

SAé 2.01 : Développement Orienté Objet

Donjon Infini

1 Sujet

L'objectif de cette SAé est d'implémenter un jeu simple d'exploration de donjon. Le projet se divise en deux parties :

- Dans la première partie, vous devrez coder le jeu de base avec les spécificités fournies par le sujet, détaillées dans la suite du document. Il y a notamment un Diagramme de cas d'usage centré sur le contrôleur (Figure 1), un diagramme de classes de l'application (Figure 2) ainsi qu'un diagramme séquence illustrant un tour de jeu (Figure 3).
- Dans un deuxième temps, vous serez amené.e.s à continuer le développement du jeu, en implémentant de nouvelles fonctionnalités de votre choix.

Vous serez évalué.e.s sur les deux rendus, ainsi que lors d'une soutenance où vous présenterez votre jeu et répondrez aux questions concernant votre code et vos choix d'implémentation.

1.1 Règles du jeu : partie 1

Le jeu se présente sous la forme d'un plateau de 3x3 cases.

Chacune de ces cases contient un élément qui peut être :

- Le Héros (ou Héroïne) : il/elle est unique sur le plateau, et possède des points de vie ainsi qu'éventuellement une arme.
- Un Monstre : c'est un adversaire du héros, il possède des points de vie.
- Une Arme : sa valeur correspond à son attaque.
- Une Potion : sa valeur correspond au nombre de points de vie récupéré lors de sa consommation.
- Un sac d'or : sa valeur correspond au nombre de points obtenus lors de sa prise.

Le jeu se déroule de la façon suivante. Le héros est initialement au milieu du plateau. Chaque autre case contient un élément choisi au hasard (Monstre 50%, Sac d'or 20%, Arme 17,5%, Potion 12,5%).

Le but est d'accumuler un maximum de points avant la fin de la partie, c'est-à-dire quand le héros meurt. Le héros peut se déplacer sur une case adjacente (haut, bas, gauche, droite) et déclencher un événement dépendant du type de case rencontrée :

Arme Le héros ramasse l'arme s'il n'en a pas ou si elle possède une valeur d'attaque supérieure à celle de son arme. Dans tous les cas, il gagne autant de points que la valeur de l'arme.

Potion Le héros gagne autant de points de vie que la valeur de la potion. La même valeur est gagnée en points.

Or La valeur du sac est ajoutée aux points.

Monstre — Si le héros n'a pas d'arme, il prend autant de dégâts que le nombre de points de vie du monstre.

- S'il a une arme, il attaque avec son arme et ne se déplace pas. Il endommage le monstre d'autant que la valeur de son arme, et l'arme perd autant de valeur que les points de vies du monstre. Si la valeur de l'arme passe à 0 ou moins, elle se casse et le héros perd son arme. Si le monstre est battu, un sac d'or apparaît de valeur égale au nombre de points de vie initiaux du monstre.

Dans tous les cas, à l'exception d'une attaque à l'arme, le héros effectue un déplacement, et un nouvel élément apparaît sur la case qu'il vient de quitter, selon les mêmes probabilités qu'à l'initialisation. Le héros peut alors se déplacer à nouveau. Un exemple d'un tour de jeu est donné par le Diagramme Séquence de la Figure 3.

Il est attendu une interface graphique lisible et claire utilisant la bibliothèque Swing de java, similairement aux TP du cours. Le score actuel du joueur devra être affiché sur l'interface à chaque tour, et le score final devra être affiché sur la sortie standard à la fin de la partie.

1.2 Extensions : partie 2

Dans la deuxième partie du projet, vous devez développer des extensions au jeu de base codé lors de la première partie. Il s'agit de proposer de nouvelles fonctionnalités (au moins 3 différents), que ce soit dans le système de jeu ou l'utilisation.

Vous êtes libre de faire les propositions qui vous intéressent. Si vous n'avez pas d'idées, voici quelques propositions :

- Système de sauvegarde, simple ou multiples.
- Système de meilleur résultats sauvegardé.
- Javadoc détaillée.
- De nouveaux types de monstres (insensibles aux armes, se déplaçant après avoir été attaqué, libérant autre chose que de l'or,...)
- De nouvelles armes (soignant le héros, se régénérant, etc...)
- De nouveaux types d'éléments (une pluie de flèches sur tous les ennemis, un bouclier bloquant tout dégâts une fois,...)
- Un système de pouvoirs à stocker, et à réutiliser quand le joueur le décide

2 Évaluation

Vous serez évalués sur la qualité de votre code, le respect de l'approche orienté objet, ainsi que votre rapport, votre présentation et la qualité des réponses aux questions.

Dates importantes :

- **Groupes de 2 :** M'envoyer les binômes par mail (luc.dartois@u-pec.fr) avant le vendredi 21 Avril 18h.
- **Date de rendu 1 :** Avant le Vendredi 19 Mai 18h.
- **Rendu attendu 1 :** Un mail par binôme, contenant une unique archive contenant le jeu codé par vos soins. Le rendu devra compiler sans erreur, et correspondre aux spécificités du jeu. Si votre programme ne fonctionne pas, votre note sera divisée par 2.
- **Date de rendu 2 :** Avant le Vendredi 9 Juin 18h.
- **Rendu attendu 2 :** Un mail par binôme, contenant une unique archive contenant tout (le jeu + vos extensions). Le rendu devra compiler sans erreur, et être accompagné d'un court rapport détaillant vos contributions (présentation, utilisation, technicité, diagrammes).

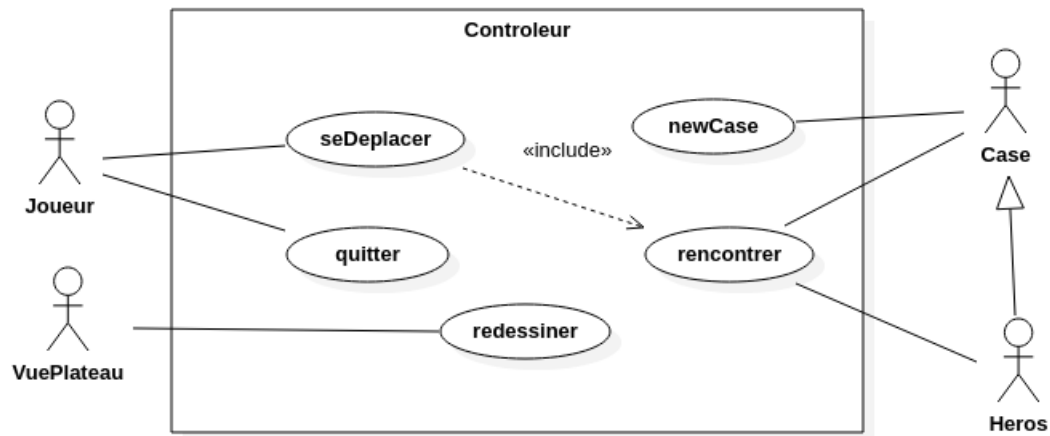


Figure 1: Un diagramme de cas d'usage du Controleur

- Une soutenance aura lieu une semaine après le rendu final. Elle consistera en une courte présentation puis une séance de questions réponses (10min+15min). La présentation pourra s'appuyer sur votre rapport.

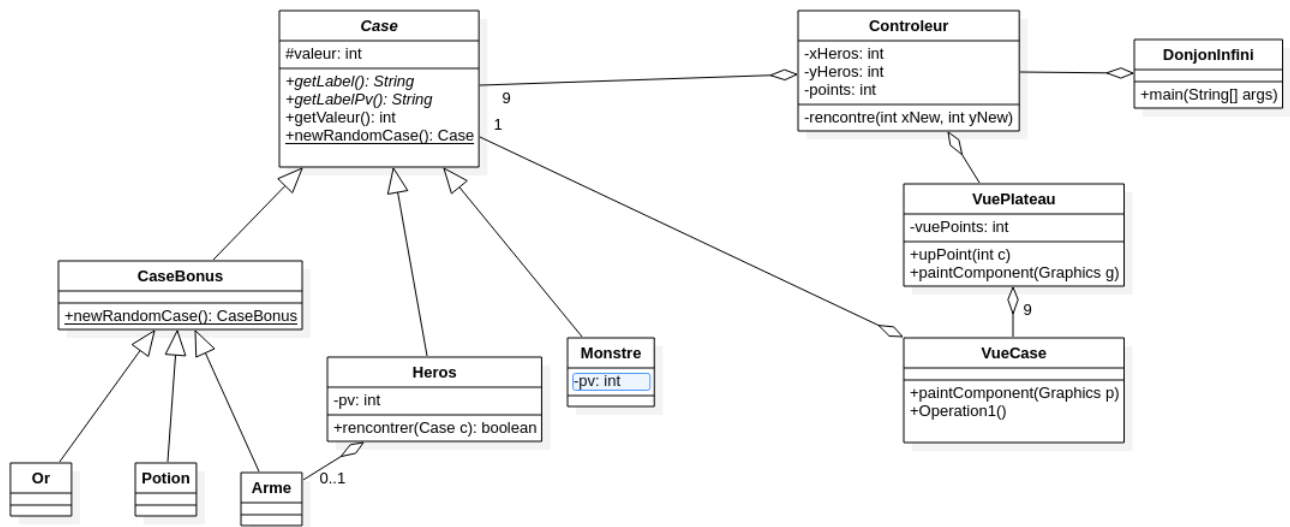


Figure 2: Le diagramme de classes du projet.

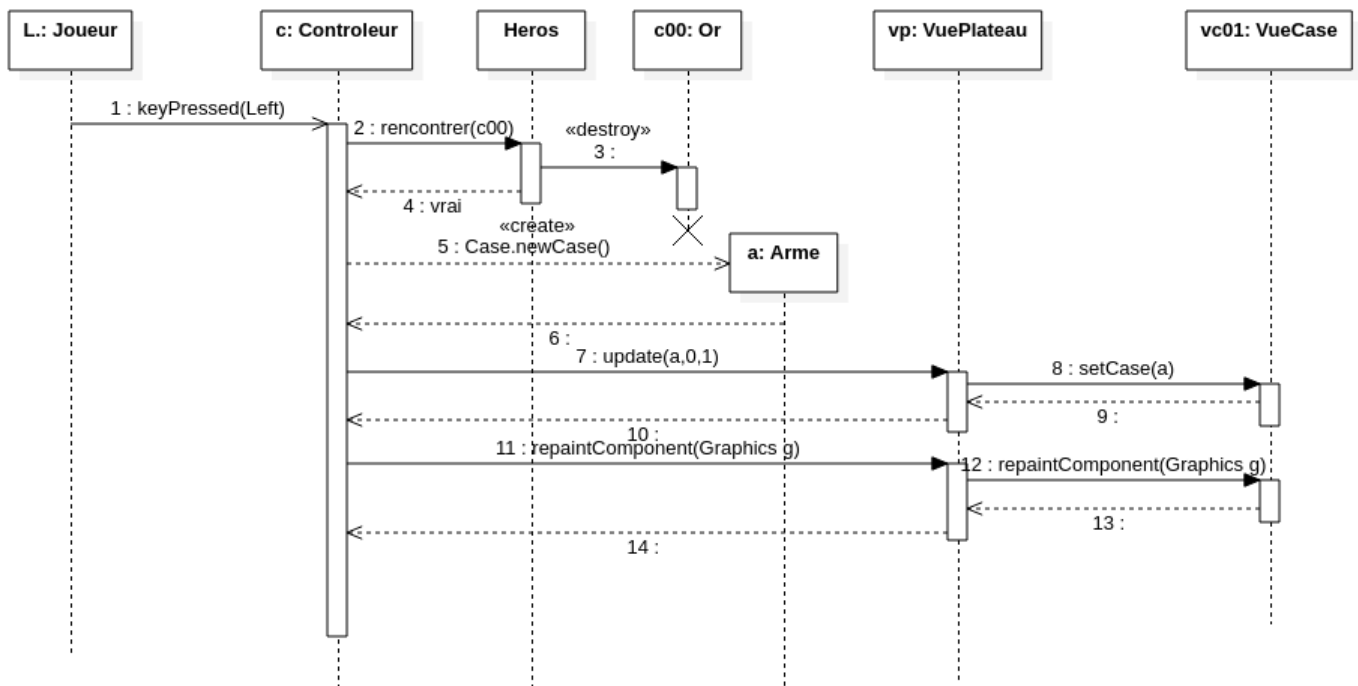


Figure 3: Un diagramme séquence illustrant ce qu'il se passe quand le joueur déplace le héros à gauche, ramassant un sac d'or. Une arme est créée à son ancienne position.