



Jeu Snake

Jeu Console en Langage C avec Mode IA

Documentation Technique Complète

Auteur	Youma DRAME
Formation	BTS SIO SLAM — École IRIS Paris
Période	Octobre 2025
Technologies	Langage C, algorithmique, console ASCII
Compatibilité	Windows & Linux
Certification	OpenClassrooms — Apprenez à programmer en C (mai 2026)

1. Contexte & Objectifs

Ce projet consiste à développer le jeu classique Snake en langage C pur, avec une interface console ASCII. Le projet inclut un mode joueur manuel et un mode IA automatique qui pilote le serpent vers la nourriture.

Objectifs du projet

- ▶ Implémenter la logique complète du jeu Snake en C
- ▶ Développer un mode joueur (clavier ZQSD + flèches)
- ▶ Créer un mode IA (algorithme glouton vers la pomme)
- ▶ Ajouter un système de sauvegarde/chargement binaire
- ▶ Proposer un menu interactif complet
- ▶ Gérer la vitesse progressive et le score
- ▶ Assurer la compatibilité Windows & Linux

2. Architecture du Code

2.1 Structure des fichiers

Fichiers du projet

- ▶ main.c — Point d'entrée, menu principal, boucle de jeu
- ▶ snake.c / snake.h — Structures et fonctions du serpent
- ▶ map.c / map.h — Gestion de la grille de jeu et rendu ASCII
- ▶ ai.c / ai.h — Algorithme IA (mode automatique)

▶	save.c / save.h — Sauvegarde/chargement binaire (.sav)
▶	input.c / input.h — Gestion clavier (Windows: conio.h / Linux: termios)

2.2 Structures de données

Structure principale du serpent :

```
typedef struct {      int x[MAX_LEN];      int y[MAX_LEN];      int longueur;      int direction; // 0=Nord 1=Est 2=Sud 3=Ouest      int score;      int vitesse; } Serpent;
typedef struct {      int x;      int y; } Pomme;
```

3. Algorithmes Clés

3.1 Boucle de jeu principale

Déroulement de la boucle	
▶	1. Afficher le menu (nouvelle partie / charger / instructions / quitter)
▶	2. Initialiser le serpent au centre de la grille
▶	3. Placer une pomme aléatoirement
▶	4. Boucle principale : lire input → calculer nouvelle position → vérifier collisions → mettre à jour affichage
▶	5. Si pomme mangée : allonger le serpent + incrémenter score + placer nouvelle pomme + accélérer
▶	6. Si collision mur ou corps : game over → afficher score final

3.2 Algorithme IA (mode automatique)

L'IA utilise un algorithme glouton qui calcule la direction optimale vers la pomme :

- Calculer les 4 directions possibles (Nord, Est, Sud, Ouest)
- Éliminer les directions menant à un mur ou au corps du serpent
- Parmi les directions valides, choisir celle qui minimise la distance à la pomme
- Distance utilisée : distance de Manhattan $|dx| + |dy|$

3.3 Système de sauvegarde binaire

La sauvegarde écrit l'état complet du jeu dans un fichier binaire :

```
// Sauvegarde void sauvegarder(Serpent *s, Pomme *p) {      FILE *f =
fopen("saves/game.sav", "wb");      fwrite(s, sizeof(Serpent), 1, f);      fwrite(p,
sizeof(Pomme), 1, f);      fclose(f); } // Chargement void charger(Serpent *s, Pomme
*p) {      FILE *f = fopen("saves/game.sav", "rb");      fread(s, sizeof(Serpent), 1,
f);      fread(p, sizeof(Pomme), 1, f);      fclose(f); }
```

4. Fonctionnalités du Menu

Option 1	Nouvelle partie — réinitialise le serpent et le score
Option 2	Charger une sauvegarde — reprend depuis saves/game.sav
Option 3	Mode IA — le serpent joue automatiquement
Option 4	Instructions — affiche les contrôles clavier
Option 5	Quitter — ferme proprement le programme

5. Tests Réalisés

Test 1	Serpent mange pomme → Score incrémenté, nouveau segment ajouté ✓
Test 2	Collision mur → Game over immédiat ✓
Test 3	Collision avec son propre corps → Game over ✓
Test 4	Sauvegarde → Fichier game.sav créé correctement ✓
Test 5	Chargement sauvegarde → Reprise exacte de l'état ✓
Test 6	Mode IA → Serpent atteint les pommes sans collision ✓
Test 7	Compilation Linux (GCC) → Programme fonctionnel ✓

6. Bilan

Compétences développées

- ▶ Maîtrise du langage C : structures, pointeurs, fichiers binaires
- ▶ Algorithmique : gestion de tableaux, boucles de jeu, IA basique
- ▶ Portabilité : gestion des différences Windows/Linux (conio.h vs termios.h)
- ▶ Gestion mémoire : allocation statique, structures imbriquées

Ce projet a été complété par l'obtention de la certification OpenClassrooms 'Apprenez à programmer en C' (mai 2026), validant officiellement les compétences acquises.