

**BTS SIO**  
SLAM LM

# Projet Snake

*Cariva*

**Benjamin, Yasser, Ryles**

01

**Présentation du Projet**  
*Snake, Equipe*

02

**Gestion du projet**  
*Trello, Gantt, Pert*

05

**Développement**  
*Problème et solution*

04

**Bilan de compétence**

**PROJET**  
SNAKE

# PROJET

## SNAKE

## Présentation du projet

Le but du projet:

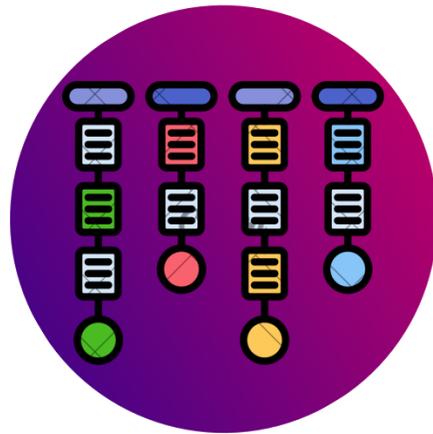
- Génération des pommes par 3 à la fois
- Manger 3 pommes pour agrandir le serpent
- Système de sauvegarde / chargement de partie
- Mode de jeu à 2 vitesses
- Bibliothèque utilisée : Conio



# PROJET

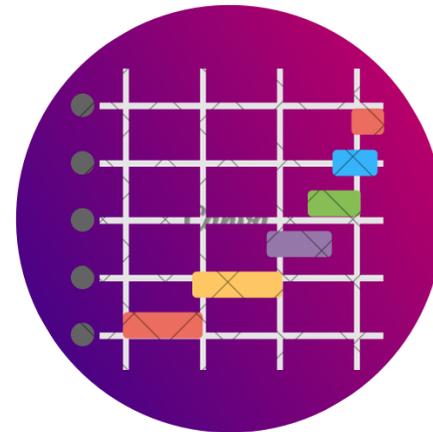
SNAKE

# Gestion de Projet



## 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 displays a Trello board with five columns:

- Réunion**: Contains two cards. The first card is titled "26/09 22H" and lists members Yasser, BenG, and Ryles. The second card is titled "Quelle bibliothèque choisir" and lists the same members.
- 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" with a background image of a waterfall.
- Terminer**: Contains three cards with due dates: "jeux avec deux vitesses" (27 sept. - 29 sept.), "Code Snake" (21 sept. - 23 sept.), and "Génération de pomme" (26 sept. - 29 sept.).

# PROJET

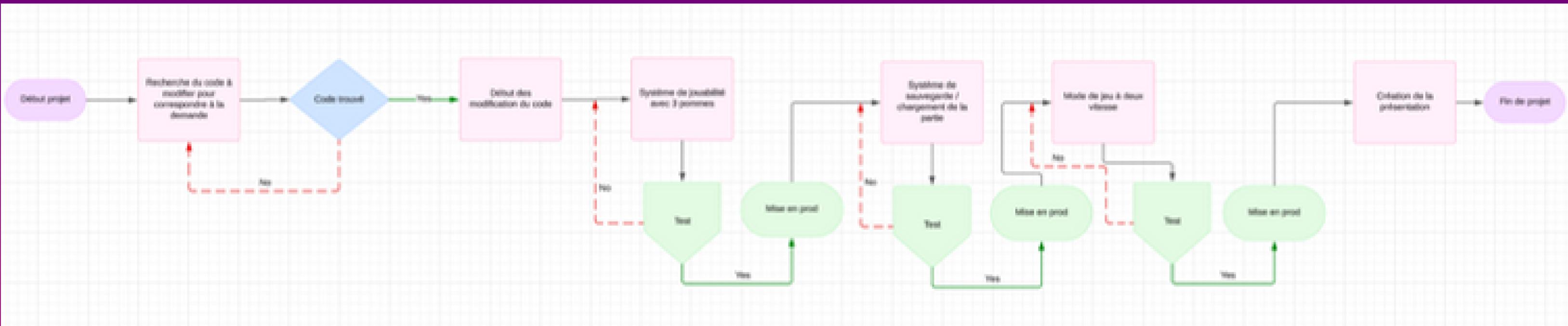
## SNAKE

# Gantt



# PROJET SNAKE

## Pert



# Developpement du jeu



# PROJET SNAKE

# GitHub / Git

Commits

main

All users All time

Commits on Sep 30, 2024

- Implémentation du mode à deux vitesse (Lent/Rapide) BenjaminDev75 committed yesterday e003d21

Commits on Sep 29, 2024

- Système de sauvegarde et de chargement de la parti BenjaminDev75 committed yesterday 5958c8d
- La génération des pommes se fait par trois à BenjaminDev75 committed yesterday b9a544b
- Code source du jeu snake BenjaminDev75 committed yesterday f773a6a
- Initial commit BenjaminDev75 committed yesterday Verified b4b18aa

Projet Classes Debug snake3.c

```
523 printf("");
524 len++;
525 if(len==length)
526     break;
527 }
528 }
529 }
530 }
531 void Boarder()
532 {
533     system("cls");
534     int i;
535     GotoXY(food.x,food.y); /*displaying food*/
536     for(int i = 0; i < MAX_FRUITS; i++)
537     {
```

CONTRÔLE DE CODE SOURCE

Message (Ctrl+Entrée valider et envoyer (...)

Valider et envoyer (push)

snake.c

```
119 int fruit_eaten_count = 0;
120 void SaveGame()
121 {
122     FILE *file = fopen("savegame.txt", "w");
123     if (file == NULL)
124     {
125         printf("Erreur lors de l'ouverture du fichier de sauvegarde
126         return;
127     }
128
129     // Sauvegarder les données de la partie
130     fprintf(file, "%d\n", length);
131     fprintf(file, "%d\n", head.x);
132     fprintf(file, "%d\n", head.y);
133     fprintf(file, "%d\n", head.direction);
134     fprintf(file, "%d\n", bend_no);
135     for (int i = 0; i < bend_no; i++)
136     {
137         fprintf(file, "%d %d\n", bend[i].x, bend[i].y);
138     }
139     for (int i = 0; i < MAX_FRUITS; i++)
140     {
141         fprintf(file, "%d %d %d\n", fruit[i].x, fruit[i].y, fruit_e
142     }
143
144     fclose(file);
```

# Fonction Fruit

```
16 + #define MAX_FRUITS 3
17 +
16 18 int length;
17 19 int bend_no;
18 20 int len;
21
22 @@ -288,11 +290,53 @@ void ExitGame()
23
24 }
25 }
26 }
27
28 + coordinate fruit[MAX_FRUITS];
29 + int fruit_eaten[MAX_FRUITS];
30 + int fruit_gen_count = 0;
31 + int fruit_eaten_count = 0;
32
33 void Food()
34 {
35 - if(head.x==food.x&&head.y==food.y)
36 + if(fruit_gen_count < MAX_FRUITS) // si moins de 3 fruits générés
37 + {
38 +     for(int i = 0; i < MAX_FRUITS; i++)
39 +     {
40 +         fruit[i].x = rand()%70;
41 +         if(fruit[i].x <= 10)
42 +             fruit[i].x += 11;
43 +         fruit[i].y = rand()%30;
44 +         if(fruit[i].y <= 10)
45 +             fruit[i].y += 11;
46 +     }
47 +     fruit_gen_count = MAX_FRUITS; // incrémenter le compteur
48 + }
49 + else // si 3 fruits générés
50 + {
51 +     length++;
52 +     for(int i = 0; i < MAX_FRUITS; i++)
53 +     {
54 +         if(head.x == fruit[i].x && head.y == fruit[i].y)
55 +         {
56 +             fruit_eaten[i] = 1; // marquer le fruit comme mangé
57 +             fruit_eaten_count++; // incrémenter le compteur de fruits mangés
58 +             if(fruit_eaten_count == MAX_FRUITS)
59 +             {
60 +                 length++; // incrémenter la longueur du serpent
61 +                 fruit_gen_count = 0; // réinitialiser le compteur
62 +                 fruit_eaten_count = 0; // réinitialiser le compteur de fruits mangés
63 +                 for(int j = 0; j < MAX_FRUITS; j++)
64 +                 {
65 +                     fruit[j].x = rand()%70;
66 +                     if(fruit[j].x <= 10)
67 +                         fruit[j].x += 11;
68 +                     fruit[j].y = rand()%30;
69 +                     if(fruit[j].y <= 10)
70 +                         fruit[j].y += 11;
71 +                     fruit_eaten[j] = 0; // initialiser l'état du fruit à non mangé
72 +                 }
73 +             }
74 +         }
75 +     }
76 +     break;
77 + }
78 + }
79 + }
```

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 rencontrés 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 - 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 vitesses

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

```
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 : |
```



**PROJET**

# **Bilan de compétence**

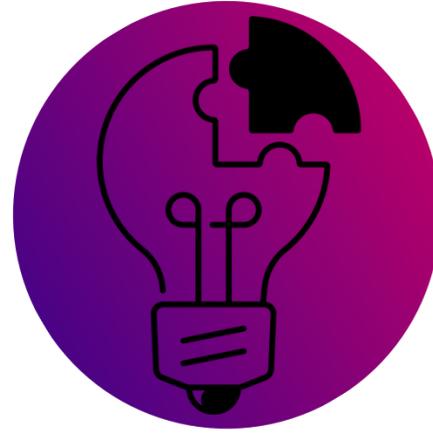
**SNAKE**





## 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

