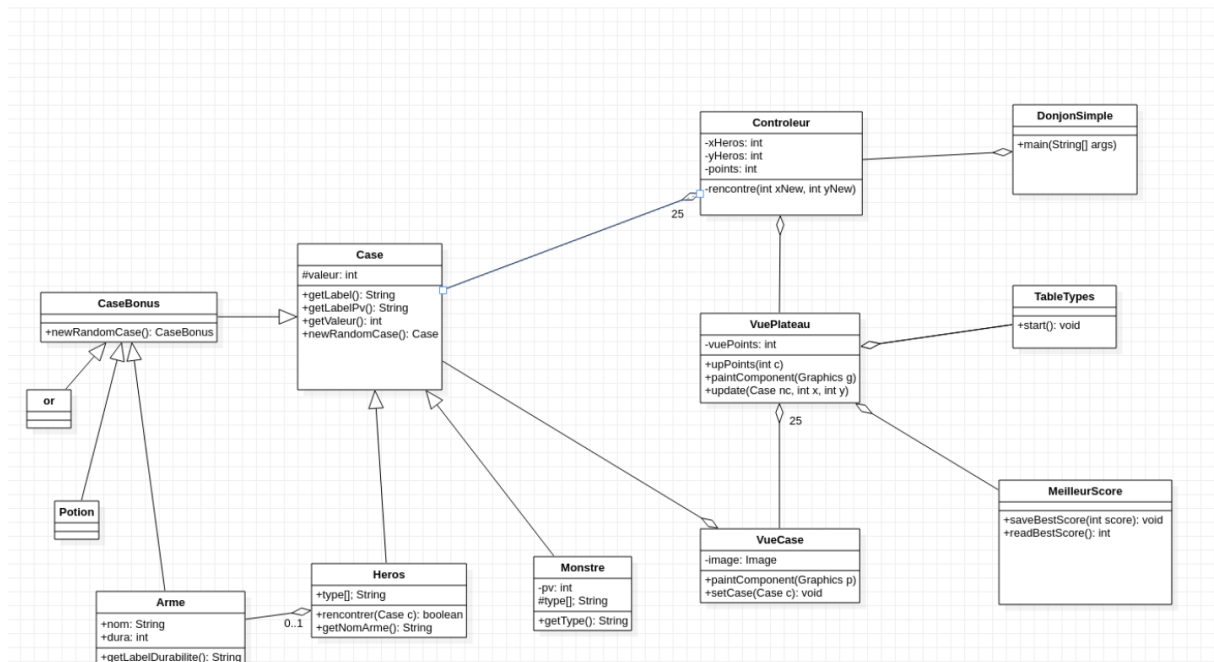


Diagramme de cas d'usage