Rapport de Maxime Drouhin

J’ai effectué ce projet avec Orfeú Mouret.

Structure du code

Nous avons essayé de respecter le modèle de conception Modèle-Vue-Contrôleur. Pour cela, nous avons décidé de structurer le contenu de notre dossier `src` en 3 dossiers : `model`, `view`, et `controller`. Dans chaque dossier, on trouve la classe associée : `GameModel`, `GameView` et `GameController`, dequelles dépendent directement les fichiers du même dossier. A cela, il faut ajouter le dossier `common`, rassemblant des classes et variables utiles qui ne sont pas directement liées à un unique aspect du modèle MVC.

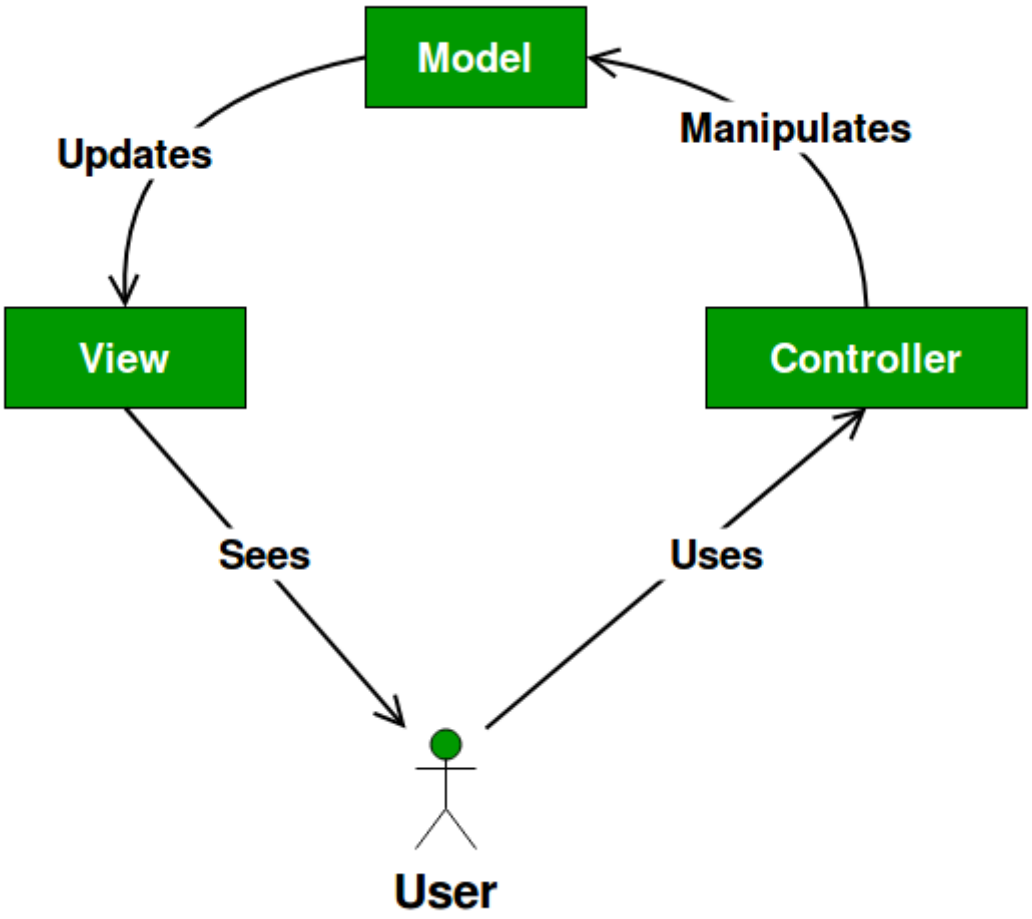
```
.
├── main.cpp
├── controller
│   ├── GameController.cpp
│   └── GameController.h
├── model
│   ├── Blinky.cpp
│   ├── Blinky.h
│   ├── Character.cpp
│   ├── Character.h
│   ├── Clyde.cpp
│   ├── Clyde.h
│   ├── GameModel.cpp
│   ├── GameModel.h
│   ├── Ghost.cpp
│   ├── Ghost.h
│   ├── Inky.cpp
│   ├── Inky.h
│   ├── MonsterDen.cpp
│   ├── MonsterDen.h
│   ├── PacMan.cpp
│   ├── PacMan.h
│   ├── Pinky.cpp
│   └── Pinky.h
├── view
│   ├── GameView.cpp
│   ├── GameView.h
│   └── Sprites.h
└── common
    ├── Direction.cpp
    ├── Direction.h
    ├── GameDimensions.h
    ├── Position.cpp
    ├── Position.h
```

```
    ├── Tile.cpp
    └── Tile.h

4 directories, 31 files
```

Le jeu se lance depuis `main.cpp`, qui instancie, initialise et lance `GameController`. C'est ensuite dans `GameController.cpp` qu'on instancie `GameModel` et `GameView`., et qu'on a la boucle principale où on demande à `GameModel` de se mettre à jour et à `GameView` de dessiner les sprites en utilisant la représentation haut-niveau offerte par `GameModel`.

On respecte donc le schéma suivant :



Fonctionnalités implémentées

Cette partie du rapport est commune à Orfeú et moi.

- affichage du maze

- affichage et logique du score
- affichage et logique du hight score
- affichage et logique des vies
- start
- game over
- affichage et control de pacman
- cornering
- affichage, animation et logique de la mort de pacman
- affichage et logique des gommes
- affichage et logique des energizer
- affichage, animation et logique des fantomes
- logique des modes des fantomes
- logique de deplacements des fantomes en fonction de leur mode
- tournants interdits pour les fantomes
- passage de droite à gauche avec les couloires de teleportation
- affichage, animation et logique de la monster den
- affichage et logique des (du) fruits
- logique des vitesse des fantomes et de pacman

Fonctionnalités non implémentées

Cette partie du rapport est commune à Orfeú et moi.

- L’animation de retour des fantomes dans la Monster Den après avoir été mangé
- Le clignotement de la map après avoir mangé tous les dot
- Le bonus de point si tous les fantomes ont été mangé à chaque fois qu’un energizer a été mangé
- L’animation des points gagné quand on mange un fantome
- L’implémentation correspond au premier niveau de pacman seulement

Mes contributions

Contributions générales

- Gestion des outils de collaboration : Git, Github, centralisation des informations sur le projet dans une page Notion, communications centralisées dans un salon Discord.
- Mise en place de la structure du code initiale respectant le modèle MVC, réorganisation régulière du code pour continuer à le respecter, configuration de CMake et ériture de `CMakeLists.txt`

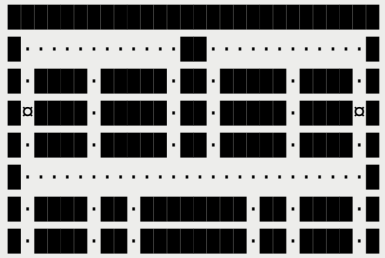
Contributions au code

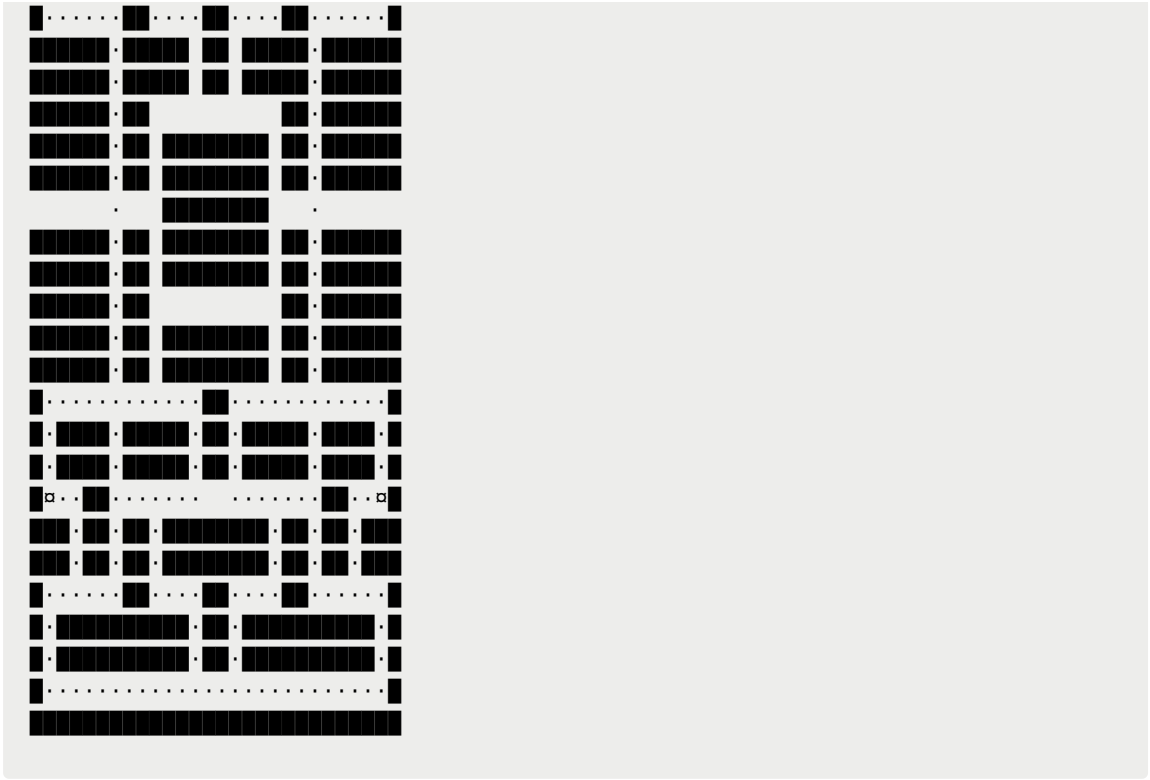
Maxime

- Utilisation de `SDL_GetTicks64()` pour assurer un framerate précis
- Ecriture des fonctions permettant de régler la vitesse des personnages en pourcentage. Elles se basent sur un principe de mouvement cumulé, de manière à ce qu'au bout de `n` frames, le personnage s'est déplacé de `n*speed/100`.
- Séparé complètement la position en pixels du modèle, qui correspond au jeu PacMan original de Namco, de la position en pixels finale qui est affichée (obtenue en mettant à l'échelle de `UPSCALING_FACTOR = 3`). Cela permet de ne pas s'embarrasser des positions réelles des éléments quand on réfléchit à la logique du jeu, et de code tout "au pixel près", ou en tout cas en essayant de respecter au plus près l'implémentation originale.
- Création d'une classe `Position` qui permet de travailler de manière plus abstraite avec des positions, sans s'embarrasser du fait qu'il existe 3 "positions" différentes : la position en "tiles", la position en pixels du centre du sprite, et la position en pixel du pixel en haut à droite des sprites. On peut oublier la position après mise à l'échelle, car elle se fait au dernier moment ! Et grâce à cette classe, pas besoin de s'embarrasser d'une position `top_left` dans le modèle, elle peut simplement être récupérée en cas de besoin par `GameView`.
- Implémentation de l'arrêt de PacMan pendant quelques frames quand il mange des gommes ou des energizers. Cela ralentit PacMan, il prend donc le risque de se faire rattraper par les fantômes !
- Modélisation du labyrinthe comme une matrice.

```
// Maze layout
static constexpr TileType W = TileType::WALL;
static constexpr TileType E = TileType::EMPTY;
static constexpr TileType D = TileType::DOT;
static constexpr TileType Z = TileType::ENERGIZER;
static constexpr TileType INITIAL_TILES_MATRIX[WINDOW_ROWS][WINDOW_COLS] = {
    {E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E},
    {E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E},
    {E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E},
    {W,W,W,W,W,W,W,W,W,W,W,W,W,W,W,W,W,W,W,W,W,W,W,W,W,W,W},
    {W,D,D,D,D,D,D,D,D,D,D,D,W,W,D,D,D,D,D,D,D,D,D,D,D,D,W},
    {W,D,W,W,W,W,D,W,W,W,W,W,D,W,W,D,W,W,W,W,W,D,W,W,W,W,D,W},
    {W,Z,W,W,W,W,D,W,W,W,W,W,D,W,W,D,W,W,W,W,W,D,W,W,W,W,Z,W},
    {W,D,W,W,W,W,D,W,W,W,W,W,D,W,W,D,W,W,W,W,W,D,W,W,W,W,D,W},
    {W,D,D,D,D,D,D,D,D,D,D,D,D,D,D,D,D,D,D,D,D,D,D,D,D,D,W},
    {W,D,W,W,W,W,D,W,W,D,W,W,W,W,W,W,W,D,W,W,D,W,W,W,W,D,W},
    {W,D,W,W,W,W,D,W,W,D,W,W,W,W,W,W,W,D,W,W,D,W,W,W,W,D,W},
    {W,D,D,D,D,D,D,D,W,W,D,D,D,D,W,W,D,D,D,D,W,W,D,D,D,D,D,W},
    {W,W,W,W,W,W,D,W,W,W,W,W,E,W,W,E,W,W,W,W,W,D,W,W,W,W,W,W},
    {W,W,W,W,W,W,D,W,W,W,W,W,E,W,W,E,W,W,W,W,W,D,W,W,W,W,W,W},
    {W,W,W,W,W,W,D,W,W,E,E,E,E,E,E,E,E,E,E,W,W,D,W,W,W,W,W,W},
    {W,W,W,W,W,W,D,W,W,E,W,W,W,W,W,W,W,W,E,W,W,D,W,W,W,W,W,W},
    {W,W,W,W,W,W,D,W,W,E,W,W,W,W,W,W,W,W,E,W,W,D,W,W,W,W,W,W},
    {E,E,E,E,E,E,D,E,E,E,W,W,W,W,W,W,W,W,E,E,E,D,E,E,E,E,E,E},
    {W,W,W,W,W,W,D,W,W,E,W,W,W,W,W,W,W,W,E,W,W,D,W,W,W,W,W,W},
    {W,W,W,W,W,W,D,W,W,E,W,W,W,W,W,W,W,W,E,W,W,D,W,W,W,W,W,W},
    {W,W,W,W,W,W,D,W,W,E,E,E,E,E,E,E,E,E,E,W,W,D,W,W,W,W,W,W},
    {W,W,W,W,W,W,D,W,W,E,W,W,W,W,W,W,W,W,E,W,W,D,W,W,W,W,W,W},
    {W,W,W,W,W,W,D,W,W,E,W,W,W,W,W,W,W,W,E,W,W,D,W,W,W,W,W,W},
    {W,D,D,D,D,D,D,D,D,D,D,D,W,W,D,D,D,D,D,D,D,D,D,D,D,D,W},
    {W,D,W,W,W,W,D,W,W,W,W,W,D,W,W,D,W,W,W,W,W,D,W,W,W,W,D,W},
    {W,D,W,W,W,W,D,W,W,W,W,W,D,W,W,D,W,W,W,W,W,D,W,W,W,W,D,W},
    {W,Z,D,D,W,W,D,D,D,D,D,D,D,E,E,D,D,D,D,D,D,D,W,W,D,D,Z,W},
    {W,W,W,D,W,W,D,W,W,D,W,W,W,W,W,W,W,W,D,W,W,D,W,W,D,W,W,W},
    {W,W,W,D,W,W,D,W,W,D,W,W,W,W,W,W,W,W,D,W,W,D,W,W,D,W,W,W},
    {W,D,D,D,D,D,D,W,W,D,D,D,D,D,W,W,D,D,D,D,D,W,W,D,D,D,D,D,W},
    {W,D,W,W,W,W,W,W,W,W,W,W,D,W,W,D,W,W,W,W,W,W,W,W,W,W,D,W},
    {W,D,W,W,W,W,W,W,W,W,W,W,D,W,W,D,W,W,W,W,W,W,W,W,W,W,D,W},
    {W,D,D,D,D,D,D,D,D,D,D,D,D,D,D,D,D,D,D,D,D,D,D,D,D,D,D,W},
    {W,W,W,W,W,W,W,W,W,W,W,W,W,W,W,W,W,W,W,W,W,W,W,W,W,W,W,W},
    {E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E},
    {E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E,E}
};
```

Représentation visuelle de cette matrice :





- Téléportation des personnages dans le tunnel

Ressources

Nous avons essayé de coller au plus près à la version originale de PacMan par Namco. Pour cela, nous avons utilisé ces ressources.

- Dossier très détaillé sur PacMan :

  https://www.gamedeveloper.com/design/the-pac-man-dossier

- Jeu de PacMan en ligne reproduisant fidèlement le PacMan de Namco (je crois qu'il s'agit d'une émulation du jeu original, mais c'est à vérifier). Par contre il faut créer un compte pour pouvoir jouer.

  https://live.antstream.com/?id=b82bbaf9-472f-47ed-9360-4ae28e7d1816&focusElement=playButton

- Vidéo YouTube de quelqu'un jouant avec l'émulateur ci-dessus :

  https://youtu.be/JTrkd-qJuQI?t=90

  Ça nous a été bien utile pour vérifier rapidement des détails sur le fonctionnement du jeu.