## ligne horizontale



Flow Game

Projet LIFAP7

Année 2021-2022

**─**

Charles ENG P1403762

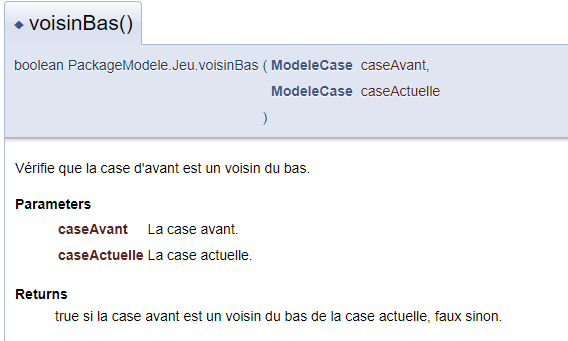
Kévin TANG P1501263

# Liste des fonctionnalités

## Détection des cases voisines

Pour détecter les cases voisines, nous avons simplement implémenté des fonctions comparant les coordonnées de la case actuelle avec celle d’avant afin de voir si c’est bien un voisin, et si oui, quel voisin (haut, bas, gauche, droite).

Exemple :



## Création de chemins

Les chemins sont des listes de cases modèles. Notre jeu est composé d’un tableau composé de ces chemins. Afin que le chemin soit validé, il y a plusieurs conditions :

* La case de départ est un symbole
* Toutes les cases sont voisines l’une de l’autre (pas de saut en sortant de la fenêtre par exemple)
* La dernière case est la même case symbole que celle du début mais diffère par ses coordonnées.

Une fois validé, le chemin est ajouté dans le jeu et l’affichage se met à jour.

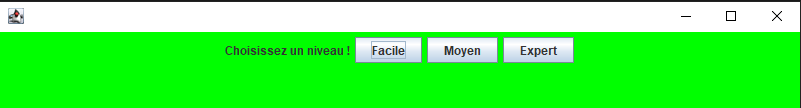
Pour supprimer les chemins, il suffit de cliquer sur le chemin ou les symboles du chemin.

## Choix du niveau

La dernière fonctionnalité que nous avons ajouté est un petit menu lancé au début de jeu qui permet de choisir entre 3 niveaux de difficultés :

* Facile : avec une grille 3x3 et 2 symboles différents uniquement
* Moyen : avec une grille 5x5 et 5 symboles différents
* Expert: avec une grille 9x9 et 9 symboles différents

Pour choisir le niveau, il suffit simplement de cliquer sur le bouton correspondant. Les niveaux sont hard codés et ne peuvent pas être modifiés.



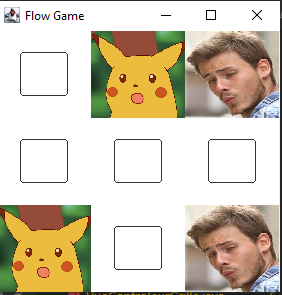
## Amélioration du jeu

Afin d’améliorer le jeu, il y aurait plusieurs fonctionnalités possibles :

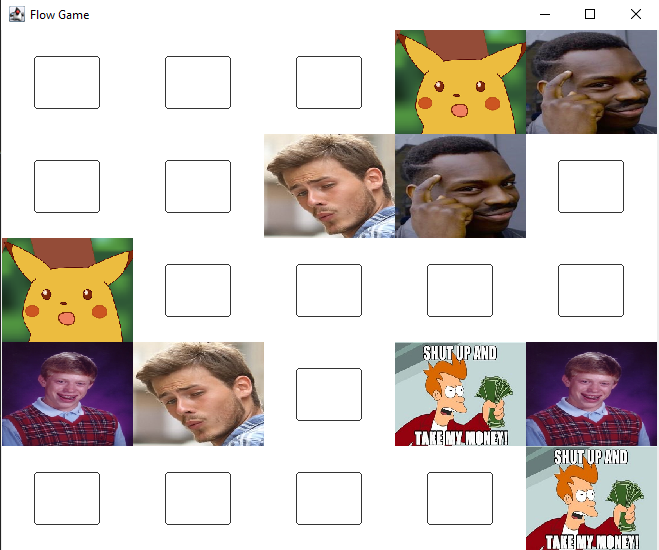
* Rafraîchir l’affichage pendant que l’utilisateur crée le chemin : c’est-à-dire qu’actuellement, le joueur ne voit pas le chemin pendant qu’il est en train de créer un chemin, ça permettrait d’améliorer grandement la jouabilité si le joueur avait un visuel du chemin pendant sa création. Il faudrait pour cela mettre à jour l’affichage à chaque action du joueur.
* Des niveaux de difficultés non hard codés : cela permettrait de grandement augmenter la durée de vie du jeu.
* Charger des fichiers pour ouvrir un niveau de jeu.
* Afficher un score et enregistrer les scores des gagnants dans un fichier pour qu’ils puissent se comparer.
* Mettre des “vies” qui permettrait au joueur d’avoir un enjeu et de ne pas avoir de tentatives infinis

## Screenshot de notre jeu

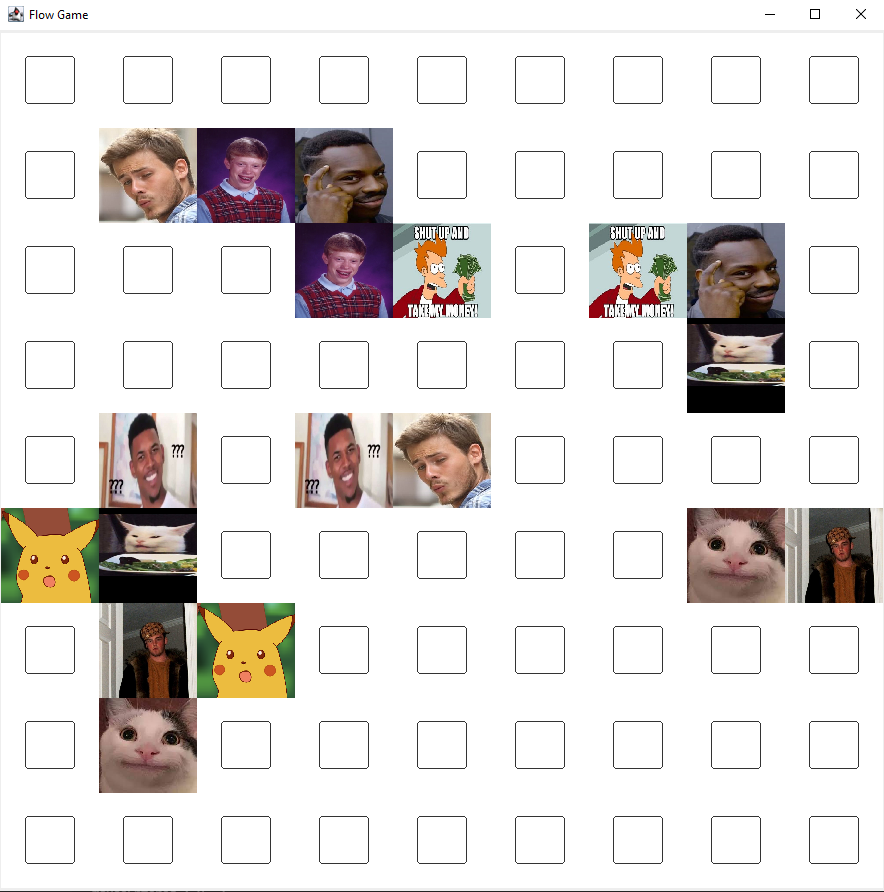
Niveau facile :



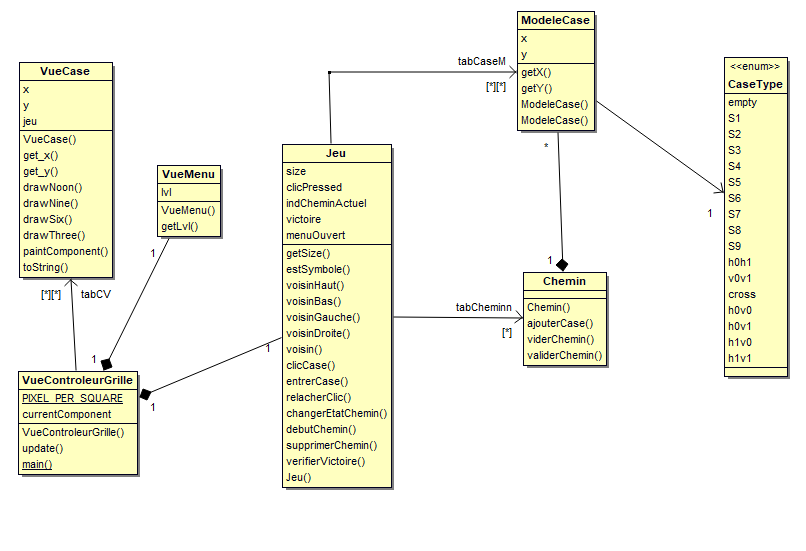
Niveau moyen :



Niveau expert :



# Diagramme UML (diagramme de classes)



# Documentation Doxygen

Nous avons commenté tout notre code afin de pouvoir générer une documentation Doxygen disponible dans doc/html/index.html.

