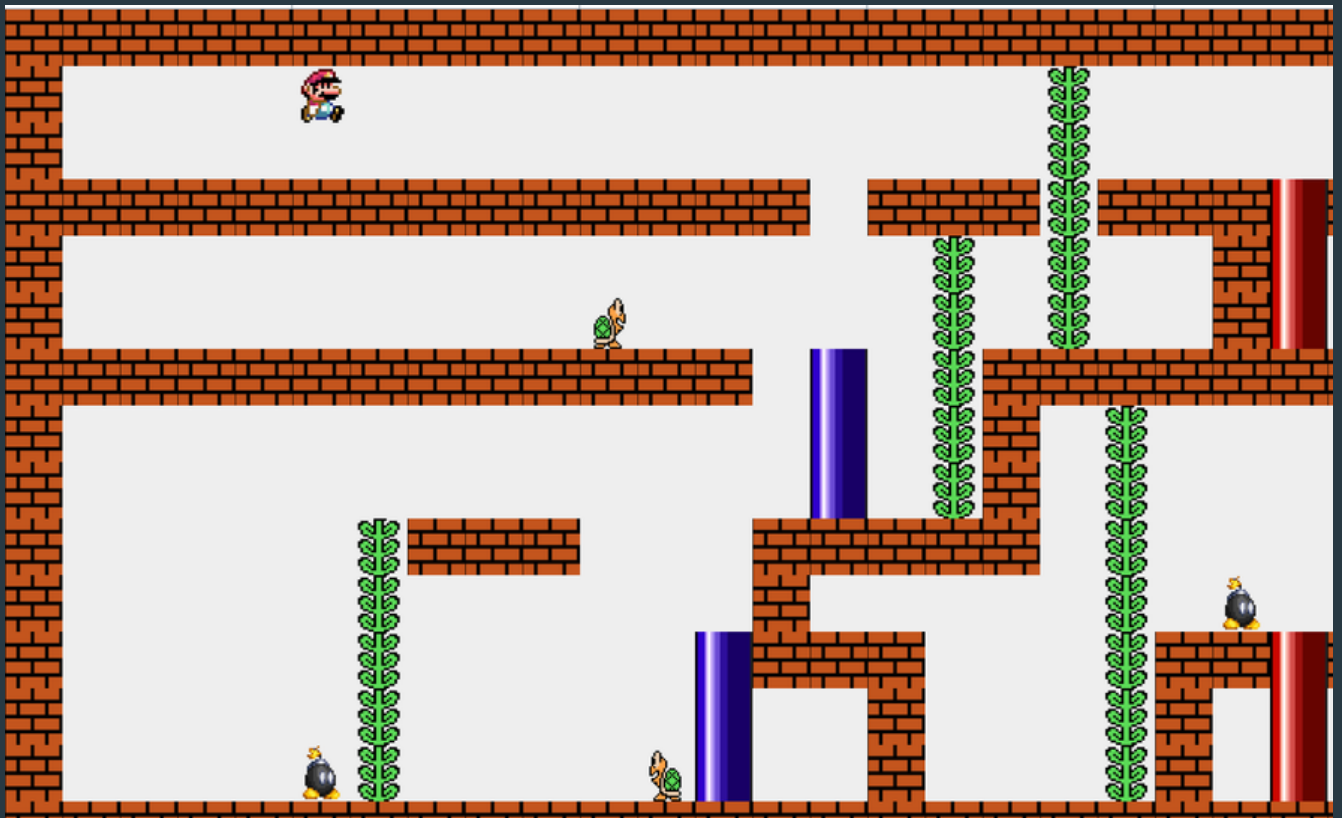


31/05/2021



GYROMARIO

Réalisé en Java avec Swing



Par Baptiste AUBERT & Rémi MARTINEZ

Fonctionnalités

- Gestion de la gravité

A chaque frame, l'ordonnanceur regarde pour toutes les entités dynamiques la case en dessous d'elle. S'il s'agit d'une case vide, l'entité tombe d'un bloc et l'opération est répétée.

- Gestion des collisions

Avant chaque déplacement, on regarde dans la direction visée si la case d'après peut être traversée.

- Monter aux cordes (lianes)

Grâce à un système de "couches" l'entité "Heros" peut superposer une entité "Corde"

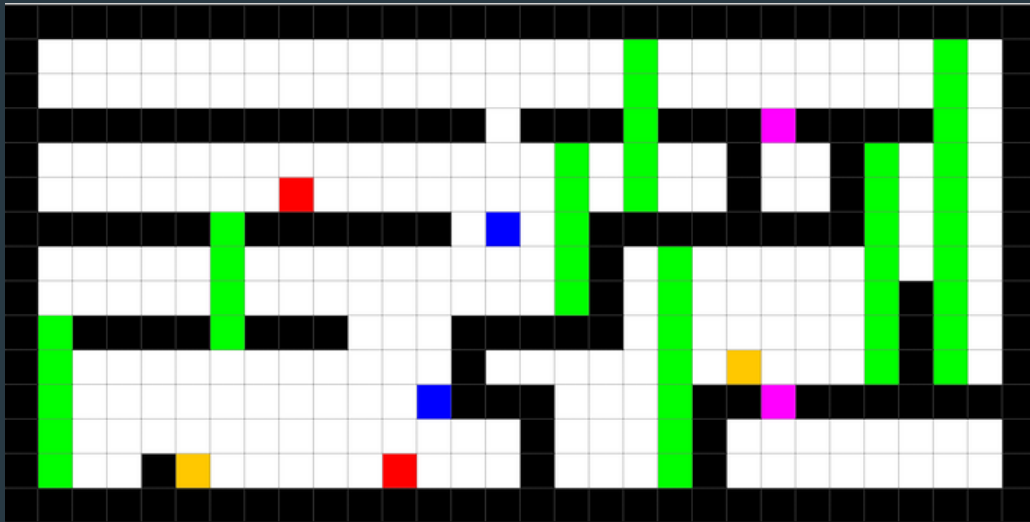
- Ramasser les bombes et les pièces
- Déplacer les colonnes et écraser les entités
 - Touche B pour les colonnes bleues
 - Touche R pour les colonnes rouges
- Système de points
 - +50 pour une pièce
 - +100 pour une bombe
 - +20 pour tuer un smicks
 - -20 pour une mort
- Mort (collision avec un ennemi ou écrasé par une colonne)
- Déplacer le héro lorsqu'il est sur une colonne



Extensions

- **Générateur de niveau**

Le code lit des images png avec des pixels de différente couleur interprétées pour instancier l'entité en question



Chaque couleur correspond à un élément différent :

- Rouge : Smicks
- Bleu : Colonne bleu
- Magenta: Colonne rouge
- Noir : Mur
- Blanc : Vide
- Orange : Bombes
- Jaune : pièce
- Rose : cœurs
- **Plusieurs niveaux:**



Plusieurs niveaux sont possibles d'être ajouté rapidement grâce à l'éditeur de niveau, et ainsi jouer plusieurs niveaux à la suite.

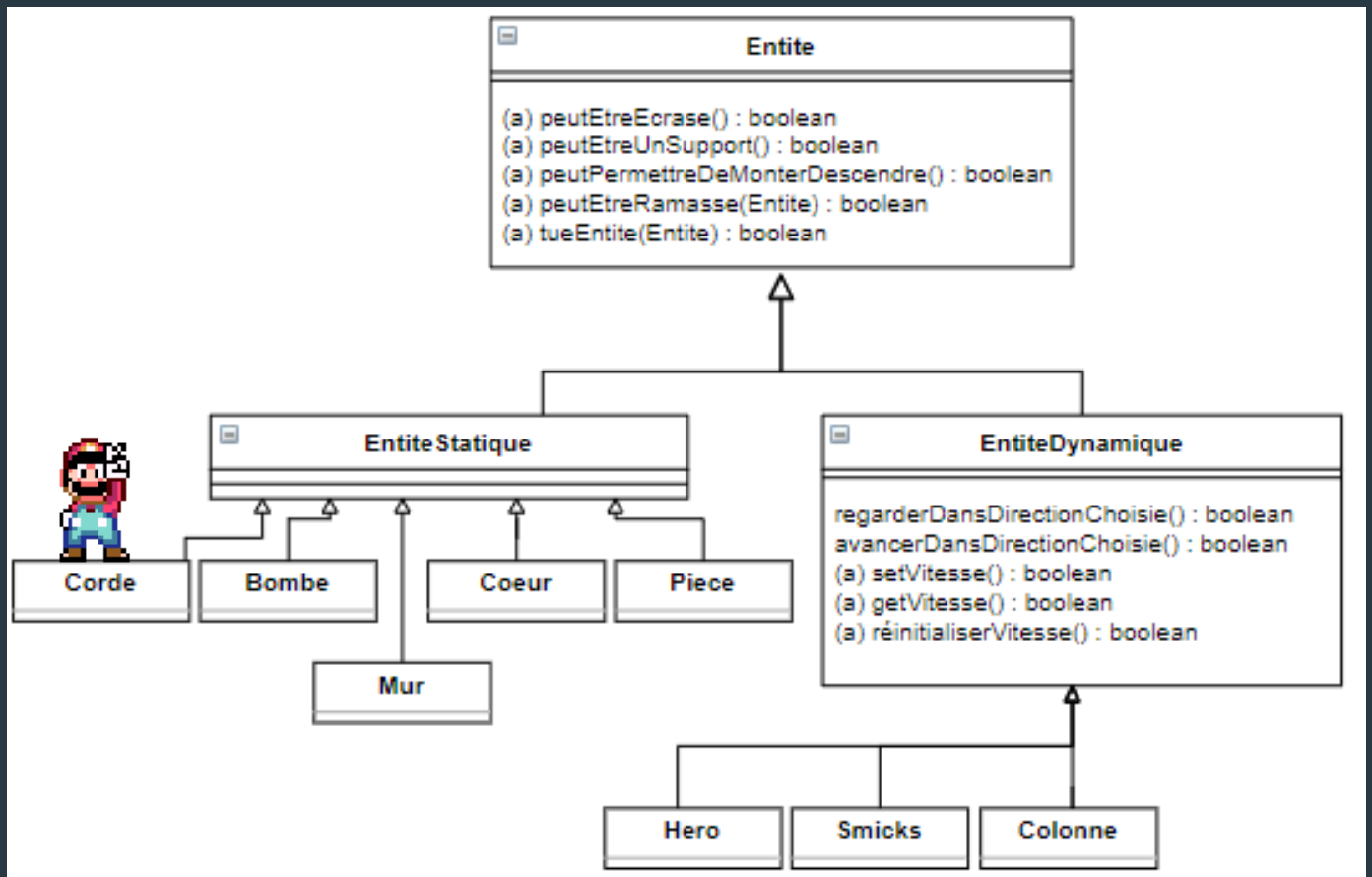
- **Meilleurs scores**

Les meilleurs scores sont enregistrées et affichés à l'écran en fin de partie.



Documentation UML

Voici un diagramme de Classe représentant notre implémentation des entités :



Choix de conception

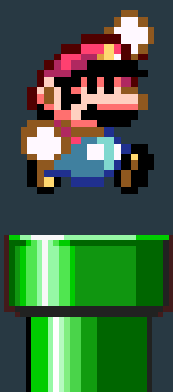
Architecture MVC Strict

L'application est réalisée en suivant le design Modèle Vue Contrôleur. Suite à notre cours et étant donné que le jeu reste une petite application nous avons choisi de réunir la vue et le contrôleur dans un seul fichier.

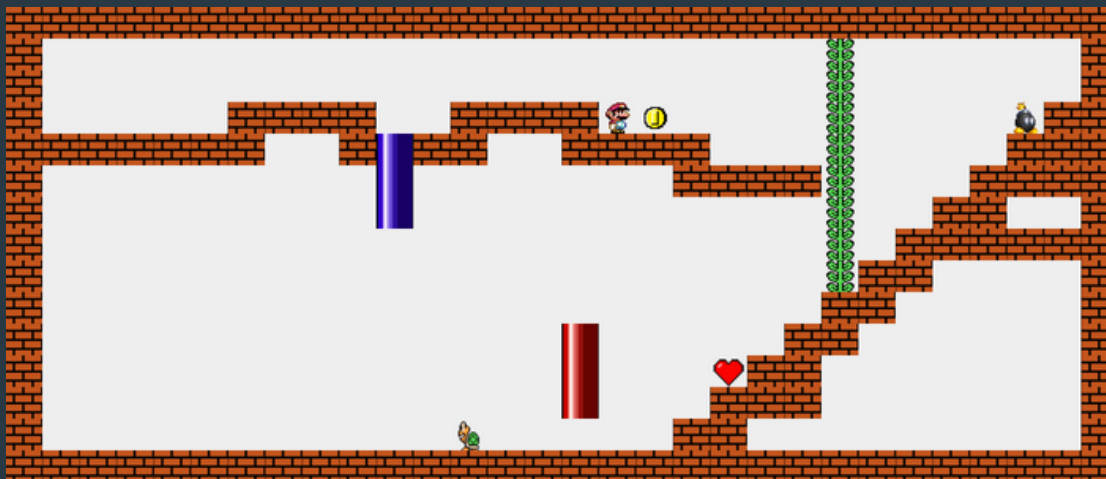
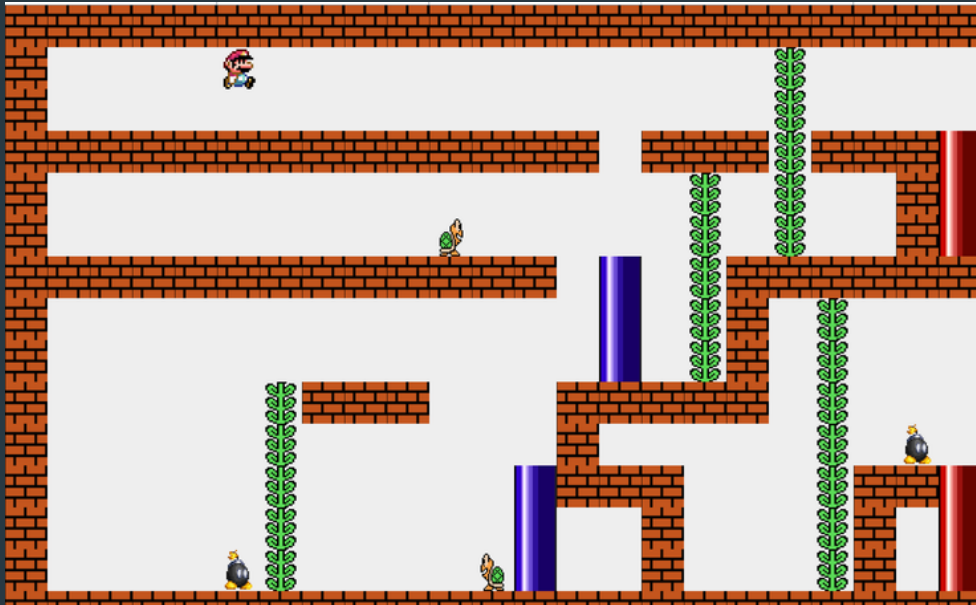
La partie modèle est séparée en deux packages (plateau et déplacement) qui comportent respectivement les différentes entités de jeu et les classes qui permettent d'effectuer les différents déplacements.

Comme vu sur le diagramme UML plus haut, les entités désignent tous les éléments du jeu avec lesquelles on peut interagir. Il y a les entités statiques et les entités dynamiques. Les entités statiques comme les murs, les bonus à ramasser n'ont pas vocation à se déplacer. Les entités dynamiques comme les héros ou les IA peuvent se déplacer entre les cases et interagir avec le héros : une colonne peut écraser le héros, un Smicks peut faire perdre une vie au héros.

Pour gérer dynamiquement la position des entités et leur affichage, nous avons utilisé deux "grilles". La première permet de gérer la position de chaque élément sur une grille virtuelle pour savoir à quelle position x et y se situent les objets sur la scène. On travaille avec une grille manipulant des cases et non des pixels (position **discrète** et non continue). La deuxième grille permet d'afficher les éléments graphiques et contient une couche supplémentaire qui permet de superposer les sprites. Comme par exemple le héros lorsqu'il est sur une liane: la couche 0 correspond à la liane et la couche 1 correspond au héros qui superpose le sprite de la liane.



Captures d'écran



```
private JPanel placerLesComposantsGraphiques() {
    JPanel game = new JPanel(new BorderLayout());
    setTitle("Gyromite");
    setSize(resX, resY);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    grilleJLabels = new JPanel(new GridLayout(sizeY, sizeX));
    tabJLabel = new JLabel[sizeX][sizeY];
    for (int y = 0; y < sizeY; y++) {
        for (int x = 0; x < sizeX; x++) {
            JLabel jlab = new JLabel();
            tabJLabel[x][y] = jlab;
            grilleJLabels.add(jlab);
        }
    }

    String data[][] = this.jeu.getScore();
    String columns[] = {"", "", "", "", "", ""};
    score = new JTable(data, columns);
    score.setFocusable(false);
    score.setRowHeight(resY/10);
    game.add(score, BorderLayout.NORTH);
    game.add(grilleJLabels, BorderLayout.CENTER);
    return game;
}
```

Merci pour votre lecture !



GYROMARIO

Par Baptiste AUBERT & Rémi MARTINEZ.