

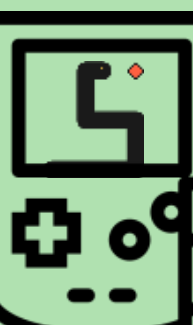
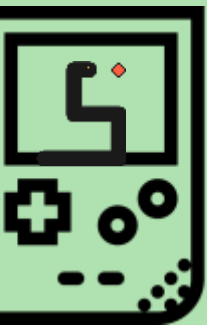


BTS SIO
SLAM 2 A

PROJET SNAKE



- Salhi Djamil
- Baghdadi Youssef
- Harnoufi Anas
- Mukendi Robby





01

PRÉSENTATION DU
PROJET
SNAKE, EQUIPE

02

GESTION DU
PROJET
TRELLO, GANTT, PERT

03

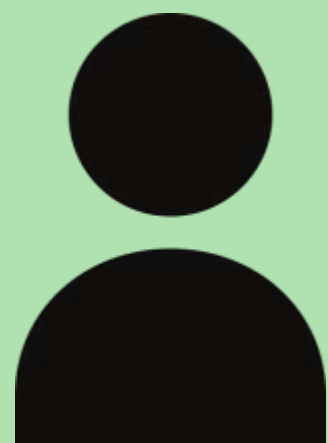
DÉVELOPPEMEN
T
PROBLÈME ET
SOLUTION

04

BILAN DE
COMPÉTENCE



Notre équipe



**Youssef
Baghdadi
Chef de projet**



**Harnoufi Anas
développeur**



**Djamil Salhi
Développeur**



**Mukendy
Robby
Développeur**

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 utilisé : Conio

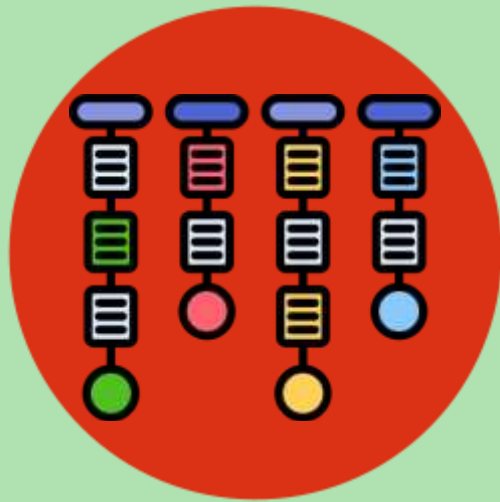


SNAKE
GAME



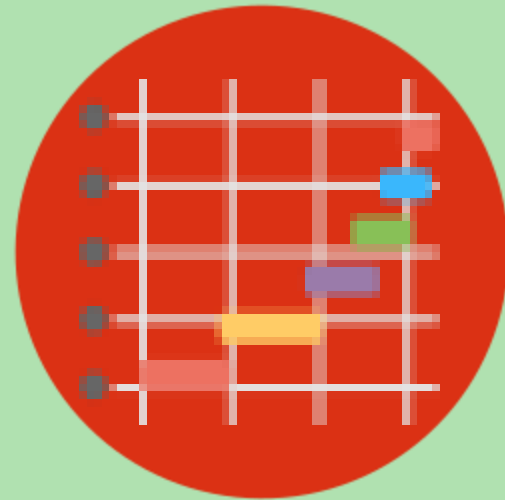


Gestion de Projet



Trello

Planification des
différentes tâche 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

Trello

The screenshot displays a Trello workspace titled "snake-game". The interface includes a left sidebar with navigation options like "Tableaux", "Membres", and "Paramètres d'espace de travail". The main area shows a Kanban board with three columns: "À faire", "En cours", and "Terminé". Each column has a header with a plus icon and a three-dot menu icon, and a button to "Ajouter une carte". The "Terminé" column contains a list of tasks, including "Organisation du projet", "Créer un GitHub collectif", "Utiliser et comprendre GitHub", "Trouver un code de jeux de serpent", "Comprendre et annoter le code", "Répartir le travail", "Modification des règles de jeu", "Fonctionne de sauvgarde", "Programmer le jeu avec l'ordi avec deux vitesses.", and "Présentation PPT". The top right of the interface features links for "Power-ups", "Automatisation", and "Filtres", along with user avatars.

Espace de travail Trello
Gratuit

snake-game ☆ Visible par l'espace de travail Tableau

Power-ups Automatisation Filtres Y AH S

À faire ++ ...
+ Ajouter une carte

En cours ++ ...
+ Ajouter une carte

Terminé ++ ...

+ Ajoutez une autre liste

Organisation du projet

Créer un GitHub collectif

Utiliser et comprendre GitHub

Trouver un code de jeux de serpent

Comprendre et annoter le code

Répartir le travail

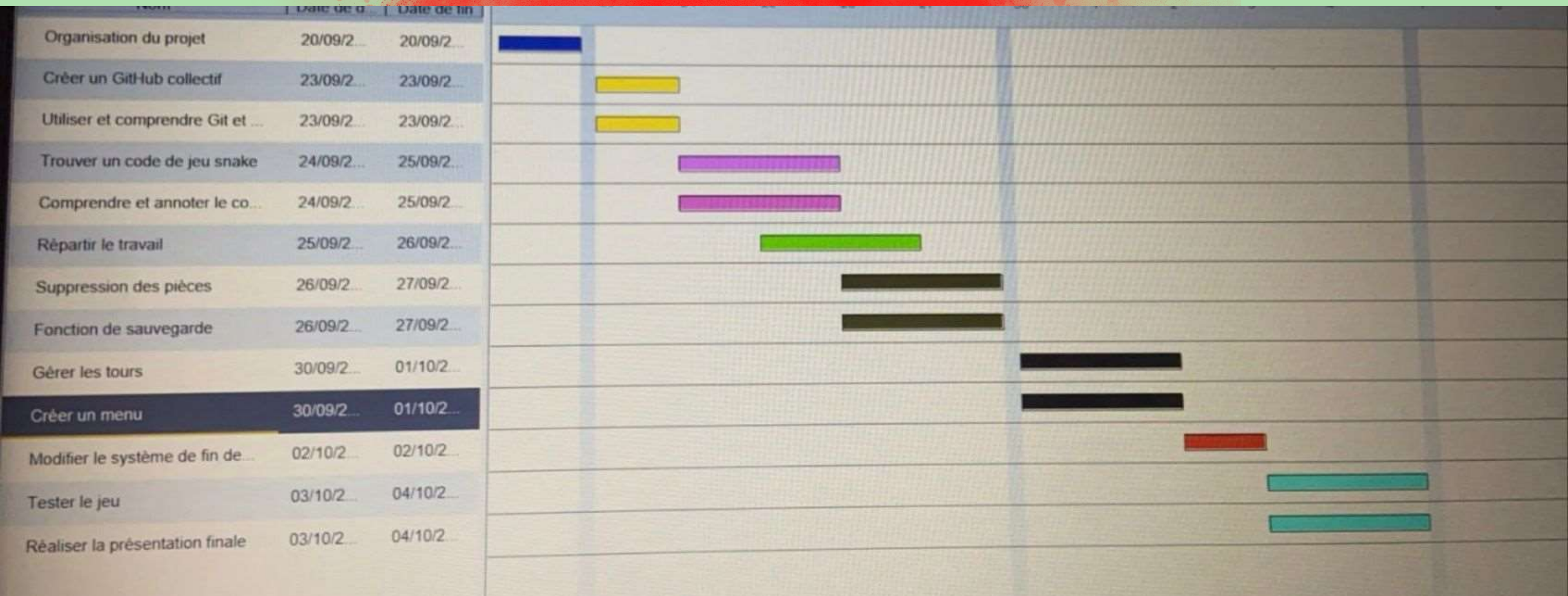
Modification des règles de jeu

Fonctionne de sauvgarde

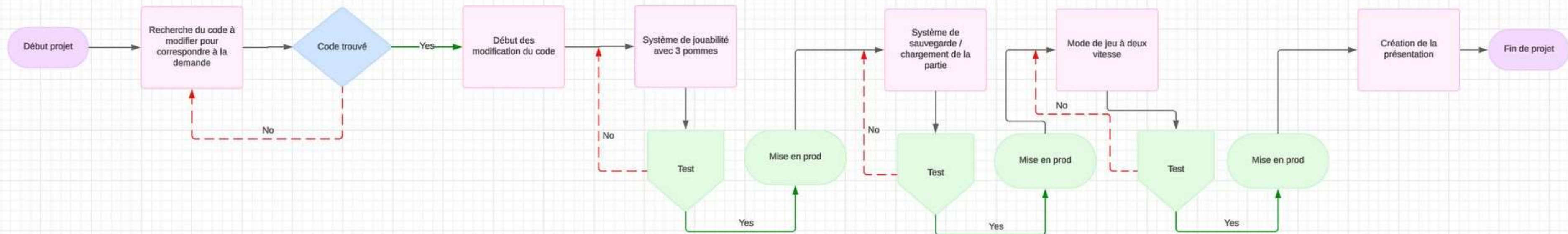
Programmer le jeu avec l'ordi avec deux vitesses.

Présentation PPT

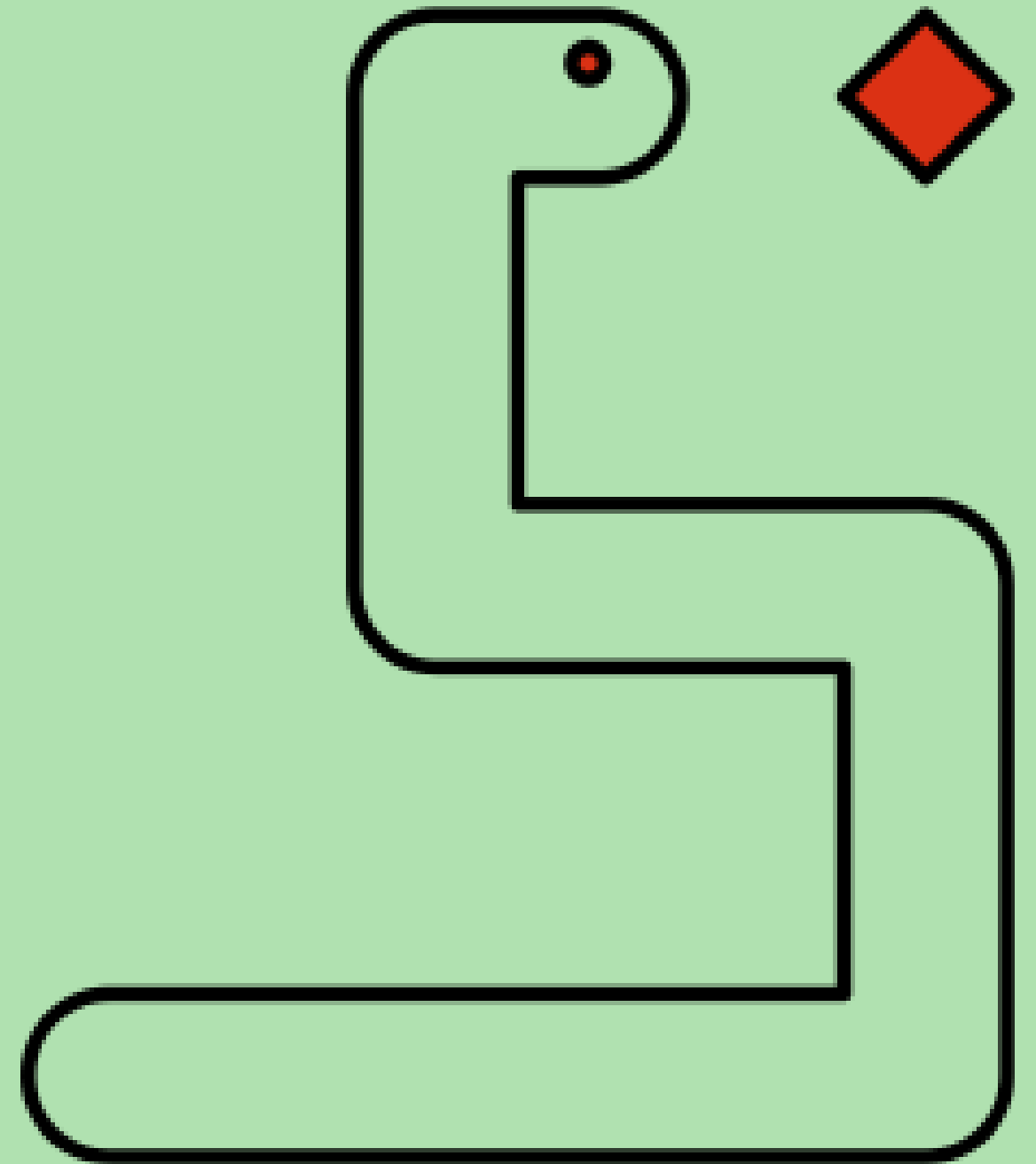
Gantt



Pert



Developpement du jeu



Gérer l'accès

Ajouter des personnes

☐

Sélectionner tout

Taper

Q

Trouver un collaborateur...

anas iris

Collaborateur

Djamilsalhi

En attente de la réponse de Djamilsalhi

Invitation en attente

robbymukendi

En attente de la réponse de robbymukendi

Invitation en attente

Obtenez des contrôles d'accès d'équipe et des discussions pour vos contributeurs au sein d'une organisation.

NOUVEAU Les dépôts privés et les membres illimités sont gratuits.

Créer une organisation

Rohanvp07 / Snake-Game-in-C

<> Code

Issues

Pull requests

Actions

Projects

Security

Insights

Files

main

+

Q

Go to file

t

README.md

record.txt

snake game.c

snake game.exe

Snake-Game-in-C / snake game.c

Rohanvp07 Add files via upload

Code

Blame

546 lines (461 loc) · 11.2 KB

Code 55% faster with GitHub Copilot

1#include <stdio.h>

2#include <time.h>

3#include <stdlib.h>

4#include <conio.h>

5#include <time.h>

6#include <ctype.h>

7#include <time.h>

8#include <windows.h>

9#include <process.h>

10

11#define UP 72

12#define DOWN 80

13#define LEFT 75

14#define RIGHT 77

15

16int length;

17int bend_no;

18int len;

19char key;

20void record();

21void load();

22int life;

23void Delay(long double);

24void Move();

25void Food();

GitHub / Git

BAGHDADI-Youssef / jeu-sanke-en-c-

<> Code

Problèmes

Demandes d'extraction

Actes

Projets

Wiki

Sécurité

Connaissances

Paramètres

jeu-sanke-en-c- Publique

Épingle

Ne plus regarder 1

Fourchette 0

Étoile 0

maître

1 succursale

0 Mots clés

Accéder au fichier

Ajouter un fichier

<> Code

01020119743 Engagement initial 1979681 · il y a 1 heure 1 Engagement

SFML-2.3.2 Engagement initial il y a 1 heure

1.dat Engagement initial il y a 1 heure

2.dat Engagement initial il y a 1 heure

Makefile.win Engagement initial il y a 1 heure

Sans titre.cpp Engagement initial il y a 1 heure

Sans titre.o Engagement initial il y a 1 heure

Sans titre1.cpp Engagement initial il y a 1 heure

ess.dat Engagement initial il y a 1 heure

jeux.c Engagement initial il y a 1 heure

main.cpp Engagement initial il y a 1 heure

main.o Engagement initial il y a 1 heure

monn.exe Engagement initial il y a 1 heure

monapplication.exe Engagement initial il y a 1 heure

À propos

Aucune description, site Web ou sujet fourni.

Activité

0 étoiles

1 regarder

0 fourchettes

Communiqués de presse

Aucun communiqué publié

Créer une nouvelle version

Paquets

Aucun package publié

Publiez votre premier package

Langues

C++ 90,6%

C 4,4%

CSS 2,6%

CMake 1,7%

Autre 0,7%

Flux de travail suggérés

En fonction de votre pile technologique

Fonction Food

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

```
1 void Food() {  
2     // Si 3 pommes sont mangées, Le serpent s'agrandit  
3     if (applesEaten == 3) {  
4         length++; // Augmenter La Longueur du serpent  
5         applesEaten = 0;  
6     }  
7  
8     // Parcourir Les 3 pommes  
9     for (int i = 0; i < 3; i++) {  
10        // Vérifier si La tête du serpent se trouve à La position de La pomme  
11        if (head.x == apples[i].x && head.y == apples[i].y) {  
12            applesEaten++; // Incréments Le compteur de pommes mangées  
13            // Générer une nouvelle position pour La pomme mangée  
14            apples[i].x = rand() % 70;  
15            if (apples[i].x <= 10)  
16                apples[i].x += 11;  
17            apples[i].y = rand() % 30;  
18            if (apples[i].y <= 10)  
19                apples[i].y += 11;  
20        }  
21        // Si La pomme n'a pas encore de position (initiale), on La génère  
22        else if (apples[i].x == 0) {  
23            apples[i].x = rand() % 70;  
24            if (apples[i].x <= 10)  
25                apples[i].x += 11;  
26            apples[i].y = rand() % 30;  
27            if (apples[i].y <= 10)  
28                apples[i].y += 11;  
29        }  
30  
31        // Afficher La pomme à sa position actuelle
```


Fonction Sauvegarde / Chargement

```
// Sauvegarder la partie
void saveGame(char* filename, int showMessage) {
    FILE *file = fopen(filename, "wb"); // Ouvrir le fichier en mode écriture binaire
    if (file == NULL) {
        printf("Erreur lors de la sauvegarde de la partie dans le fichier: %s\n", strerror(errno));
        return;
    }

    // Sauvegarder les informations importantes sur le jeu
    fwrite(&head, sizeof(coordinate), 1, file); // Sauvegarder la position et la direction de la tête
    fwrite(&length, sizeof(int), 1, file); // Sauvegarder la longueur du serpent
    fwrite(body, sizeof(coordinate), length, file); // Sauvegarder le corps du serpent
    fwrite(&bend_no, sizeof(int), 1, file); // Sauvegarder le nombre de courbes du serpent
    fwrite(bend, sizeof(coordinate), bend_no, file); // Sauvegarder les positions des courbes
    fwrite(apples, sizeof(apples), 1, file); // Sauvegarder les positions des pommes
    fwrite(&applesEaten, sizeof(int), 1, file); // Sauvegarder le nombre de pommes mangées
    fwrite(&life, sizeof(int), 1, file); // Sauvegarder le nombre de vies restantes

    fclose(file); // Fermer le fichier après l'écriture

    // Si showMessage est vrai, afficher un message de confirmation
    if (showMessage) {
        printf("Partie sauvegardée dans le fichier: %s\n", filename);
    }
}
```

```
// Charger la partie
void loadGame(char* filename) {
    FILE *file = fopen(filename, "rb"); // Ouvrir le fichier en mode lecture binaire
    if (file == NULL) {
        printf("Erreur: Fichier %s non trouvé ou corrompu : %s.\n", filename, strerror(errno));
        exit(1);
    }

    // Charger les informations sauvegardées
    fread(&head, sizeof(coordinate), 1, file); // Charger la position et la direction de la tête
    fread(&length, sizeof(int), 1, file); // Charger la longueur du serpent
    fread(body, sizeof(coordinate), length, file); // Charger le corps du serpent
    fread(&bend_no, sizeof(int), 1, file); // Charger le nombre de courbes du serpent
    fread(bend, sizeof(coordinate), bend_no, file); // Charger les positions des courbes
    fread(apples, sizeof(apples), 1, file); // Charger les positions des pommes
    fread(&applesEaten, sizeof(int), 1, file); // Charger le nombre de pommes mangées
    fread(&life, sizeof(int), 1, file); // Charger le nombre de vies restantes

    fclose(file); // Fermer le fichier après la lecture

    // Réinitialiser la première position de courbe
    bend[0] = head;

    // Nettoyer l'écran et afficher la partie chargée
    system("cls");
    printf("Partie chargée depuis le fichier: %s !\n", filename);
    Boarder(); // Afficher la bordure du jeu

    // Afficher le serpent et les éléments du jeu à leur position lors de la sauvegarde
    displaySnake(); // Afficher le serpent complet

    Delay(1, 1); // Attendre un moment pour éviter une collision immédiate
    Move(1, filename); // Reprendre le jeu
}
```

Extrait de code des fonction chargement et sauvegarde

La fonction sauvegarde {save} :
Sauvegarde les données du jeu

**Pour charger la partie sauvegardée,
on va charger les dernières positions
écrites dans le fichier de sauvegarde**

**Cette fonctionnalité est disponible
uniquement lors de la partie.**

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

Mode à deux vitesses

```
Creation d'une nouvelle partie...  
Choisissez une vitesse :  
1. Lente  
2. Rapide
```

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

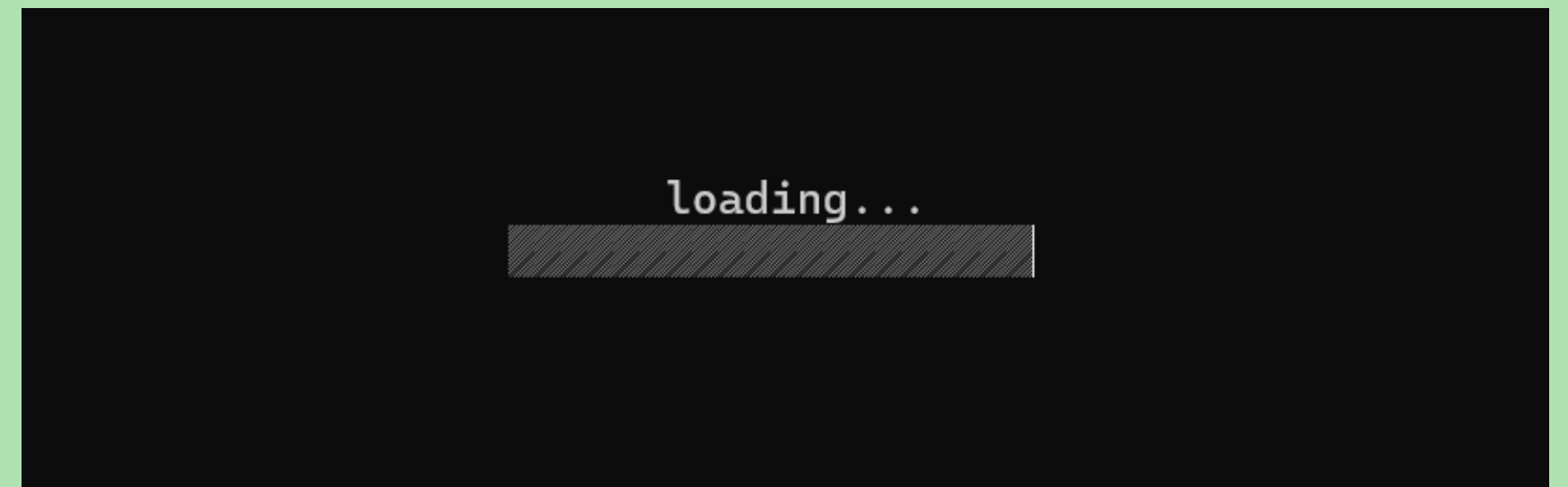
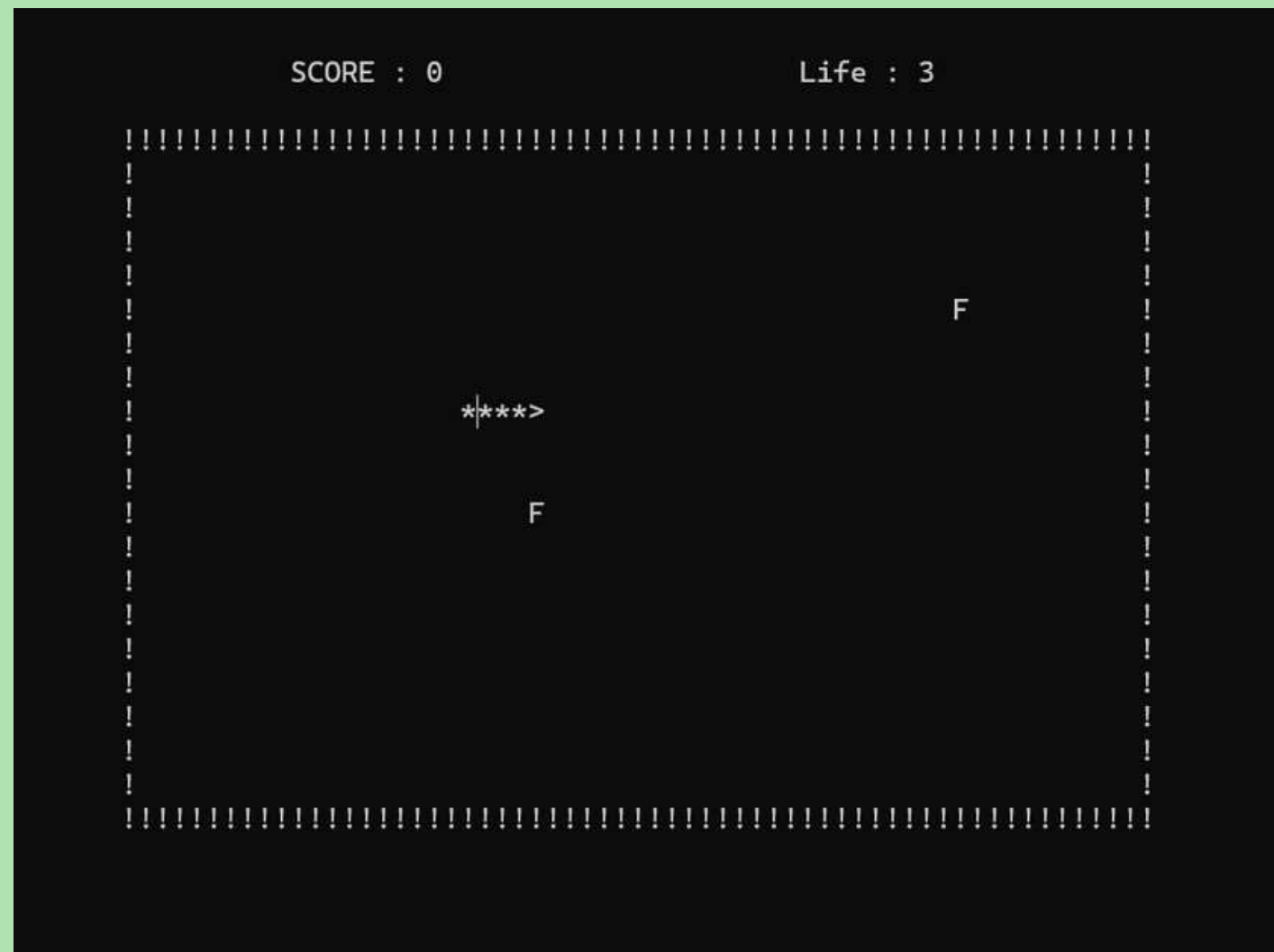
A l'exécution du code on laisse au joueur le choix de la vitesse de jeu

- 1 = Lent
- 2 = Rapide

```
/ Fonction pour gérer le délai (vitesse du serpent)  
void Delay(long double k, int withInput) {  
    long double i;  
    for(i=0; i<=(10000000); i++);  
    if(withInput == 1 && k == 1) {  
        k=0.01;  
        usleep(k * 1000000);  
    }  
    usleep(k * 1000000); // Attendre pour ajuster la vitesse  
}
```

Image gameplay

```
Bienvenue dans Snake !  
1. Nouvelle Partie  
2. Charger une Partie  
Votre choix : 1  
Entrez le nom de fichier pour sauvegarder cette partie (sans extension) : test
```



Bilan de compétence



Bilan de compétence



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

Merçi



SNAKE
GAME