

**BTS SIO**  
SLAM LM

# Projet Snake

**Benjamin, Yasser, Ryles**

01

## Présentation du Projet

*Snake, Equipe*

02

## Gestion du projet

*Trello, Gantt, Pert*

03

## Développement

*Problème et solution*

04

## Bilan de compétence

**PROJET**  
SNAKE

# PROJET

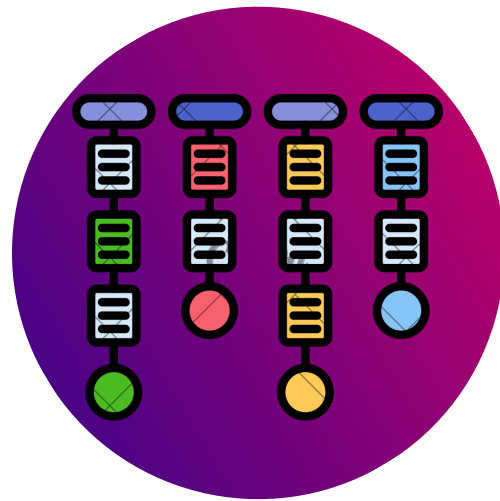
## SNAKE

## Présentation du projet

Le but du projet:

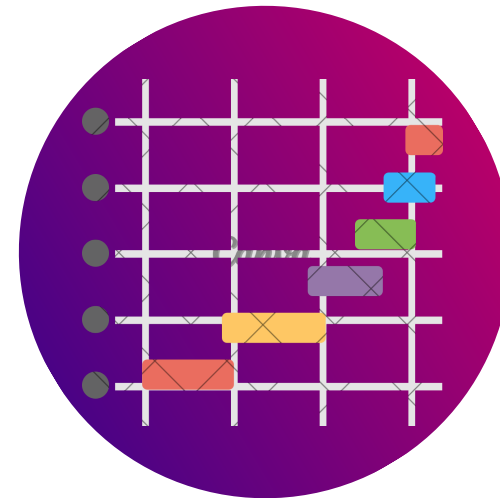
- Génération des pommes par 3 à la fois
- Manger 3 pommes pour agrandir le serpent
- Système de sauvegarde / chargement de partie
- Mode de jeu à 2 vitesses
- Bibliothèque utilisée : Conio





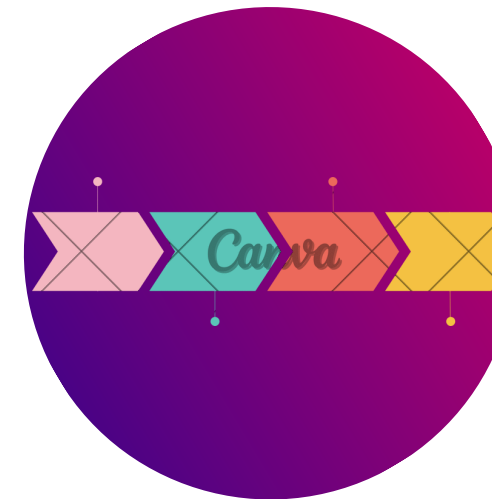
## Trello

Planification des différentes tâches au membre du groupe



## Gantt

Ordonnancement du projet avec les dates limites



## Pert

Analyse du projet avant le commencement du développement du code



## Github / Git

Utilisation de git / GitHub à partir de github.dev  
Push / commit à partir de VS-Code

# PROJET SNAKE

# Trello

The screenshot shows a Trello board for the 'PROJET SNAKE' project. The board is organized into five columns: Réunion, Problème à résoudre / Analyse, A Faire, En cours, and Terminer. Each column contains cards representing tasks or meetings, with assignees, due dates, and status indicators.

- Réunion**: Contains two cards. The first card is assigned to Yasser, BenG, and Ryles, with a due date of 26/09 22H. The second card is assigned to Yasser, BenG, and Ryles, with the title 'Quelle bibliothèque choisir'.
- Problème à résoudre / Analyse**: Contains one card titled 'Uniformiser la vitesse du serpent (trop rapide vers le haut ou bas)'.
- A Faire**: Contains one card titled 'Ajouter une carte'.
- En cours**: Contains one card titled 'Gestion du GitHub'.
- Terminer**: Contains three cards. The first card is assigned to Yasser and BenG, with a due date of 27 sept. - 29 sept. The second card is assigned to BenG and Ryles, with a due date of 21 sept. - 23 sept. The third card is assigned to BenG, with a due date of 26 sept. - 29 sept.

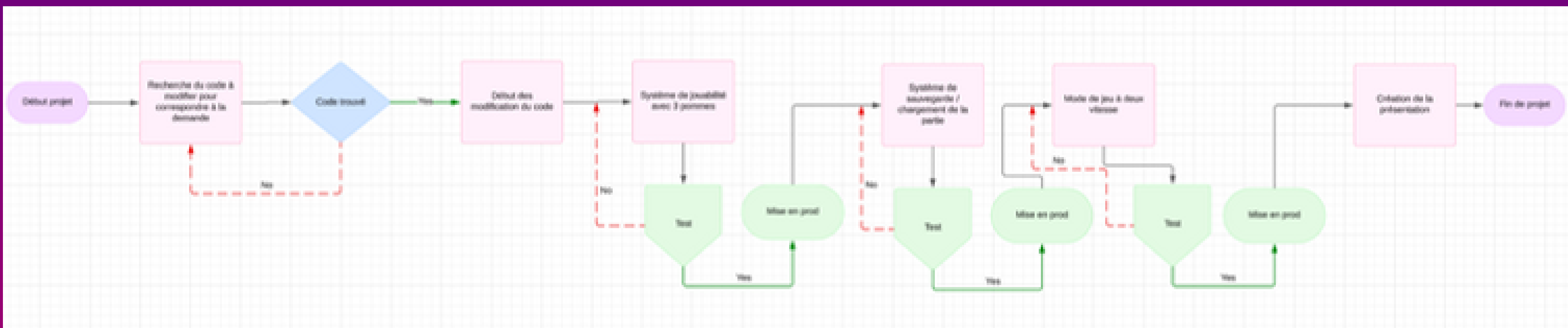
PROJET  
SNAKE

Gantt



# PROJET SNAKE

## Pert



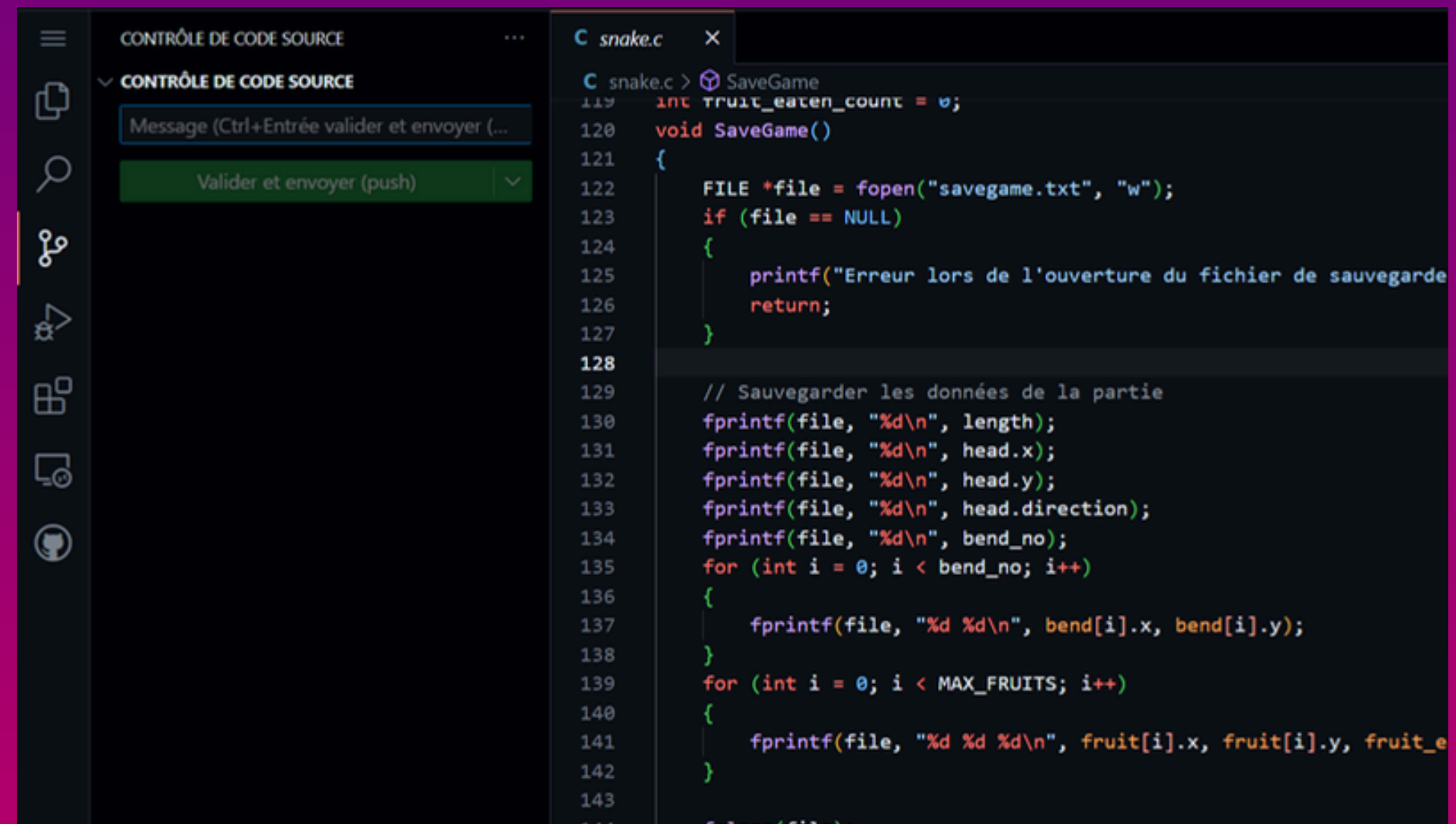
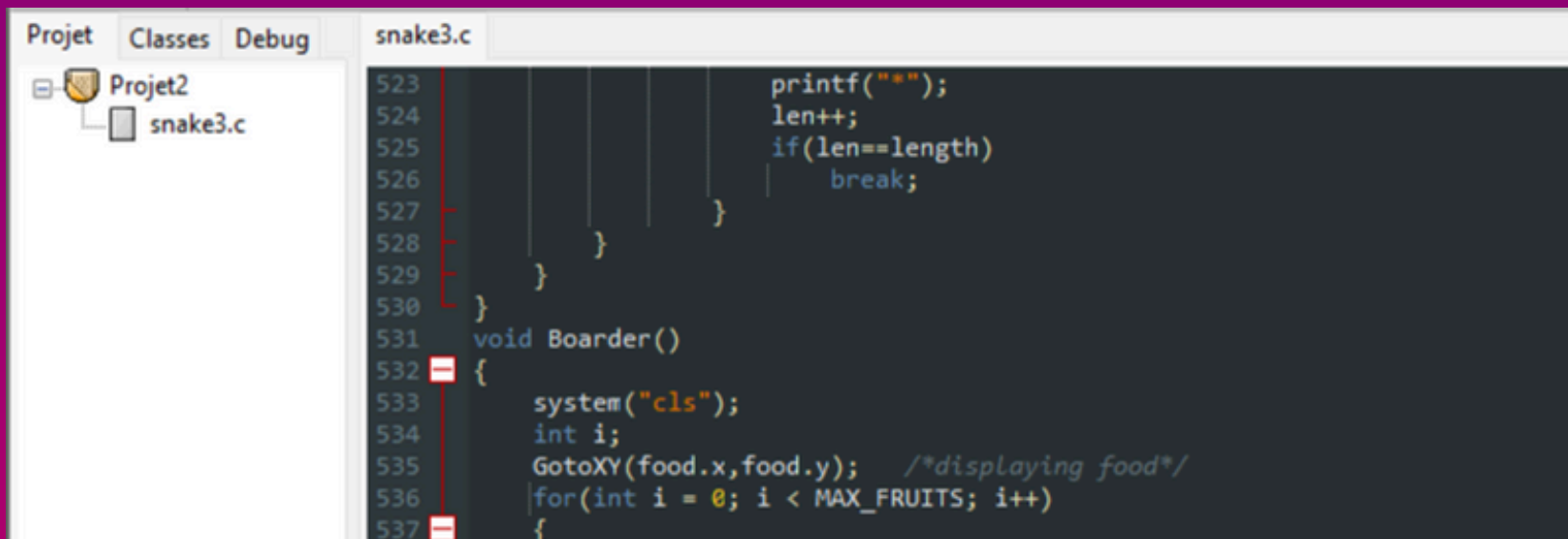
# Developpement du jeu





# PROJET SNAKE

# GitHub / Git



# Fonction Fruit

```
16  * #define MAX_FRUITS 3
17  *
16 18  int length;
17 19  int bend_no;
18 20  int len;
+
+
@@ -288,11 +290,53 @@ void ExitGame()
288 290      }
289 291      }
290 292      }
+
+ coordinate fruit[MAX_FRUITS];
+ int fruit_eaten[MAX_FRUITS];
+ int fruit_gen_count = 0;
+ int fruit_eaten_count = 0;
291 297  void Food()
292 298  {
293  - if(head.x==food.x&&head.y==food.y)
299  + if(fruit_gen_count < MAX_FRUITS) // si moins de 3 fruits générés
300  + {
301  +     for(int i = 0; i < MAX_FRUITS; i++)
302  +     {
303  +         fruit[i].x = rand()%70;
304  +         if(fruit[i].x <= 10)
305  +             fruit[i].x += 11;
306  +         fruit[i].y = rand()%30;
307  +         if(fruit[i].y <= 10)
308  +             fruit[i].y += 11;
309  +     }
310  +     fruit_gen_count = MAX_FRUITS; // incrémenter le compteur
311  + }
312  + else // si 3 fruits générés
294 313  {
295  -     length++;
314  +     for(int i = 0; i < MAX_FRUITS; i++)
315  +     {
316  +         if(head.x == fruit[i].x && head.y == fruit[i].y)
317  +         {
318  +             fruit_eaten[i] = 1; // marquer le fruit comme mangé
319  +             fruit_eaten_count++; // incrémenter le compteur de fruits mangés
320  +             if(fruit_eaten_count == MAX_FRUITS)
321  +             {
322  +                 length++; // incrémenter la longueur du serpent
323  +                 fruit_gen_count = 0; // réinitialiser le compteur
324  +                 fruit_eaten_count = 0; // réinitialiser le compteur de fruits mangés
325  +                 for(int j = 0; j < MAX_FRUITS; j++)
326  +                 {
327  +                     fruit[j].x = rand()%70;
328  +                     if(fruit[j].x <= 10)
329  +                         fruit[j].x += 11;
330  +                     fruit[j].y = rand()%30;
331  +                     if(fruit[j].y <= 10)
332  +                         fruit[j].y += 11;
333  +                     fruit_eaten[j] = 0; // initialiser l'état du fruit à non mangé
334  +                 }
335  +             }
336  +             break;
337  +         }
338  +     }
339  + }
```

Le but de celle-ci est de pouvoir gérer le fait de devoir manger 3 pommes pour permettre au serpent de s'agrandir

L'un des problèmes rencontré était l'affichage constant des fruits à la même position après les avoir mangés

# Fonction Sauvegarde / Chargement

Extrait de code des fonction chargement et sauvegarde

La fonction sauvegarde va écrire dans un fichier texte les positions des fruits et du serpent

Pour charger la partie sauvegardé, on va charger les dernière positions écrite dans le fichier de sauvegarde

Cette fonctionnalité est disponible uniquement lors de la partie.

- S / s = sauvegarder
- L / l = charger

```
91 + FILE *file = fopen("savegame.txt", "w");
92 + if (file == NULL)
93 + {
94 +     printf("Erreur lors de l'ouverture du fichier de sauvegarde.\n");
95 +     return;
96 + }
88 97
89 97 - do
98 + // Sauvegarder les données de la partie
99 + fprintf(file, "%d\n", length);
100 + fprintf(file, "%d\n", head.x);
101 + fprintf(file, "%d\n", head.y);
102 + fprintf(file, "%d\n", head.direction);
103 + fprintf(file, "%d\n", bend_no);
104 + for (int i = 0; i < bend_no; i++)
90 105 {
106 +     fprintf(file, "%d %d\n", bend[i].x, bend[i].y);
107 + }
108 + for (int i = 0; i < MAX_FRUITS; i++)
109 + {
110 +     fprintf(file, "%d %d %d\n", fruit[i].x, fruit[i].y, fruit_eaten[i]);
111 + }
91 112
92 - food();
93 - fflush(stdin);
113 + fclose(file);
114 + printf("Partie sauvegardée avec succès.\n");
115 + }
94 116
95 - len=0;
117 + void loadGame()
118 + {
119 +     FILE *file = fopen("savegame.txt", "r");
120 +     if (file == NULL)
121 +     {
122 +         printf("Erreur lors de l'ouverture du fichier de sauvegarde.\n");
123 +         return;
124 +     }
96 125
97 - for(i=0; i<30; i++)
126 + // Charger les données de la partie
127 + fscanf(file, "%d\n", &length);
128 + fscanf(file, "%d\n", &head.x);
129 + fscanf(file, "%d\n", &head.y);
130 + fscanf(file, "%d\n", &head.direction);
131 + fscanf(file, "%d\n", &bend_no);
132 + for (int i = 0; i < bend_no; i++)
133 + {
134 +     fscanf(file, "%d %d\n", &bend[i].x, &bend[i].y);
135 + }
136 + for (int i = 0; i < MAX_FRUITS; i++)
137 + {
138 +     fscanf(file, "%d %d %d\n", &fruit[i].x, &fruit[i].y, &fruit_eaten[i]);
139 + }
98 140
99 - {
```



# Mode à deux vitesse

Extrait du code pour le choix du mode à deux vitesse (Rapide / Lent)

A l'exécution du code on laisse au joueur le choix de la vitesse de jeu

- 1 = Lent
- 2 = Rapide

```
86 +  
87 + void Menu()  
88 + {  
89 +     int choix, delay;  
90 +  
91 +     printf("Bienvenue dans le jeu de Snake !\n");  
92 +     printf("Veuillez sélectionner la vitesse de jeu :\n");  
93 +     printf("1. Mode lent\n");  
94 +     printf("2. Mode rapide\n");  
95 +     printf("Entrez votre choix : ");  
96 +     scanf("%d", &choix);  
97 +  
98 +     if (choix == 1)  
99 +     {  
100 +         // Mode lent  
101 +         delay = 100; // ajustez la valeur de delay pour obtenir la vitesse désirée  
102 +     }  
103 +     else if (choix == 2)  
104 +     {  
105 +         // Mode rapide  
106 +         delay = 10; // ajustez la valeur de delay pour obtenir la vitesse désirée  
107 +     }  
108 +     else  
109 +     {  
110 +         printf("Choix invalide. Veuillez réessayer.\n");  
111 +         Menu();  
112 +     }  
113 + }  
114 +  
115 +
```

```
Bienvenue dans le jeu de Snake !  
Veuillez selectionner la vitesse de jeu :  
1. Mode lent  
2. Mode rapide  
Entrez votre choix : |
```

# Image gameplay

```
Bienvenue dans le jeu de Snake !
Veuillez selectionner la vitesse de jeu :
1. Mode lent
2. Mode rapide
Entrez votre choix : |
```

loading...

SCORE : 0

Life : 3

\*\*\*\*\*

F

F

**PROJET**

# **Bilan de compétence**

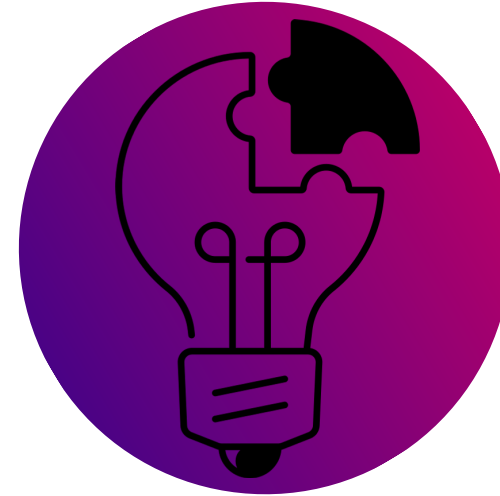
**SNAKE**





## Adaptation

Savoir s'adapter à un code d'une autre personne.



## Compréhension

Comprendre le code et le programme d'un autre pour pouvoir s'adapter



## Chef de projet

Apprendre le rôle de chef de projet et de savoir manager un groupe



## Compétence

Acquérir de nouvelle compétence avec l'utilisation de git, ou d'approfondir les connaissance en C

PROJET ↗

SNAKE

Merci

