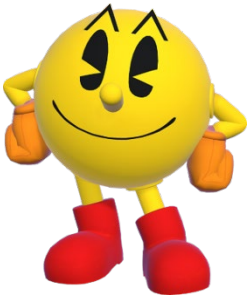
30/03/2024

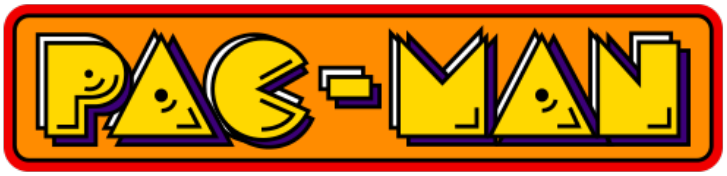
Projet d'Interaction Homme-Machine (IHM) et de programmation Visual C++ sous l'IDE Visual Studio

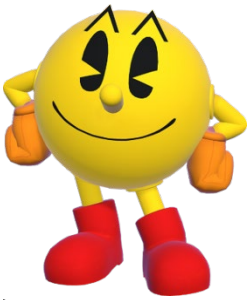
Programmation d'un jeu, le Pacman

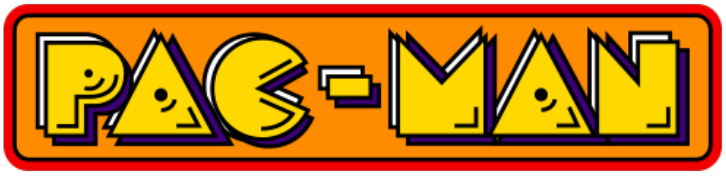
mohamed boudhina/ Gaetan André
SEATECH

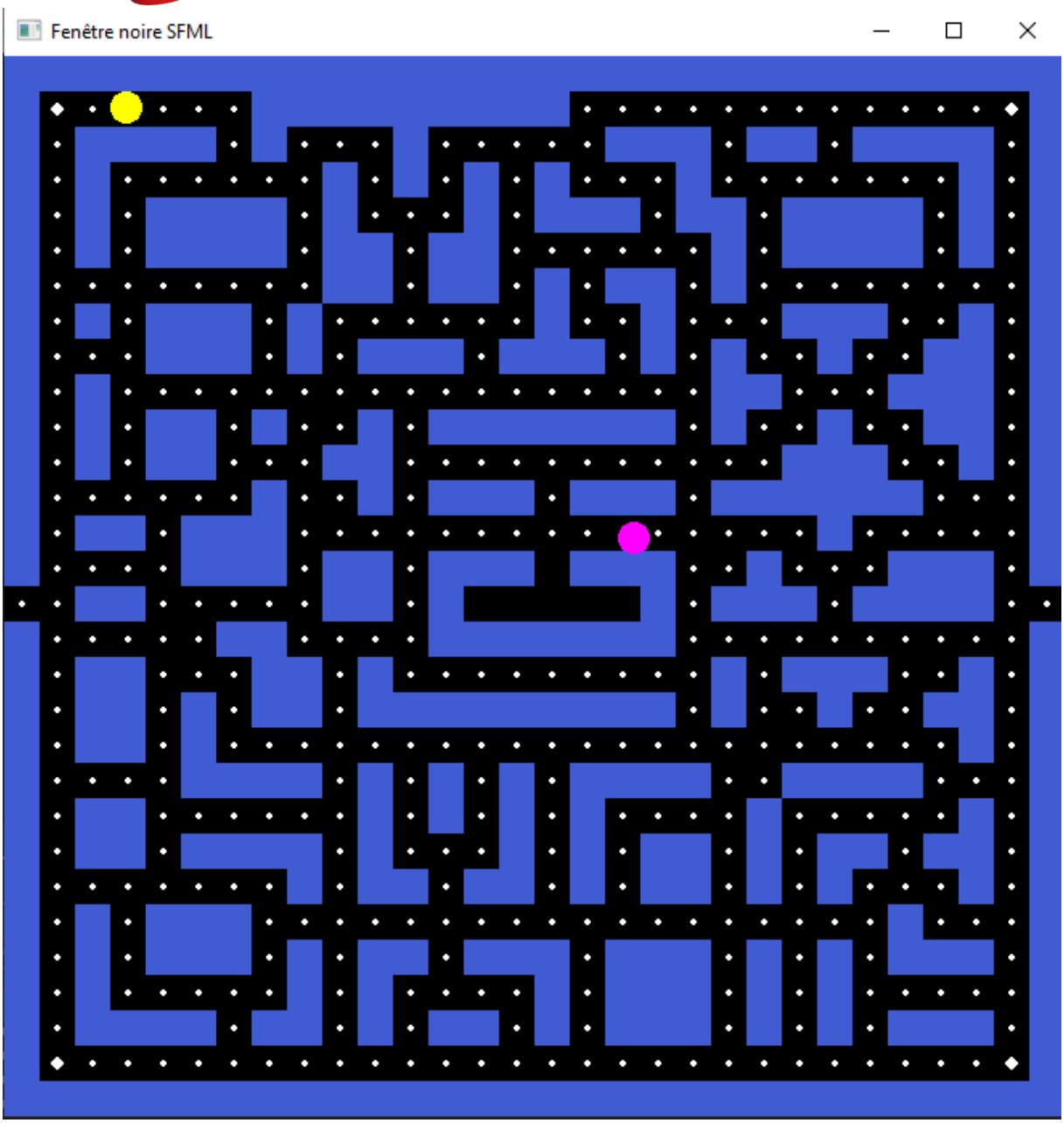
Table des matières








PAC-MAN


Fenêtre noire SFML

Chapitre 1 : Introduction

1.1 Présentation du projet

Ce projet consiste à développer un jeu Pacman en utilisant les concepts de programmation événementielle sous l'environnement de développement Visual Studio. Le jeu Pacman est un classique des années 80, dans lequel un personnage jaune en forme de bouche (Pacman) doit parcourir un labyrinthe pour manger des pac-gommes tout en évitant quatre fantômes qui cherchent à l'attraper.

1.2 Cahier des charges

Le cahier des charges du projet comprend les éléments suivants :

- Le jeu doit comporter au moins deux niveaux de difficulté avec des labyrinthes de taille proportionnelle à la fenêtre d'affichage.

- Le jeu doit être cadencé par un timer, avec la possibilité de le mettre en pause via la touche Espace. La vitesse de jeu doit être réglable via une boîte de dialogue.

- L'affichage du nombre de pac-gommes restantes doit être prévu.

- Le rafraîchissement de l'affichage doit se faire de manière dynamique, en ne redessinant que les éléments qui ont changé depuis la vue précédente.

- L'utilisation de bitmaps est autorisée en alternative aux objets graphiques de Visual C++.

- Les fantômes doivent pouvoir se déplacer de manière intelligente vers Pacman, en utilisant un algorithme de pathfinding de type A* ou Dijkstra.

Chapitre 2 : Objectifs et choix de programmation

2.1 Objectifs du projet

Les objectifs principaux de ce projet de développement d'un jeu Pacman sont les suivants :

1. Concevoir une structure de classes modulaire et évolutive permettant de représenter le jeu Pacman dans son ensemble. Cette structure doit faciliter l'ajout de nouvelles fonctionnalités à l'avenir.

2. Implémenter les fonctionnalités clés du jeu, telles que le déplacement de Pacman, le comportement des fantômes, la gestion des scores et des niveaux.

3. Assurer une bonne fiabilité et une expérience de jeu agréable pour l'utilisateur, en respectant les exigences du cahier des charges.

4. Utiliser l'interface de Visual Studio et la bibliothèque SFML pour l'affichage graphique du jeu, conformément aux recommandations du sujet.

2.2 Choix de programmation

Afin de répondre à ces objectifs, les choix de programmation suivants ont été effectués :

2.2.1 Architecture de l'application

L'application a été conçue selon une architecture orientée objet, avec une structure de classes hiérarchique. Les principales classes sont :

- `Main` : fichier pour représenter les éléments du jeu (Pacman, fantômes, pac-gommes, etc.)

- `Pacman` : classe qui permet de gérer le personnage principal

- `Fantome` : classe qui représente les fantômes

- `Carte` : classe qui permet de dessiner la Map.

Cette structure modulaire permet d'ajouter facilement de nouvelles fonctionnalités à l'avenir, en créant de nouvelles classes dérivées ou en étendant les classes existantes.

2.2.2 Gestion des événements

Le jeu est cadencé par un délai de 33ms entre chaque frame, qui met régulièrement à jour les positions des objets. Les entrées clavier du joueur sont gérées via des événements, permettant de contrôler les déplacements de Pacman.

2.2.3 Algorithmes de déplacement

Pour le déplacement des fantômes, l'algorithme choisi est le déplacement aléatoire des fantômes pour faciliter la programmation, prenant en compte les collisions.

La gestion des changements d'état des fantômes (normal, retour à la base) a également été implémentée, mais les Fantômes sont invulnérables.

2.2.4 Affichage et interface utilisateur

L'affichage graphique du jeu utilise la bibliothèque SFML, intégrée à l'interface de Visual Studio. Cela permet d'avoir un rendu visuel plus élaboré que l'utilisation de bitmaps.

L'affichage des informations du jeu (pac-gommes restantes, etc.) se fait de manière dynamique, en ne redessinant que les éléments qui ont changé depuis la vue précédente.

Chapitre 3 : Échéancier de réalisation

Un diagramme de Gantt détaillant l'échéancier prévisionnel et réel du projet est fourni en annexe.

Les principales étapes du projet sont :

1. Conception de la structure de classes (1 semaine)

2. Implémentation du déplacement de Pacman et de la gestion des collisions (1 semaine)

3. Développement du comportement des fantômes (1 semaine)

5. Ajout des fonctionnalités supplémentaires (écran de victoire, fin de jeu) (1 semaine)

6. Tests, et finalisation du projet (1 semaine)

Des actions correctives ont été mises en place lors de retards sur certaines tâches, permettant de respecter les délais prévus.

Chapitre 4 : Travail réalisé

4.1 Tests pour la validation du code

Afin de garantir le bon fonctionnement de notre jeu Pacman, nous avons mis en place un processus de tests rigoureux. Ces tests ont été conçus pour vérifier chaque partie du jeu et leur bon fonctionnement ensemble.

Nous avons d'abord testé chaque élément de code séparément pour s'assurer qu'il fonctionne correctement de manière individuelle. Puis, nous avons vérifié l'intégration de ces différents éléments et leur interaction au sein du jeu.

Les aspects que nous avons testés incluent :

- Le déplacement des personnages (Pacman et fantômes)

- La détection des collisions (avec les murs, les pac-gommes, les bonus, etc.)

- L'affichage des éléments graphiques (labyrinthe, personnages, etc.)

- Le fonctionnement global du jeu

- La réaction du jeu aux différentes actions du joueur (contrôle de Pacman, écran de victoire, etc.)

Nous avons utilisé diverses situations de jeu pour couvrir un maximum de cas de figure et identifier tout problème potentiel. Après chaque ajout ou modification de code, nous avons effectué des tests approfondis pour valider le bon fonctionnement de l'ensemble.

4.2 Problèmes rencontrés

Malgré nos efforts, nous avons rencontré quelques difficultés lors du développement de notre jeu Pacman :

1. Gestion du Timer :Trouver le bon ratio de temps pour que le jeu ne soit pas trop gourmand en calcul tout en restant fluide a été un défi majeur. Il a fallu ajuster minutieusement le timer pour garantir une expérience de jeu optimale.

2. Déplacements Complexes : Les déplacements des personnages, en particulier celui de Pacman, ont posé des difficultés. Lorsque Pacman changeait de direction, il devait être centré pour éviter qu'il ne se déplace partiellement à travers les murs. De plus, le déplacement par incréments de 4 pixels a compliqué la lecture de la position, surtout près des tunnels de téléportation, provoquant des mesures de position négative et engendrant divers problèmes. Cette situation a été résolue en ajustant le moment où le personnage est téléporté dans une zone de 5 pixels, évitant ainsi les sorties involontaires de la matrice.

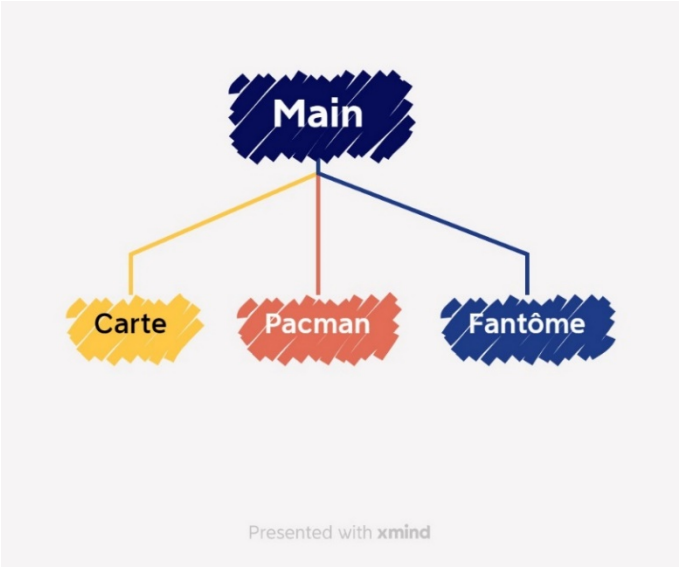
Ces obstacles ont nécessité des ajustements minutieux et des solutions créatives pour garantir le bon fonctionnement du jeu Pacman malgré ces défis techniques rencontrés lors du développement.

3. Installation de la bibliothèque SFML :L'un des principaux défis que nous avons rencontrés a été l'installation et l'intégration de la bibliothèque SFML dans notre projet Visual Studio. Bien que la documentation de SFML soit détaillée, nous avons dû faire face à plusieurs problèmes de configuration, notamment liés aux chemins des bibliothèques et des en-têtes.

Après plusieurs tentatives, nous avons finalement réussi à configurer correctement notre projet pour utiliser SFML. Cependant, ce processus a pris plus de temps que prévu et a retardé le développement de certaines fonctionnalités du jeu.

4.3 Description de la structure du programme

La structure de notre programme Pacman est organisée en plusieurs fichiers, chacun étant responsable d'une partie spécifique du jeu :



Chapitre 5 : Conclusion

5.1 État final du projet

Malgré quelques limitations telles que l'absence d'intelligence artificielle et la non-possibilité de manger les fantômes lorsqu'ils sont invincibles, le projet de développement du jeu Pacman a été mené à bien. L'application respecte les exigences du cahier des charges et offre une expérience de jeu satisfaisante pour l'utilisateur. La structure modulaire du programme, la fiabilité des fonctionnalités clés, et la gestion efficace des événements sont autant de points forts qui ont contribué au succès du projet.

5.2 Points forts et points faibles

Points forts :

- Structure de classes modulaire et évolutive

- Implémentation fiable des fonctionnalités clés du jeu

- Bonne gestion des événements (entrées clavier, timer, etc.)

- Possibilité d'ajouter facilement de nouvelles fonctionnalités à l'avenir

Points faibles :

- Aspects graphiques relativement simples

- Manque d'une interface utilisateur plus élaborée (menus, options, etc.)

- Possibilités d'amélioration du système de gestion des niveaux (chargement dynamique, sauvegarde des scores, etc.)

5.3 Améliorations possibles

Parmi les améliorations possibles pour ce projet, on peut envisager :

- Développer une interface utilisateur plus conviviale avec des menus, des options, etc.

- Ajouter des effets sonores et de la musique pour améliorer l'expérience de jeu

- Implémenter un système de sauvegarde et de chargement des parties

- Enrichir le contenu du jeu avec de nouveaux niveaux, bonus, ennemis, etc.