

Zeldiablo

Antonin ROUSSEAU - Célian BAUER - Tom CLAUDEL -
Tung Duong PHAM - Paul LOISIL



Introduction

Présentation globale du jeu et de ses fonctionnalités

Les différentes fonctionnalités

Labyrinthe

Labyrinthe généré
aléatoirement

Attaque

Le joueur peut attaquer les
monstres et inversement

Deplacements

Deplacements fluides et libres
entre les cases

Camera

La caméra vibre à certains
moments et suis le joueur

Versions

Présentation des fonctionnalités par version

Version 1

- **Plusieurs cases:** (Paul)

Créer plusieurs cases, essentiellement une case traversable et une non traversable pour constituer un labyrinthe

- **Un Labyrinthe:** (Tom et Tung Duong)

Assembler plusieurs cases pour réaliser un labyrinthe de taille personnalisable, avec une entrée et une sortie

- **Un Aventurier:** (Antonin)

Créer un aventurier avec un nom et une position dans le labyrinthe

- **Un Jeu:** (Célian)

Créer un jeu permettant de stocker le labyrinthe et le joueur de la partie

Version 2

– Implémenter le moteur de jeu: (Tout le monde)

Implémenter le moteur de jeu fourni à notre structure de jeu déjà existante.

Version 3

- **Gestion du système de positions:** (Paul)
Définir les tailles des cases, le système de coordonnées du jeu, ...
- **Déplacement du joueur:** (Antonin)
Pouvoir faire bouger le joueur dans le labyrinthe en fonction des touches pressées en prenant en compte les collisions avec les cases
- **Points de vie des entités:** (Tung Duong)
Système de points de vie des entités, perdre des points de vie, en gagner, mourir, soigner et attaquer d'autres entités
- **Caméra:** (Tom)
Créer une caméra qui suit sa cible et peut changer de cible à suivre
- **Monstres:** (Célian)
Création de monstres dans le labyrinthe

Version 4

– Ajout des premières textures : (Paul)

Les textures s'affichent correctement, il y a plusieurs types de textures

– Déplacement du joueur : (Antonin)

Pouvoir faire bouger le joueur dans le labyrinthe en fonction des touches pressées en prenant en compte les collisions avec les cases

– Les entités peuvent être armées : (Tung Duong)

Une entité peut en attaquer une autre, en soigner une autre, possèdent une arme

– Création et Labyrinthe : (Tom)

Travail sur un algorithme de création aléatoire de labyrinthe (basé sur l'algorithme de Prism).

– Monstres : (Célian)

Création de monstres avec des points de vie

Version 5

- **Gestion de la création du Labyrinthe :** (Paul)
Génération aléatoire de labyrinthe avec une sortie accessible
- **Interface graphique :** (Antonin)
Ajout d'une interface graphique en ajoutant une barre de vie au joueur et aux monstres
- **Initialisation des projectiles :** (Tung Duong)
Création de projectiles
- **Système d'attaque :** (Tom)
Ajout d'un système d'attaque pour le joueur
- **Design des entités :** (Célian)
Création des différents sprites pour les entités du jeu
- **Déplacement des monstres :** (Célian)
Déplacement des monstres

Version 6

– Réalisation de plusieurs niveaux de labyrinthe : (Paul)

Plusieurs étages de labyrinthe sont créés, des escaliers permettent de monter à l'étage suivant, fin au 3ème étage avec une amulette.

– Animation des personnages avec leur direction: (Antonin)

Lorsque le personnage change de direction, son sprite est modifié

– Initialisation boss d'étage : (Tung Duong)

Le boss a plus de points de vie, est plus gros. Début de la méthode de déplacement

– Test du système d'attaque et optimisation : (Tom)

Réalisation des tests sur l'attaque et changement de la méthode par rapport à ceux-ci

– Déplacement des monstres : (Célian)

Travail sur l'algorithme de déplacement

Version 7

- **Création d'interfaces :** (Paul)

Création d'un écran de début de jeu et un écran game over

- **Gestion des cases :** (Antonin)

Les cases réagissent mieux au joueur (à l'aide de piège pas exemple)

- **Test sur le boss :** (Tung Duong)

Le boss arrive-t-il à se déplacer, a-t-il le bon nombre de pv.

- **Système d'attaque des monstres :** (Tom)

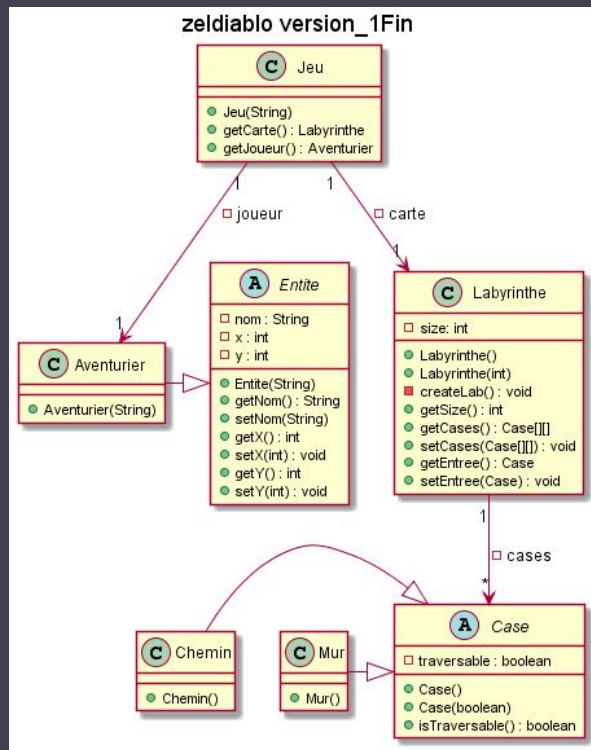
Les monstres peuvent désormais attaquer le joueur et lui faire perdre des points de vie

- **Monstres:** (Célian)

Gestion de déplacement des monstres (deuxième partie)

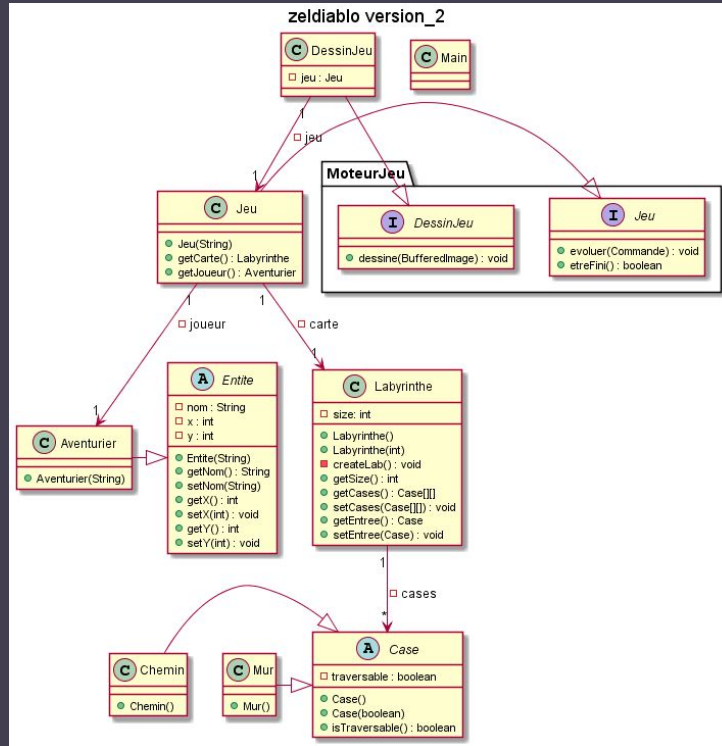
Evolution du diagramme de classe

Version 1



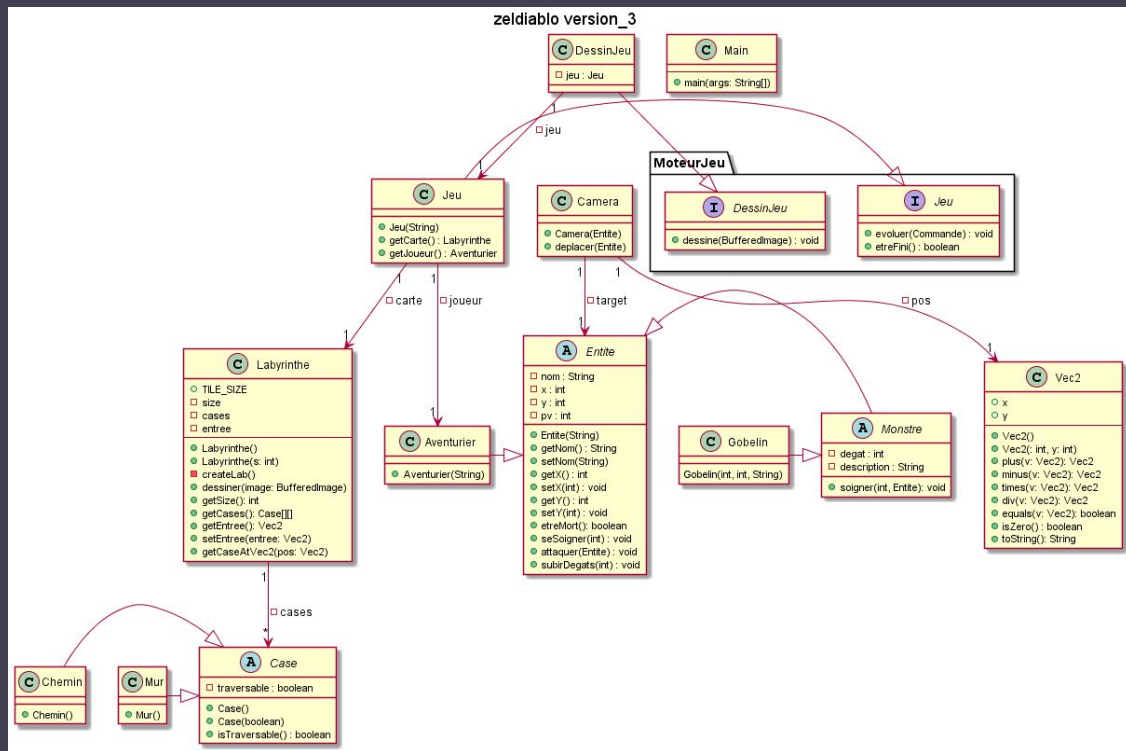
Evolution du diagramme de classe

Version 2



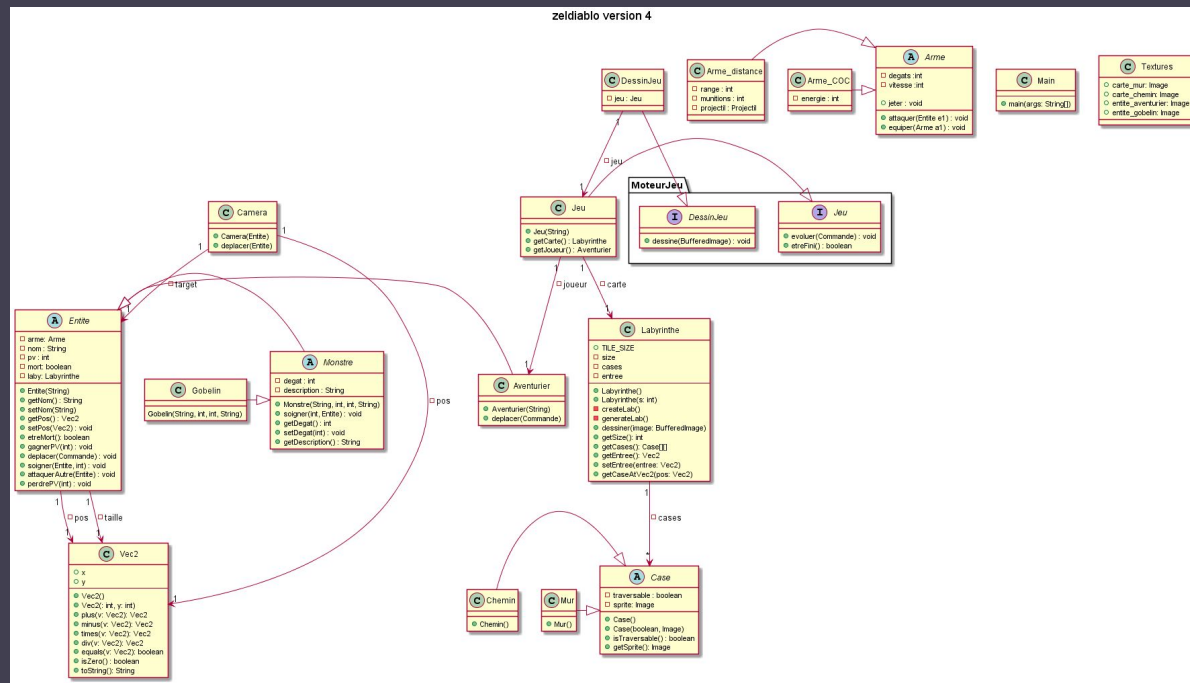
Evolution du diagramme de classe

Version 3



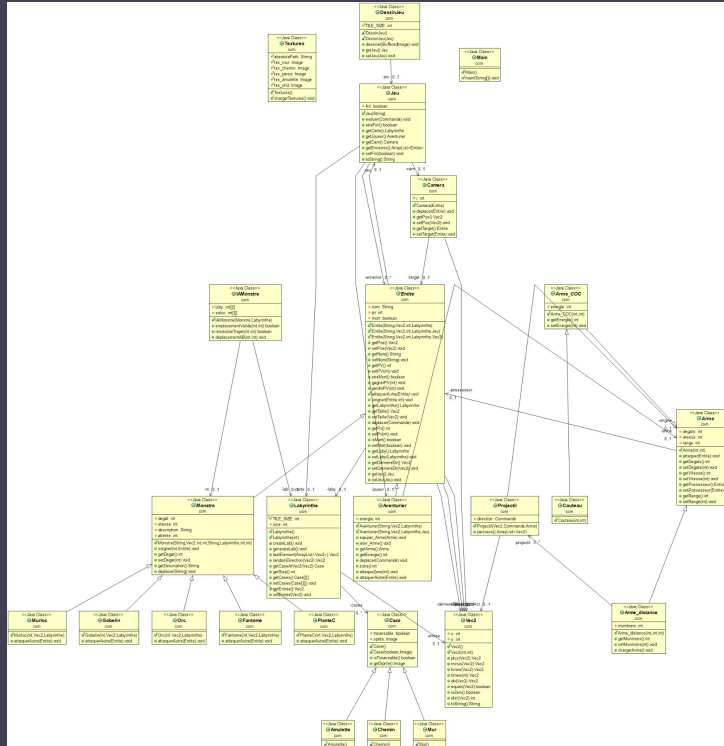
Evolution du diagramme de classe

Version 4



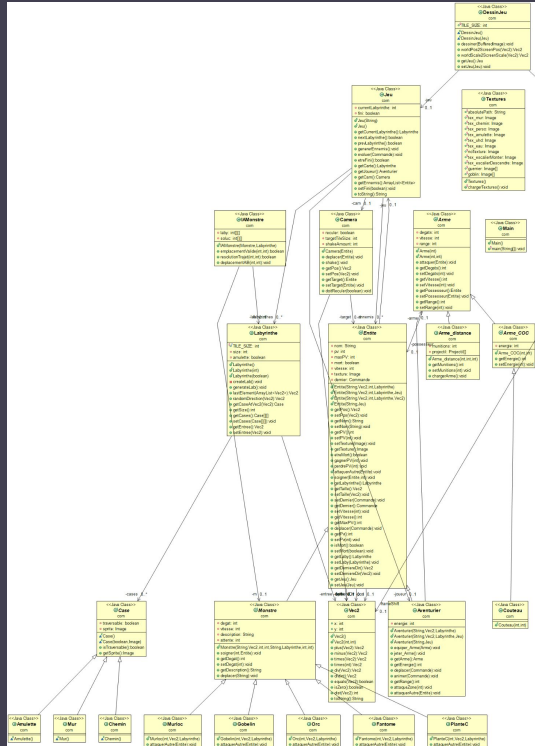
Evolution du diagramme de classe

Version 5



Evolution du diagramme de classe

Version 6



Evolution du diagramme de classe

Version 7

Iteration en cours ...

Points intéressants de conception

- Factorisation du code:
 - Plus facile de créer de nouvelles cases
 - Moins de code dupliqué
 - Modifications plus faciles
- Système d'entités:
 - Tous les monstres sont des entités (déplacement commun entre eux)
 - Les monstres peuvent attaquer le joueur mais aussi les monstres
 - Pareil pour soigner une entité

```
126      126      setVitesse(2);
127      -      if(proch instanceof Amulette){
128      -      //temporaire
129      -      System.out.println("Le jeu est fini");
130      -      this.getJeu().setFin(true);
131      -      }else if(proch instanceof Eau) {
132      -      setVitesse(1);
133      -      } else if (proch instanceof EscalierMonter) {
134      -      if (this.getJeu().nextLabyrinthe()) {
135      -      this.setPos(this.getJeu().getCurrentLabyrinthe().getEntree().times(Labyrinthe.TILE_SIZE));
136      -      this.getJeu().genererEnnemis();
137      -      }
138      -      }
127 +      proch.action(getJeu());
139      128      // System.out.println("ca passe");
```

Conclusion

Difficultés rencontrés lors de ce projet:

- Les classes nécessitant des algorithmes plutôt complexes (Déplacement des monstres, génération aléatoire du labyrinthe)
- Communication plus difficile par moment

Avis général:

Dans l'ensemble le projet s'est plutôt bien déroulé malgré quelques difficultés. L'ambiance au sein du groupe était très conviviale, il était donc très agréable de travailler ensemble sur ce projet Zeldiablo.

Merci

Avez-vous des questions ?