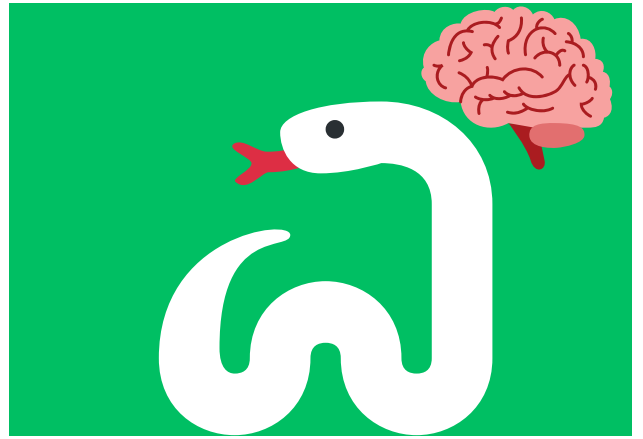


# SNAKE MEMORY

AXEL CANUET - RAYLIB

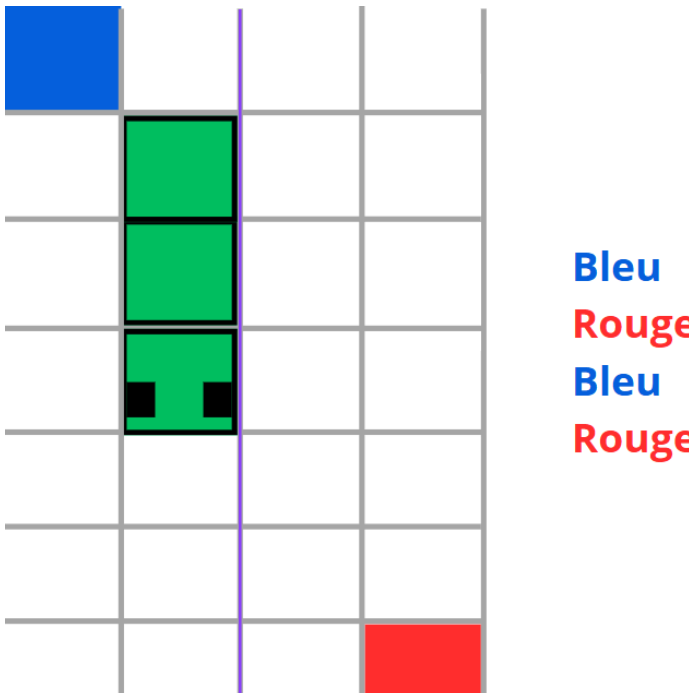


## Gameplay



Le jeu part sur le principe du snake classique, récupérer des pommes pour faire grandir le snake et avoir le plus de point possible.

Dans ma version, le snake devra respecter un ordre de couleur de pommes a récupérer.



- A droite de l'écran l'objectif
- Quand une pomme est récupérée le snake s'allonge
- Si l'ordre n'est pas respecté, -1 PV et le snake diminue de taille
- Si le snake perd 2 PV alors Game Over
- Si le snake récupère toutes les pommes de l'objectif alors c'est gagné



## Niveau

Il y a 3 niveaux :

- LvL 1 : 4 Couleurs 
- LvL 2 : 8 Couleurs 
- LvL 3 : 12 Couleurs 

## Contrôle



Le snake avance tout seul et peut être contrôlé avec :

- Les flèches directionnelles



## Architecture code

- **Menu.cs** → Point d'entrée du jeu : initialise Raylib, gère la boucle principale et les écrans (menu, difficulté, jeu).
- **GameState.cs** → Définit les différents états possibles du jeu (menu, en cours, victoire, game over, etc.).
- **Grid.cs** → Représente et dessine la grille de jeu avec ses cases et lignes.
- **Snake.cs** → Gère le serpent : positions, direction, déplacements, collisions et affichage.
- **Apple.cs** → Représente une pomme avec sa position et sa couleur et la dessine. C'est aussi dedans que l'on énumère toutes les couleurs possibles.
- **SongManager.cs** → Charge et joue les effets sonores/musiques spécifiques au jeu.
- **SnakeGame.cs** → Logique de gameplay, règles, score, PV, objectifs.