

Construction du Jeu Pacman avec Allegro

Objectifs

1. Concevoir et développer un jeu vidéo inspiré du classique **Pacman**, en utilisant la bibliothèque graphique **Allegro**.
2. Comprendre les concepts de programmation graphique, gestion des événements, animation et interaction dans un environnement C.

Étapes du Projet

1. Préparation et Initialisation

1. Définition des besoins :

- Pacman doit se déplacer dans un labyrinthe en collectant des points.
- Les fantômes doivent poursuivre Pacman.
- Implémenter les règles principales :
 - Pacman perd une vie s'il est touché par un fantôme.
 - Le jeu est gagné lorsque tous les points sont collectés.

2. Structure du projet :

- Fichiers sources (main.c, game.c, etc.).
- Headers pour structurer le code (game.h, graphics.h, etc.).
- Répertoire pour les assets (images, sons, etc.).

2. Conception

1. Conception du Labyrinthe :

- Représenter le labyrinthe sous forme de matrice 2D (int labyrinthe [rows][cols]).
- 0 pour un chemin libre, 1 pour un mur, 2 pour un point à collecter.

2. Gestion des Entités :

- Définir des structures pour Pacman, les fantômes et les points :

```
typedef struct {  
    int x, y; // Position actuelle  
    int direction; // Direction de déplacement  
} Entity;
```

3. Cycle de jeu :

- Initialiser les entités.
- Gérer les événements (clavier pour Pacman, pour les fantômes).
- Mettre à jour les positions et vérifier les collisions.
- Dessiner les éléments sur l'écran.

3. Développement

1. Initialisation du jeu :

- Initialiser Allegro et ses modules (affichage, clavier, temporisation).
- Créer une fenêtre et un contexte de jeu.

2. Affichage graphique :

- Charger et afficher le labyrinthe (convertir la matrice en éléments visuels).
- Dessiner Pacman, les fantômes et les points.

3. Gestion des Entrées :

- Implémenter les déplacements de Pacman en fonction des touches (flèches du clavier).
- Assurer que Pacman ne traverse pas les murs (vérification avec la matrice).

4. Déplacement des Fantômes :

- Implémenter un déplacement simple pour les fantômes (par exemple, mouvement aléatoire ou ciblé vers Pacman).
- Ajouter une logique pour détecter les collisions avec Pacman.

5. Gestion des Collisions :

- Vérifier si Pacman collecte un point (mise à jour du score et suppression du point).
- Vérifier si Pacman entre en contact avec un fantôme (perte de vie).

6. Système de Score et Vies :

- Afficher le score et les vies restantes à l'écran.
- Terminer le jeu lorsque toutes les vies sont perdues ou que tous les points sont collectés.

4. Tests et Ajustements

1. Tests unitaires :

- Vérifier les fonctions critiques comme les déplacements, les collisions et la gestion des scores.

2. Optimisation :

- Améliorer les performances graphiques et les temps de réponse.
- Ajuster la difficulté (vitesse des fantômes, taille du labyrinthe).

3. Debugging :

- Résoudre les éventuels bugs liés à la gestion des événements ou aux animations.

5. Ajout de Fonctionnalités

1. Animations :

- Ajouter des animations pour les mouvements de Pacman et des fantômes.
- Effets visuels pour les collisions et la collecte des points.

2. Effets sonores :

- Intégrer des sons pour les déplacements, les collisions et les événements de jeu.

3. Niveaux multiples :

- Créer plusieurs labyrinthes avec une difficulté croissante.

Livrables

1. Code source complet avec commentaires.
2. Exécutable du jeu.
3. Rapport technique (description du projet, choix techniques, difficultés rencontrées).