



Le Jeu Snake en Langage C

Une exploration technique d'un projet de programmation emblématique, de la conception à l'implémentation.

Sy Ama Mariane

Drame Youma

Noubissi Kenmogne Marc-Bryan

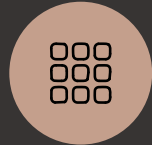
Un Classique Intemporel du Gaming

Snake est l'un des jeux vidéo les plus emblématiques de l'histoire, popularisé par les téléphones Nokia dans les années 90. Sa simplicité apparente cache une richesse de concepts de programmation fondamentaux.

Créer Snake en C représente un excellent exercice pour maîtriser les structures de données, la gestion de la mémoire, et la programmation temps réel. C'est un projet parfait pour consolider vos compétences en développement.



Architecture Globale du Projet



Gestion du Terrain

Matrice 2D représentant l'aire de jeu avec détection des bordures et des collisions



Logique du Serpent

Liste chaînée pour gérer les segments, déplacement fluide et croissance dynamique



Système de Nourriture

Génération aléatoire de positions et calcul des points de score



Interface Utilisateur

Affichage console avec ncurses et gestion des entrées clavier en temps réel

Structures de Données Clés

1

Structure Segment

Chaque segment du serpent contient des coordonnées (x, y) et un pointeur vers le segment suivant, formant une liste chaînée.

2

Structure GameState

Encapsule l'état complet du jeu : position du serpent, direction actuelle, score, statut de la partie, et position de la nourriture.

3

Enum Direction

Type énuméré définissant les quatre directions possibles (HAUT, BAS, GAUCHE, DROITE) pour garantir la cohérence du mouvement.

Boucle Principale de Jeu

```
while (gameState.isRunning) {  
    // 1. Capturer l'entrée utilisateur  
    processInput(&gameState);  
  
    // 2. Mettre à jour la logique  
    updateSnake(&gameState);  
    checkCollisions(&gameState);  
  
    // 3. Rendu visuel  
    renderGame(&gameState);  
  
    // 4. Contrôle du timing  
    usleep(GAME_SPEED);  
}
```

Cette boucle événementielle assure un gameplay fluide à 60 images par seconde. La fonction `usleep()` contrôle la vitesse du jeu, tandis que les fonctions modulaires facilitent la maintenance et les tests.

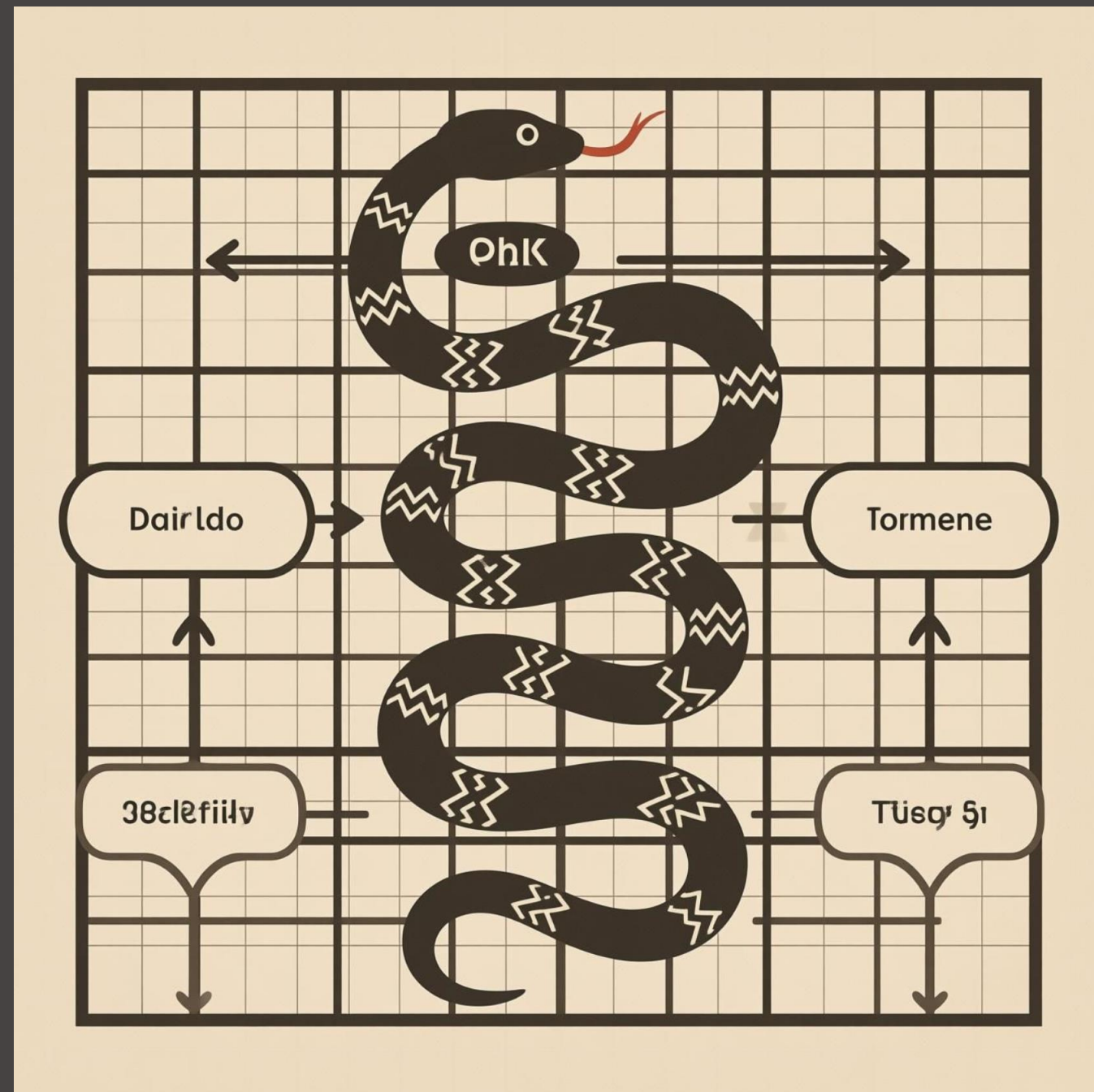
Gestion du Mouvement

Algorithme de Déplacement

Le serpent se déplace en ajoutant une nouvelle tête dans la direction actuelle et en supprimant la queue, sauf quand il mange.

Défi technique : Empêcher les demi-tours instantanés qui causeraient une collision immédiate avec le corps.

La validation de direction compare l'input avec la direction actuelle avant de l'accepter.



Détection de Collisions



Bordures du Terrain

Vérification si la tête du serpent dépasse les limites de la matrice de jeu



Auto-collision

Parcours de la liste chaînée pour détecter si la tête touche un segment du corps



Capture de Nourriture

Comparaison des coordonnées pour déclencher la croissance et incrémenter le score

Ces trois types de collisions sont évalués à chaque itération de la boucle pour maintenir l'intégrité du jeu et assurer une expérience utilisateur cohérente.

Optimisations et Améliorations



Niveaux de Difficulté

Réduction progressive du délai `usleep()` pour accélérer le jeu à mesure que le score augmente



Système de High Score

Persistance des meilleurs scores dans un fichier texte avec lecture/écriture via `fopen()` et `fprintf()`



Personnalisation Visuelle

Utilisation de `init_pair()` pour définir des schémas de couleurs personnalisés et améliorer l'esthétique

Compétences Développées



100%

Structures de données dynamiques

Maîtrise des listes chaînées et allocation mémoire



100%

Programmation événementielle

Gestion de boucles temps réel et entrées asynchrones



100%

Logique algorithmique

Résolution de problèmes de collision et mouvement

Ce projet Snake en C constitue une excellente introduction à la programmation de jeux et aux concepts avancés du langage. Il combine théorie et pratique pour créer une expérience d'apprentissage complète et engageante.