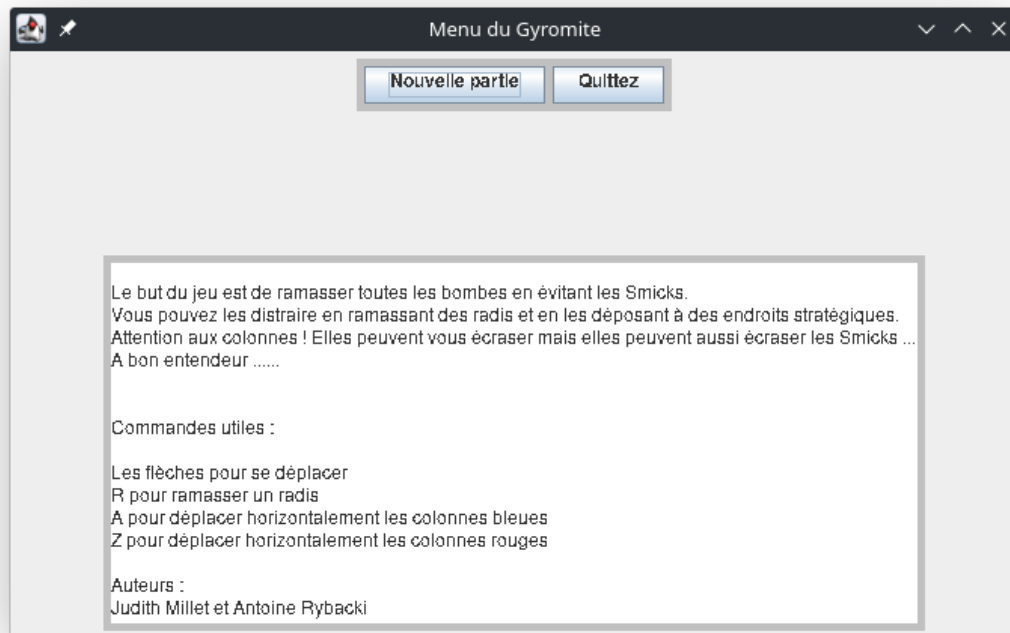


Projet POO

Gyromite - 2020/2021



Auteurs :
MILLET Judith
RYBACKI Antoine

Encadrants :
CHAPUT Remy
ARMETTA Frederic

Fonctionnalités et extensions implémentées :

- Modification de l'environnement et ajout des classes nécessaires dans le modèle
 - Ajout des cordes (5%)
 - Ajout des colonnes (10%)
 - Ajout des bombes (3%)
 - Ajout des radis (3%)
- Ajout de Smicks et gestion de l'IA (20%)
- Capacité de monter/descendre aux cordes lorsque le Prof. est sur une corde (10%)
- Gestion mouvement des colonnes (10%)
- Gestion des collisions (5%)
- Ramassage des bombes (5%)
- Ramassage et dépôt des radis (7%)
- Distraction Smicks grâce aux Radis (5%)
- Incrémentation d'un score (5%)
- Fin de niveau gagnante ou perdante (5%)
- Ajout d'un menu (5%)
- redimension de la fenêtre (2%)

Documentation UML

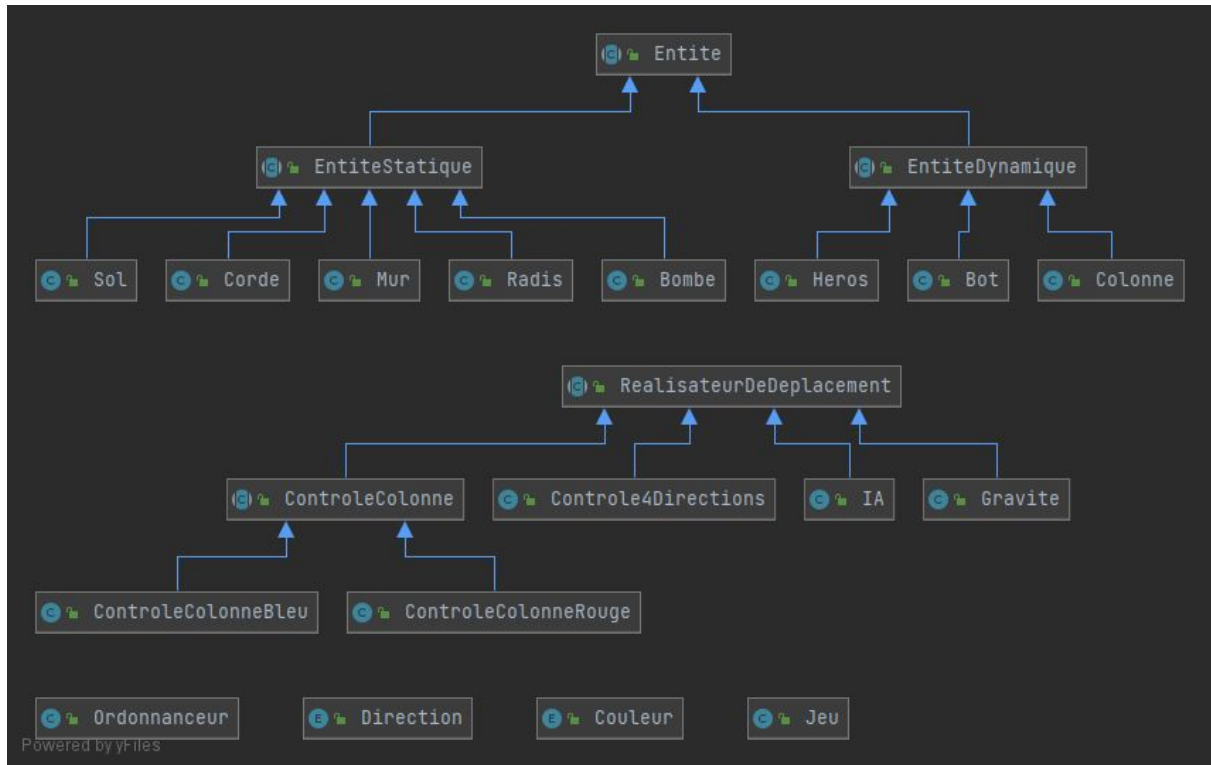


Figure 1 : Diagramme de Classes simplifié (seulement les héritages)

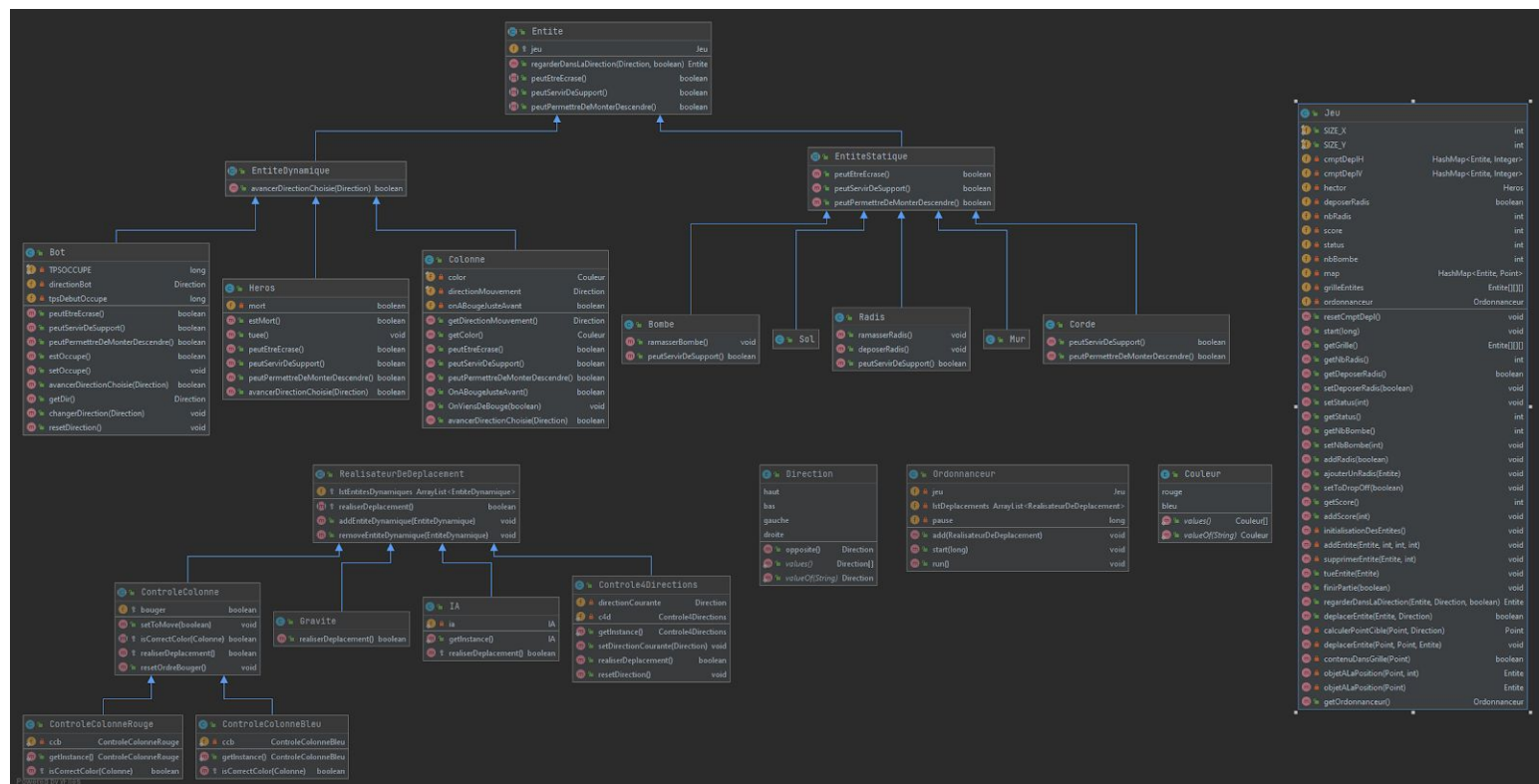


Figure 2 : Diagramme de Classes simplifié avec attributs et méthodes (seulement les héritages)



Figure 3 : Diagramme de Classes avec toutes les dépendances

Choix de conception

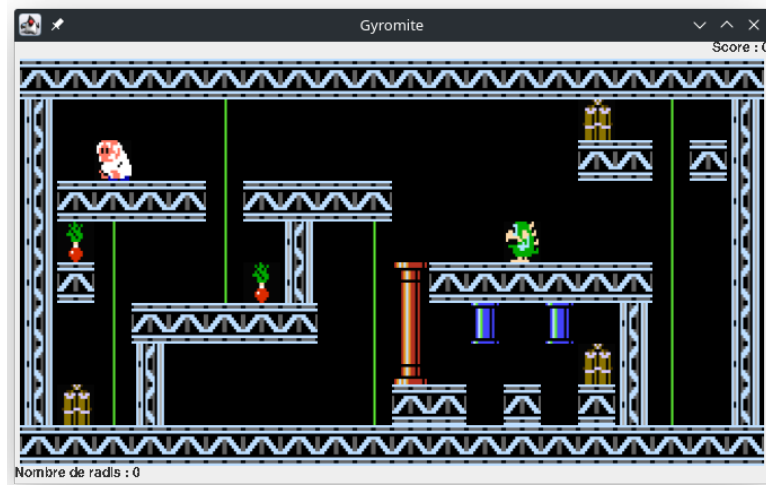


Figure 4 : Partie en cours

Implémentation des objets de l'environnement :

Nous avons décidé d'ajouter une 3ème dimension à la grille des entités du jeu (`grilleEntites[x][y][dim]` dans la classe `Jeu`). Cela a permis de dissocier les éléments statiques de l'environnement (dimension 0) et les éléments dynamiques tels que le Héros, les Smicks (dimension 1). Cas particulier : les colonnes sont dans les deux dimensions.

Capacité de monter/descendre des cordes :

Ainsi, nous avons pu implémenter les cordes en regardant les entités à la dimension 0 et 1 simultanément. De ce fait, si une entité dynamique (donc à la dimension 1) se trouve sur une corde, elle pourra monter ou descendre.

Gestion de l'IA des Smicks :

Cette gestion se fait dans la classe `IA`, héritant de la classe `RealisateurDeDeplacement`. Elle sera un singleton dans le jeu.

Un déplacement de Bot se fait de la sorte : nous avons affecté une direction aléatoire horizontale pour chaque Smick à sa construction. De là, il tente de se déplacer dans cette direction. Mais si cela ne fonctionne pas, il essaiera de se déplacer dans les directions aléatoires verticales (l'une puis son opposé) puis aléatoires horizontales de la même manière. Si le déplacement n'est toujours pas possible, alors aucun déplacement n'est possible et sa direction devient "null".

Gestion des mouvements des colonnes :

Pour gérer les colonnes bleues et rouges, on utilise une classe abstraite `ControleColonne` héritant de la classe `RealisateurDeDeplacement`, elle implémente la gestion du mouvement en lui-même.

Pour différencier les colonnes rouges des colonnes bleues, on a créé les classes `ControleColonneBleu` et `ControleColonneRouge`, implémentant une fonction abstraite `isCorrectColor()` qui permet de différencier la couleur des colonnes. De plus on se sert d'une fonction statique `getInstance` pour que ces classes de Contrôle soient des singletons. Avec ce modèle, on pourrait potentiellement rajouter une autre couleur de colonne sans difficulté.



Figure 5 : Partie en cours avec déplacement de la colonne rouge

Ramasser et déposer des radis par le Heros :

Le ramassage et le dépôt de radis est géré dans la fonction `avancerDirectionChoisie(Direction d)` de la classe `Heros`. Ainsi, on regarde si le Heros peut avancer et s'il se retrouve sur un radis, alors il le ramassera. Cela incrémentera `nbRadis` et le radis ramassé sera supprimé. Pour le dépôt, le joueur doit appuyer sur la touche R et si le Heros n'est pas sur une corde, on dépose le radis à la position du Heros. Le placer sur une corde supprimerait celle-ci si le radis est ramassé de nouveau ou consommé par le Bot car il serait placé à la dimension 0, où se trouve la corde.

La distraction des Smicks :

La collision entre un Smick et un Radis se fait dans la fonction `avancerDirectionChoisie(Direction d)` de la classe `Bot`. En effet, le Bot se retrouve distrait s'il se retrouve à la position d'un radis. Il sera immobilisé 3 secondes et pourra avancer à la fin de ce temps. Le radis consommé sera alors supprimé.