

CodingWeeks – 2048

Projet n°8 : Pac-Man

Bénéat—Byron Aymeric

Chaine Loris

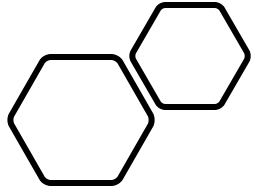
Legrain Basilisse

Mahaut Raphaël

Norodom Thomas

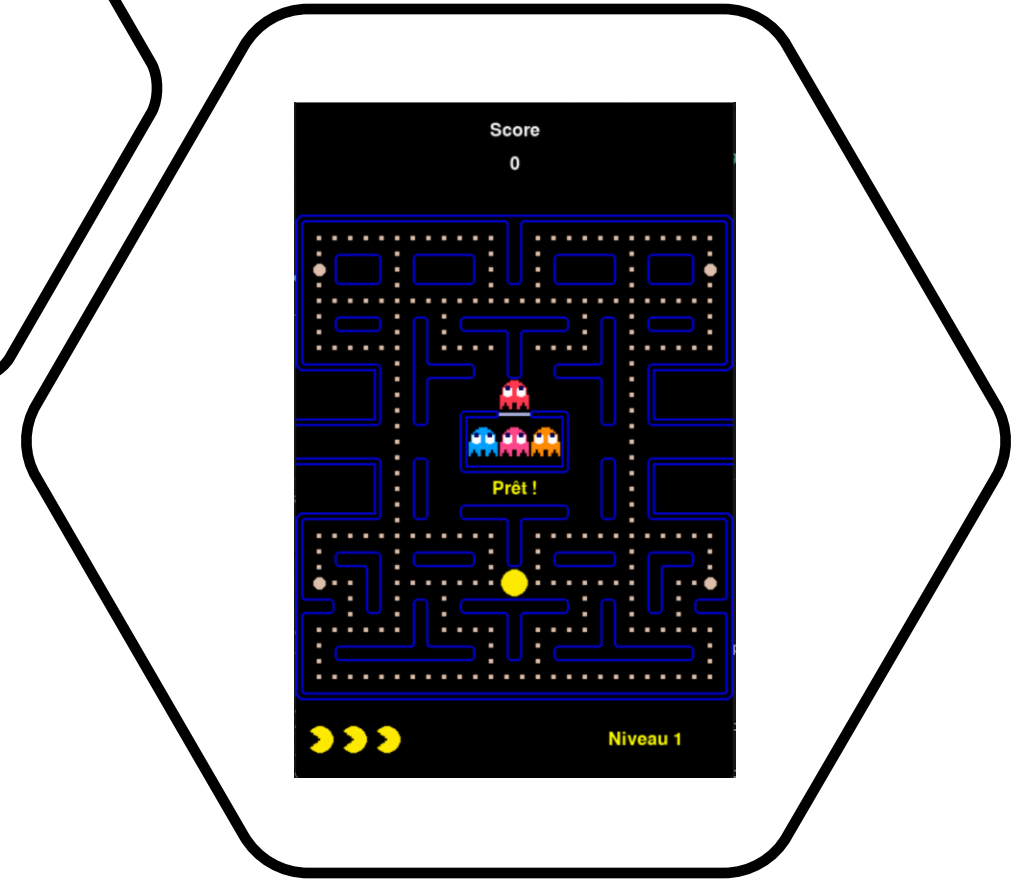
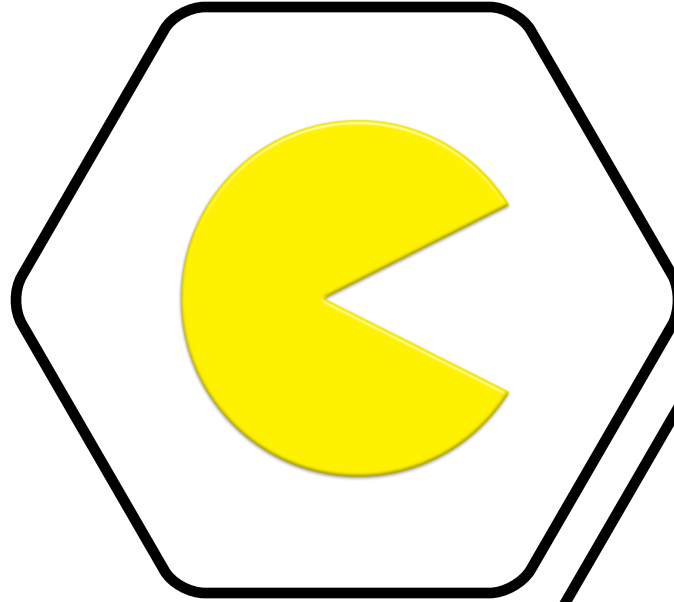
Roze Candice

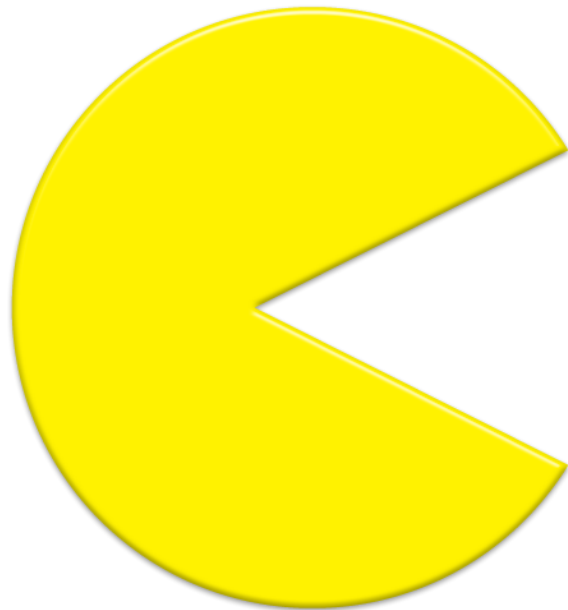
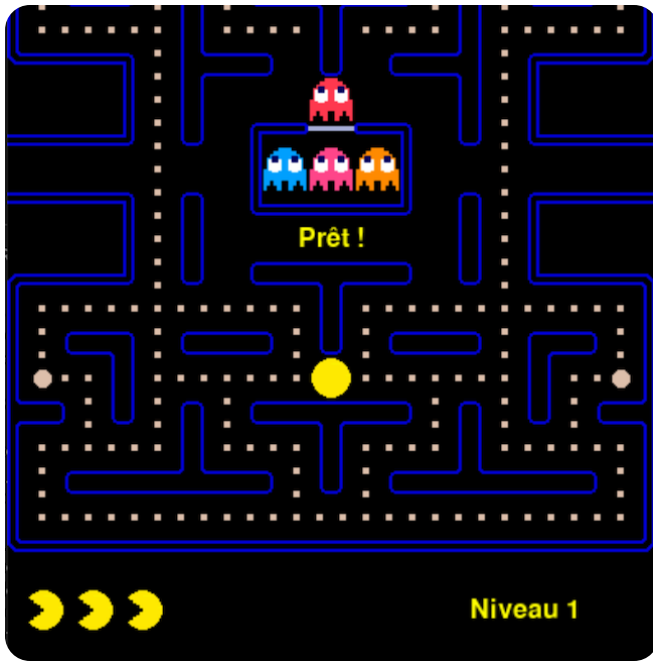
Lien vers le dépôt Gitlab : <https://gitlab-ovh-01.cloud.centralesupelec.fr/thomas.norodom/plant-man>



La motivation du projet :

- jeu culte
- assez complexe dans le comportement du joueur et des fantômes
- pour qui : tous ceux qui veulent s'amuser !





Différentes fonctionnalités retenues :

- Contrôle de Pac-Man avec les flèches du clavier
- Fantômes qui chassent Pac-Man de manière intelligente
- Gommes et fruits spéciaux qui comptent pour le calcul du score
- Boules spéciales qui font que les fantômes ont peur de Pac-Man et ne peuvent temporairement plus l'attaquer
- Trois vies, perte d'une vie à chaque collision avec un fantôme
- Plusieurs niveaux de difficulté croissante
- Victoire à un niveau lorsque toutes les gommes sont attrapées, défaite lorsque aucune vie restante

Découpage des tâches en 3 MVP et 9 fonctionnalités

* 1er MVP :

Fonctionnalité 1 : Créer la grille de jeu

Fonctionnalité 2 : Contrôler Pac-Man

- Étape 1 : Afficher Pac-Man
- Étape 2 : Faire bouger Pac-Man

Fonctionnalité 3 : Gérer les gommes

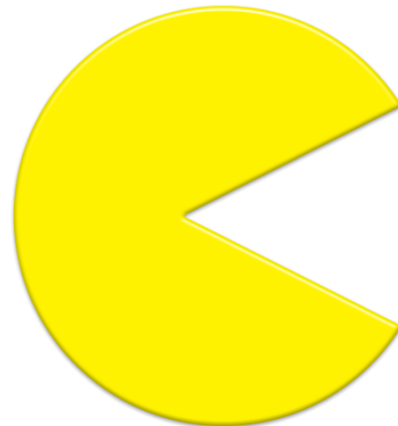
- Étape 1 : Afficher les gommes
- Étape 2 : Gérer les collisions entre Pac-Man et les gommes
- Étape 3 : Fin du jeu (toutes les gommes collectées)

* 2e MVP :

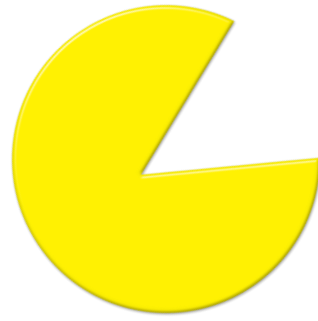
Fonctionnalité 4 : Afficher l'interface

Fonctionnalité 5 : Ajouter les fantômes

- Étape 1 : Afficher les fantômes
- Étape 2 : Faire bouger les fantômes (de manière aléatoire dans un premier temps)
- Étape 3 : Gérer la collision Pac-Man / fantôme



Découpage des tâches en 3 MVP et 9 fonctionnalités



* 3e MVP :

Fonctionnalité 6 : Prise en compte de boules spéciales et changement de l'interaction entre Pac-Man et les fantômes lorsque cette boule est attrapée

Fonctionnalité 7 : Ajout de paramètres de jeu

- Étape 1 : Affichage du score
- Étape 2 : Affichage des vies
- Étape 3 : Affichage des fruits

Fonctionnalité 8 : Paramétrer le mouvement des fantômes, avoir des fantômes intelligents

- Étape 1 : Chaque fantôme a une manière singulière de se déplacer, ils peuvent ainsi se coordonner pour capturer Pac-Man :
 - Blinky, le fantôme rouge, vise la case de Pac-Man (il le poursuit directement)
 - Pinky, le fantôme rose, vise 4 cases devant Pac-Man (cela le fait contourner Pac-Man (il essaie donc de le prendre en tenaille avec Blinky))

- Clyde, le fantôme orange, chasse Pac-Man comme Blinky si ce dernier est loin (plus de 8 cases), sinon il prend une direction aléatoire

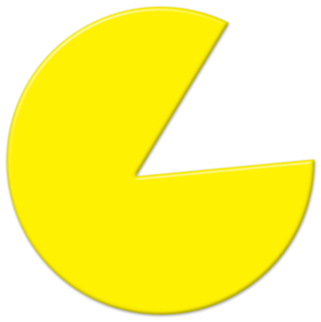
- Inky, le fantôme bleu, vise la case définie par la case de Blinky et le double du vecteur entre Pac-Man et Blinky (il essaie donc d'aller là où Pac-Man fuit logiquement Blinky)

- Étape 2 : le fantôme rouge se met en colère lorsque un certain nombre de gommes est attrapé, il s'accélère

- Étape 3 : Cycles de 30s. Pendant 24-27 secondes, les fantômes chassent Pac-Man. Pendant le temps restant, chaque fantôme part dans un coin de la grille

Fonctionnalité 9 : Gérer les différents niveaux

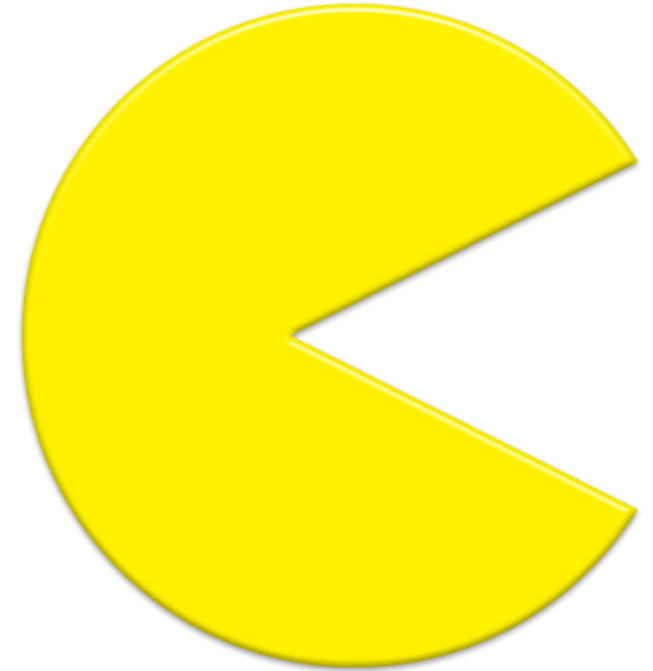
- Étape 1 : Créer les différents niveaux
- Étape 2 : Gérer l'apparition des fruits
- Étape 3 : Modifier la vitesse des fantômes selon le niveau



Répartition des tâches

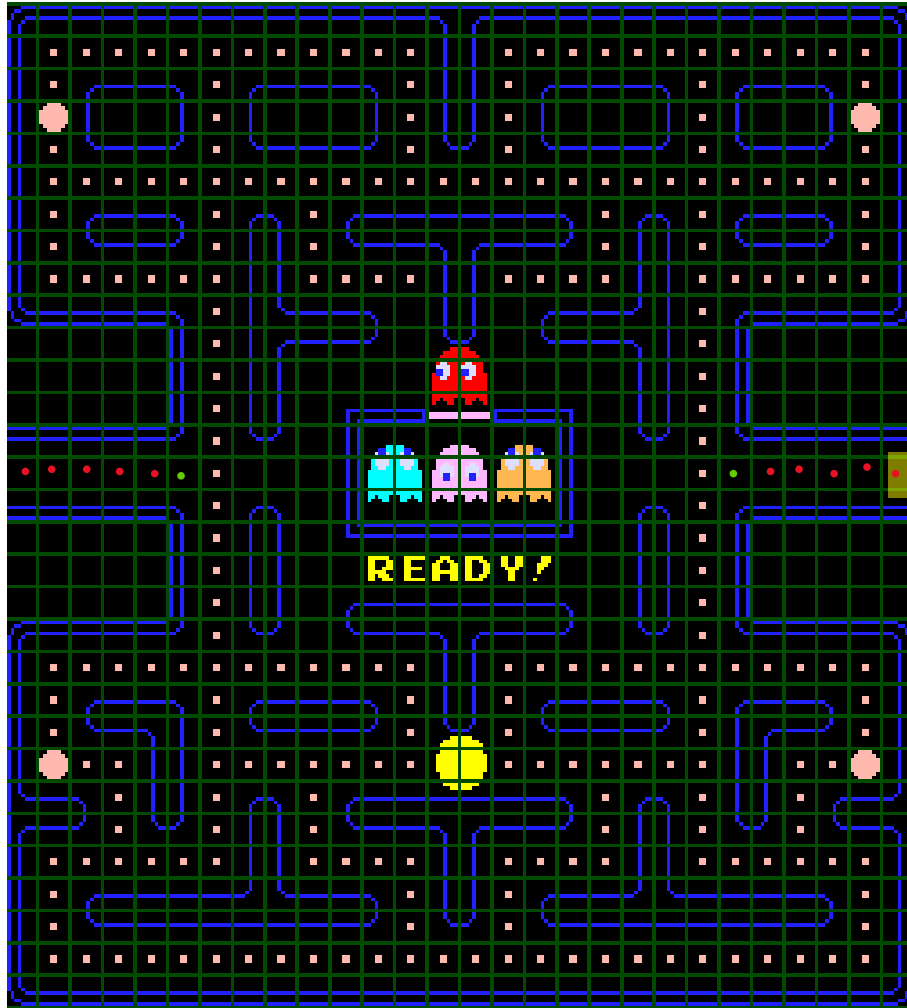
- Bénéat—Byron Aymeric : Comportement des fantômes + grille des mouvements aux intersections
- Chaine Loris : Musique + initialisation + Menu
- Legrain Basilisse : Design des fantômes, de Pac-Man, des murs et des fruits + Game-Over / Victoire + Menu
- Norodom Thomas : Bêta-test + correction de bugs
- Roze Candice (Forum) : Readme + présentation
- Mahaut Raphaël : Main + entités + mouvements joueurs + vitesses

- Grille de jeu
 - Création de la grille et des murs
 - Gestion des Gommages
 - Design des différents éléments
- Gestion des mouvements de Pac-Man
 - Changements de direction au niveau des murs
 - Collision avec les fantômes
- Comportement des fantômes
 - Déplacement en fonction de l'emplacement de Pac-Man
 - Singularité des fantômes



Retour sur le code des principales fonctionnalités

Grille



Clique
texte

[illegible]

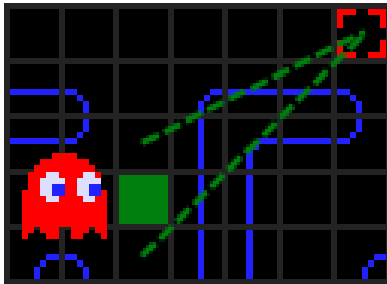
Vitesse

```
[0, 0, 0, 0, 1]
```

```
["mort", "tunnel", "peur", "colere", "normal"]
```

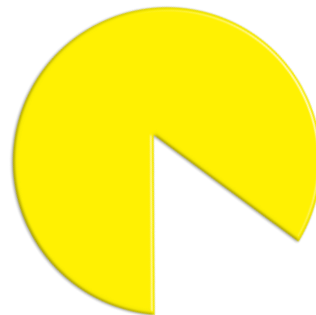
Comportement des fantômes

- Blinky (fantôme rouge):
 - Phase de chasse (entre 24 et 27s) : poursuit directement Pac-Man (vise la case du joueur et emprunte le chemin le plus court)



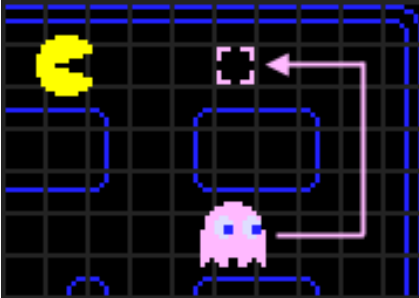
Pour cela, il compare la distance au joueur des cases où il peut aller, puis choisi celle où la distance est la plus courte.

- Phase de 'scatter' (entre 8 et 10s) : retourne au point en haut à droite



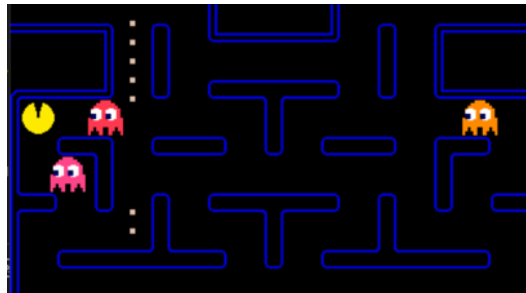
Comportement des fantômes

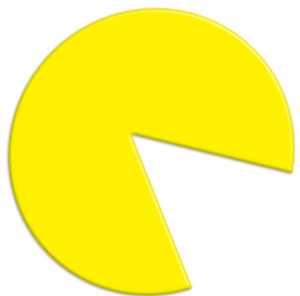
- Pinky (fantôme rose):
 - Phase de chasse (entre 24 et 27s) : essaie de contourner Pac-Man, pour le prendre en tenaille. Pour cela, il vise 4 cases devant Pac-Man, et choisit le chemin le plus court pour y accéder.



Pour cela, il compare la distance au joueur des cases où il peut aller, puis choisi celle où la distance est la plus courte.

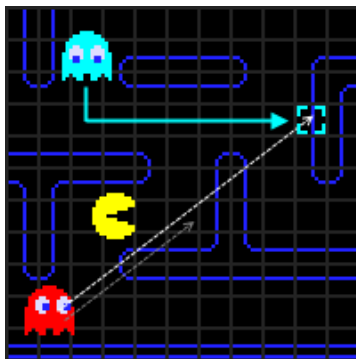
- Phase de 'scatter' (entre 8 et 10s) : retourne au point en bas à droite





Comportement des fantômes

- Inky (fantôme rose):
 - Phase de chasse (entre 24 et 27s) : essaie de prédire la direction de fuite de Pac-Man. Il vise la case définie par la position de Blinky, et 2 fois le vecteur entre Blinky et Pac-Man. Cela a pour effet, de le positionner vers la position de fuite la plus logique.



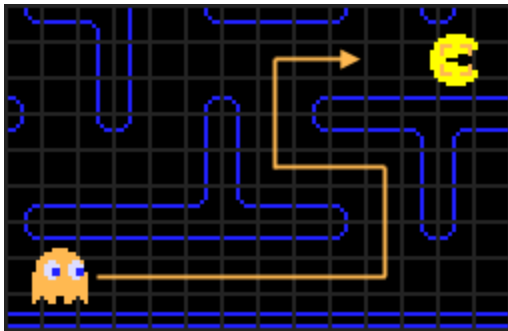
Pour cela, il compare la distance au joueur des cases où il peut aller, puis choisi celle où la distance est la plus courte.

- Phase de 'scatter' (entre 8 et 10s) : retourne au point en haut à gauche



Comportement des fantômes

- Clyde (fantôme orange):
 - Phase de chasse (entre 24 et 27s) : chasse comme Blinky, s'il est assez loin. S'il est trop proche, il prend une direction aléatoire.

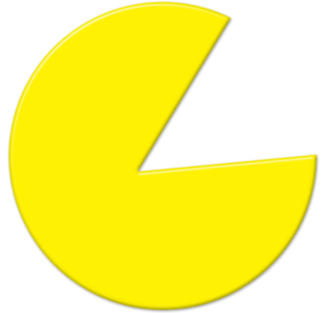
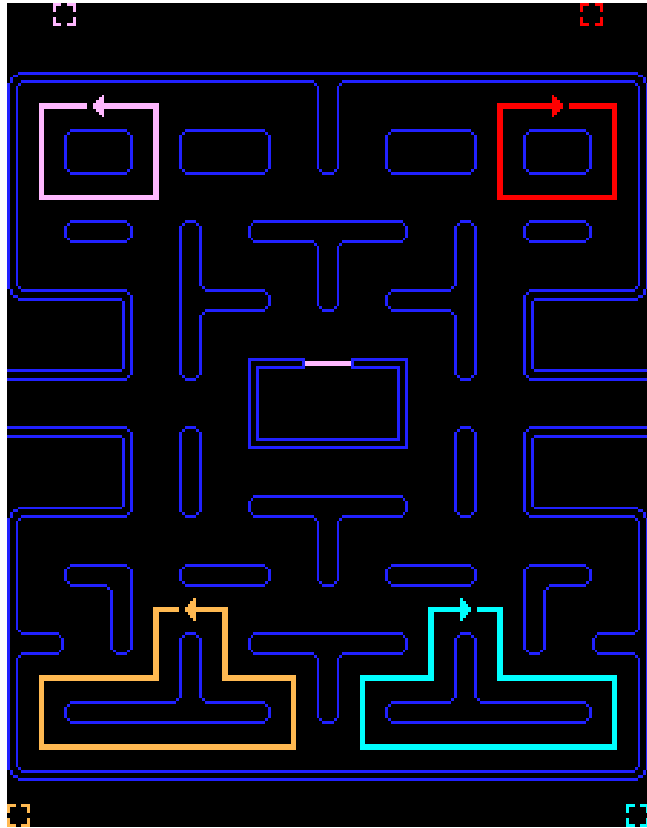


Pour cela, il compare la distance au joueur des cases où il peut aller, puis choisi celle où la distance est la plus courte.

- Phase de 'scatter' (entre 8 et 10s) : retourne au point en haut à gauche

Comportement des fantômes

- Le scatter :



Après les 25-30s de chasse, 8-10 s de scatter.
Selon le même algorithme que Blinky, ils visent leurs cases respectives dans les coins.



Comportement des fantômes

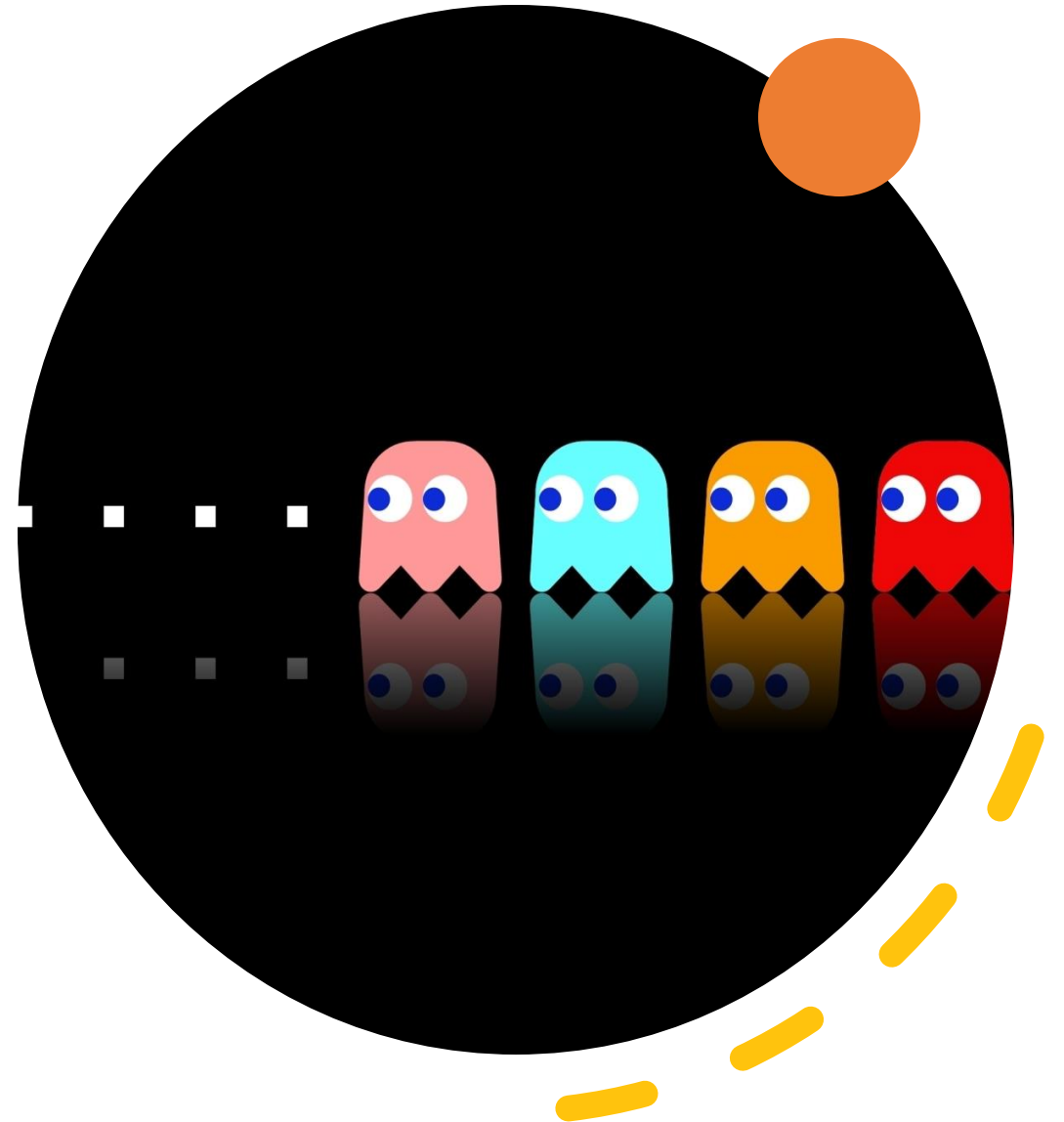
- La peur :



Lorsque Pac-Man prend une super gomme, les fantômes le fuient selon l'algorithme inverse de Blinky.
Au moment où Pac-Man prend la super gomme, ils prennent la direction inverse.

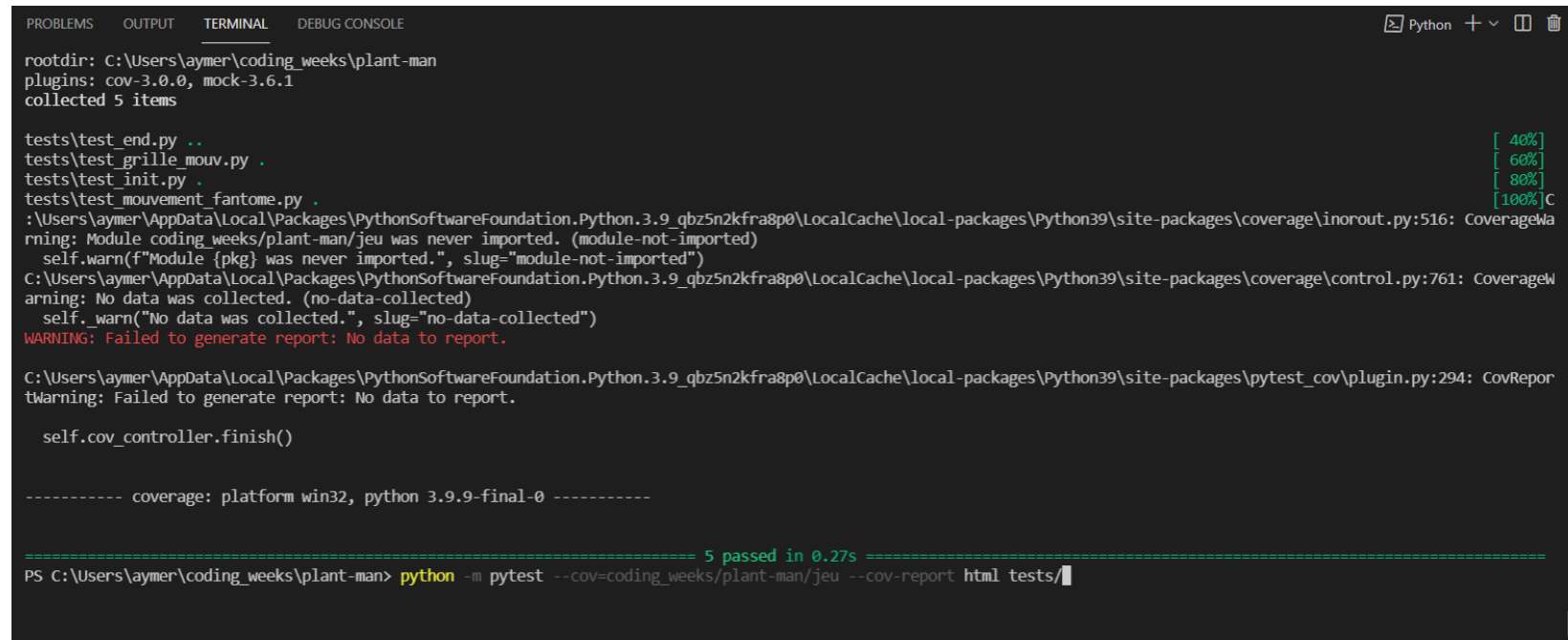
Comportement des fantômes

- La colère :
 - Selon le niveau, à partir d'un certain nombre de Pac-Gomme mangées, les fantômes se mettent en colère. Ils vont plus vite et leur phase de scatter est réduite.



Qualité du code

- Jeu réalise sous pygame, majorité des actions directement liées à l'interface graphique (difficile à tester) => tester à la main et au jeu !
- Tests quand c'est possible, utilisation de mockinputs
- Rapport sous pytest



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
rootdir: C:\Users\aymer\coding_weeks\plant-man
plugins: cov-3.0.0, mock-3.6.1
collected 5 items

tests\test_end.py .. [ 40%]
tests\test_grille_mouv.py . [ 60%]
tests\test_init.py . [ 80%]
tests\test_mouvement_fantome.py . [100%]C

:Users\aymer\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9_qbz5n2kfra8p0\LocalCache\local-packages\Python39\site-packages\coverage\inorout.py:516: CoverageWa
rning: Module coding_weeks/plant-man/jeu was never imported. (module-not-imported)
  self.warn(f"Module {pkg} was never imported.", slug="module-not-imported")
C:\Users\aymer\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9_qbz5n2kfra8p0\LocalCache\local-packages\Python39\site-packages\coverage\control.py:761: CoverageW
arning: No data was collected. (no-data-collected)
  self.warn("No data was collected.", slug="no-data-collected")
WARNING: Failed to generate report: No data to report.

C:\Users\aymer\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9_qbz5n2kfra8p0\LocalCache\local-packages\Python39\site-packages\pytest_cov\plugin.py:294: CovRepor
tWarning: Failed to generate report: No data to report.

  self.cov_controller.finish()

----- coverage: platform win32, python 3.9.9-final-0 -----

===== 5 passed in 0.27s =====
PS C:\Users\aymer\coding_weeks\plant-man> python -m pytest --cov=coding_weeks/plant-man/jeu --cov-report html tests/
```

Qualité du code

- Pour les fonctions testables :

Coverage report: 98%

<i>Module</i>	<i>statements</i>	<i>missing</i>	<i>excluded</i>	<i>coverage</i>
plant-man\end.py	47	3	0	94%
plant-man\grille_mouv.py	98	0	0	100%
plant-man\init.py	25	1	0	96%
plant-man\mouvement_fantome.py	45	0	0	100%
Total	244	6	0	98%

coverage.py v6.1.1, created at 2021-11-12 10:35 +0100

Qualité du code

- Modularité :
 - Il est facile de créer une nouvelle grille, les fonctions se chargent d'automatiser son principe et son fonctionnement
 - Il existe même une fonction créant une grille à partir du dessin
 - Les textures sont changeables facilement
 - L'IA des fantomes peut être modifiée, ainsi que leur nombre
 - La musique peut être remplacée
 - Le seul module à importer est pygame

Qualité du code

- Documentation:

- <https://zestedesavoir.com/tutoriels/pdf/846/pygame-pour-les-zesteurs.pdf>
- <https://www.pygame.org/docs/ref/rect.html>
- <https://www.grospixels.com/site/pacman1.php>
- <https://strategywiki.org/wiki/Pac-Man/Gameplay>
- <https://gitlab-ovh-01.cloud.centralesupelec.fr/thomas.norodom/plant-man/-/blob/e5cd4f1d7a5eb39330f552f67b0764329f10cdab/README.md>

(README)

VIDEO DE PRESENTATION