

# Documentation Technique

## 1. Introduction

- **Présentation du projet** : Le projet consiste à développer une version moderne et fonctionnelle du jeu vidéo classique Pac-Man, tout en intégrant les normes actuelles de codage et les bonnes pratiques de développement logiciel. Pac-Man est un jeu d'arcade emblématique où le joueur contrôle un personnage principal, Pac-Man, qui doit collecter tous les points dans un labyrinthe en évitant des ennemis appelés fantômes. Le développement de ce projet s'inscrit dans une démarche de qualité logicielle, en mettant en œuvre des concepts tels que la modularité du code, les tests automatisés, et la gestion de projet collaboratif via GitHub. Une attention particulière sera portée à la lisibilité et à la documentation du code, afin de garantir une facilité de maintenance et d'évolution.
- **Objectif du projet** : L'objectif principal de ce projet est de recréer une version de Pac-Man à la fois ludique et respectueuse des standards modernes de développement. Le projet vise à fournir une expérience de jeu fidèle à l'original, tout en intégrant des fonctionnalités et des outils adaptés aux exigences actuelles. Plus précisément, les objectifs incluent :
  - Développer une version fonctionnelle de Pac-Man avec les éléments de base tels que le labyrinthe, les déplacements de Pac-Man, les fantômes ennemis, et le système de score.
  - Respecter les normes de codage modernes, en veillant à la structuration, la lisibilité et la documentation du code.
  - Assurer la qualité du logiciel en utilisant des tests unitaires, des tests d'intégration, et des revues de code systématiques via GitHub.
  - Mettre en œuvre une interface utilisateur intuitive, incluant des écrans de menu, de jeu, et de fin de partie.
  - Faciliter le travail collaboratif grâce à une gestion rigoureuse des versions, des branches, et des pull-requests sur GitHub.

## 2. Architecture générale

- **Langage et environnement de développement** : Dans le cadre de ce projet, nous avons opté pour le langage de programmation Python, accompagné de la bibliothèque Pygame pour la gestion de l'affichage et des éléments graphiques.
- **Structure du code** : Le projet est séparé en 7 fichiers :
  - Matrice.py

Le fichier Matrice.py contient la matrice nommée MATRICE, qui représente la structure du labyrinthe du jeu. Chaque élément de cette matrice correspond à une case spécifique, indiquant si celle-ci est un mur, un chemin, un point collectable, ou un autre élément. Cette matrice est la base sur laquelle s'appuient les autres modules pour déterminer la configuration du labyrinthe. Elle est essentielle pour gérer les positions des éléments et leurs

déplacements. En outre, elle sert à dessiner graphiquement le labyrinthe dans le jeu et à placer les éléments comme Pac-Man, les fantômes et les points.

- Labyrinthe.py

Le fichier Labyrinthe.py définit la classe Labyrinthe, responsable de la gestion visuelle et fonctionnelle du labyrinthe. Cette classe utilise la matrice définie dans Matrice.py pour construire et afficher le labyrinthe en jeu. Elle s'appuie sur la bibliothèque Pygame pour dessiner les murs et autres éléments visuels en fonction des cases définies. De plus, le module inclut des méthodes pour vérifier les interactions avec les cases du labyrinthe, telles que la gestion des collisions entre Pac-Man ou les fantômes et les murs. Ce fichier permet également de vérifier les murs autour de chaque bloc pour ajuster la navigation des entités.

- Graphe.py

Le fichier Graphe.py propose une structure de graphe correspondant au labyrinthe. Ce graphe est une représentation abstraite qui stocke les connexions possibles entre les différentes cases de la matrice. Il indique les positions accessibles et les directions dans lesquelles Pac-Man et les fantômes peuvent se déplacer. Ce module est essentiel pour calculer les chemins possibles et gérer les déplacements stratégiques des fantômes en fonction de leurs comportements. La navigation dans le labyrinthe est facilitée par ce graphe, rendant possible des algorithmes d'intelligence artificielle pour les fantômes.

- Partie.py

Le fichier Partie.py gère la boucle principale du jeu, orchestrant l'ensemble des modules et événements. Il initialise l'écran de jeu et les objets centraux comme le labyrinthe, Pac-Man, et les fantômes. Ce fichier utilise la bibliothèque Pygame pour gérer les entrées clavier, les interactions en temps réel et les rafraîchissements de l'écran. Il contient également la logique pour redessiner le plateau de jeu, mettre à jour les positions des entités, et contrôler la vitesse du jeu à l'aide d'une horloge. Partie.py est le point d'entrée principal du projet, où tous les autres fichiers collaborent pour former une expérience de jeu cohérente.

- Fantômes.py

Le fichier Fantomes.py définit la classe Fantome, qui hérite de la classe générique Entity. Ce module est consacré à la gestion des fantômes du jeu, y compris leurs comportements distincts, tels que la poursuite, la fuite, ou les déplacements aléatoires. Les fantômes ont également des caractéristiques spécifiques, comme une vitesse et une apparence qui peuvent changer selon les circonstances (par exemple, lorsque Pac-Man consomme un Power Pellet). Fantomes.py utilise Pygame pour charger et afficher les sprites des fantômes, tout en gérant leurs déplacements et leurs interactions avec le labyrinthe et Pac-Man.

- Pacman.py

Le fichier Pacman.py contient la définition de la classe PACMAN, qui hérite également de la classe Entity. Ce module gère tous les aspects du personnage principal du jeu, notamment ses mouvements, son affichage et ses interactions avec le labyrinthe et les fantômes. Pacman.py inclut la logique nécessaire pour collecter les points dans le labyrinthe et activer les bonus, comme les Power Pellets. Les sprites de Pac-Man sont chargés et affichés via

Pygame, offrant une animation fluide lors des déplacements. Ce module est également crucial pour gérer les collisions avec les fantômes, la gestion des vies, et les interactions avec les éléments du labyrinthe.

### 3. Fonctionnalités principales

Le jeu propose des fonctionnalités essentielles permettant une expérience fidèle au jeu original Pac-Man. Le joueur contrôle Pac-Man à l'aide des touches directionnelles du clavier pour collecter des Pac-Dots répartis dans un labyrinthe composé de murs infranchissables. Les fantômes ennemis au comportement varié (chasse, fuite, aléatoire) poursuivent Pac-Man. Des Power Pellets permettent temporairement à Pac-Man d'inverser les rôles et de manger les fantômes. Un système de vie et de score affiche en temps réel les performances du joueur.

### 4. Détails techniques

Les différents classes et leur méthodes:

- Classe Entity :
- Classe Labyrinthe :
- Classe Pacman :

### 5. Optimisation et gestion des performances

- 
- 

### 6. Déploiement et configuration

#### Pré-requis techniques

1. Langage de programmation : Python 3.6 ou supérieur.
2. Dépendances : pygame, une bibliothèque utilisée pour la création de jeux en 2D.

#### Instructions d'installation et d'exécution

1. **Installer Python** : Téléchargez et installez Python depuis le site officiel [python.org](https://python.org).
2. **Installer les dépendances** : Ouvrez un terminal ou une invite de commandes, puis exécutez :

```
pip install pygame
```

3. **Télécharger le code source depuis GitHub** : Dans un terminal, clonez le dépôt GitHub en utilisant la commande suivante :

```
git clone
```

```
https://github.com/Ysope/COURMONTAGNE\_TERRIER\_GRISLIN\_Projet.git .
```

4. **Lancer le jeu** : Accédez au répertoire contenant le jeu dans un terminal, puis exécutez la commande :

```
python Partie.py
```