PROJET SNAKE

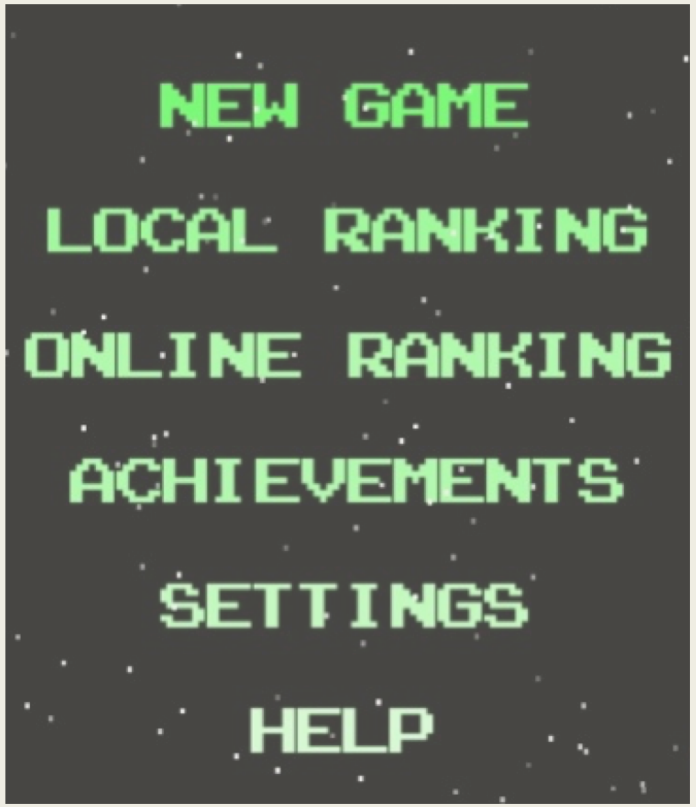
Beya Ben Ayed, Jules Duchesne, Ysé de Reydet de Vulpillieres, Antoine Do Nascimento, Lola Escande, Guillaume Franchino

Accessible par le lien suivant :

git@gitlab-cw1.centralesupelec.fr:beya.benayed/g18_snake.git

SOMMAIRE

- Présentation du produit
- Règles du jeu et fonctionnalités
- Structure du projet
- Construction du projet :
  - Définition des tâches
  - Réalisation
  - Expérimentations et corrections
  - Ajouts
- Bibliographie



LE PRODUIT

- Version du célèbre jeu Snake : un jeu d'arcade puis fer de lance des applications mobiles avec Nokia
- Pour qui ?
  - Public jeune pouvant s'amuser avec ce jeu simple
  - Public nostalgique des jeux rétros
- Des besoins :
  - Divertissement à la maison
  - Faire passer le temps à l'extérieur : transports en commun, attente, etc.

➢ Un produit : jeu vidéo, pour l'instant exécutable en Python

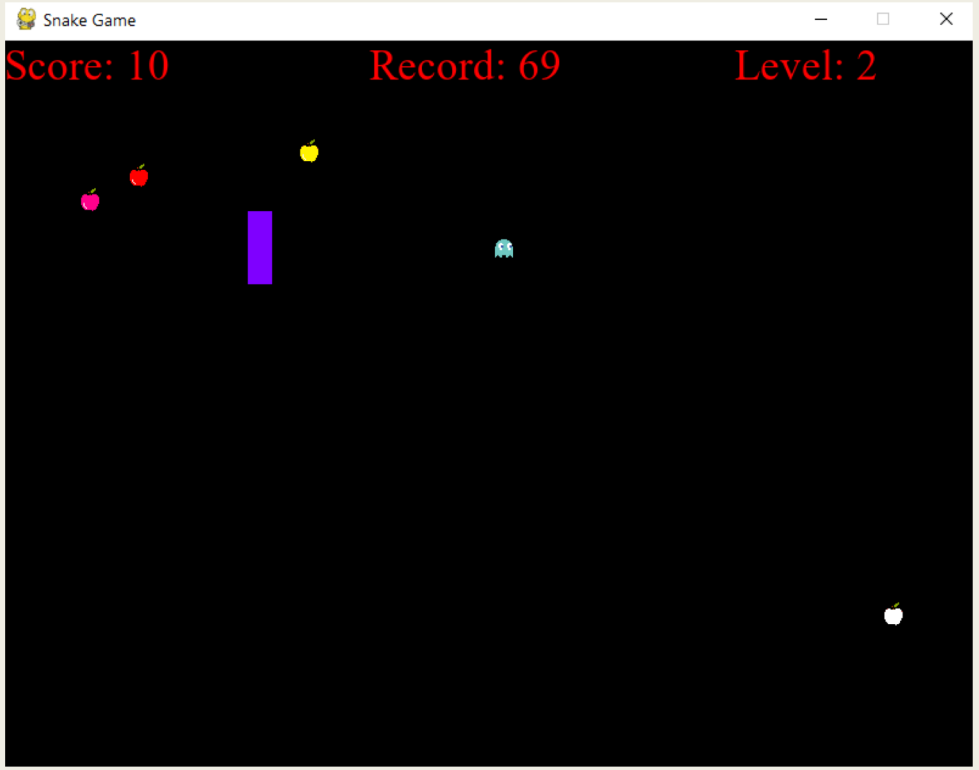
RÈGLES DU JEU

- Le joueur démarre une partie : une fenêtre s'ouvre avec un serpent de taille 3
- But du jeu : récolter des pommes pour marquer le plus de points possible
- Le joueur décide de la direction du serpent
- Fin du jeu : le joueur perd si son serpent rentre dans un mur ou se rentre dedans

LES BONUS

Rendre le jeu plus attractif → Nouveaux bonus aux effets divers

- Pomme rouge : pomme classique
- Pomme verte : coupe le serpent en deux.
- Pomme blanche : accélère la vitesse du serpent pendant 10 secondes, fait gagner 10 points
- Pomme rose : fait gagner 3 points
- Pomme turquoise / fantôme : enlève les bords de la grille pendant 10 secondes
- Pomme jaune : décélère la vitesse du serpent pendant 10 secondes.
- Les niveaux : plus on augmente de niveaux et plus la vitesse du serpent augmente.



Écran de jeu en cours

FONCTIONNALITÉS

- Lancer le jeu
- Diriger le serpent à n'importe quel moment : touches directionnelles
- Mettre sur pause : touche « p »
- Rejouer : touche « c »
- Quitter : touche « q »
- Affichage du score pendant la partie
- Affichage du score final et du meilleur score à la fin de la partie
- Affichage des niveaux

Écran de pause

STRUCTURE DU PROJET

- Un document Word de mise en route du projet
- Un fichier README présentant le projet
- Des fichiers Python pour le code :
  - Un fichier Main pour le programme principal
  - Des fichiers regroupant les différentes fonctions utilisées
    - Affichage : position et affichage du serpent
    - Avance : déplacements et collisions
    - Pommes : génération des pommes et effets associés
  - Un fichier test

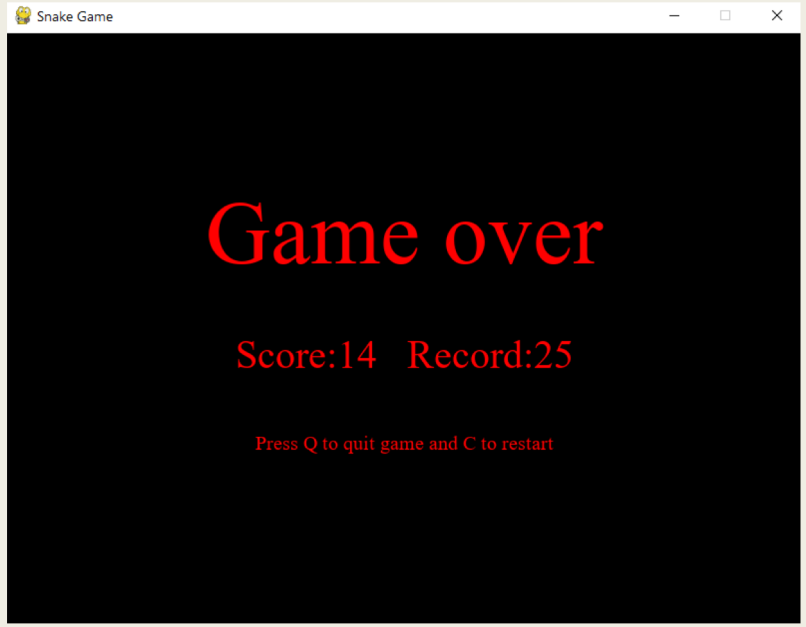
➢ Utilisation du module Pygame et d'autres modules pour le code : random, time, math, copy

DéFINITION DES TâCHES

- Un premier document Word avec les objectifs : MVP puis amélioration du produit
- Répartition du travail sur 3 jours
- Utilisation de Git pour travailler en groupe
- Découverte de nouveaux outils pour le corps du code : Pygame
- Révision des tâches à réaliser
  - Concertation pour définir les fonctionnalités
  - Proposition de nouvelles idées au fur et à mesure
  - Concertation pour définir une interface graphique pour le produit final

RÉALISATION

- Compréhension de Pygame et premières écritures : à faire par tous
- Répartition du travail selon les préférences et compétences des personnes du groupe, par exemple :
  - Beya : première interface, pomme bonus, insertion image
  - Antoine : gestion du bord, quitter le jeu, score et record
  - Ysé : fonctions de mouvement, pommes bonus, refactoring
  - Guillaume : collision avec une pomme bonus, suivi du travail des autres pour la présentation
  - Lola : message de fin de partie, rejouer la partie, refactoring, test
  - Jules : collision avec un fruit malus, accélération et ralentissement



Écran de fin de partie

EXPÉRIMENTATIONS ET CORRECTIONS

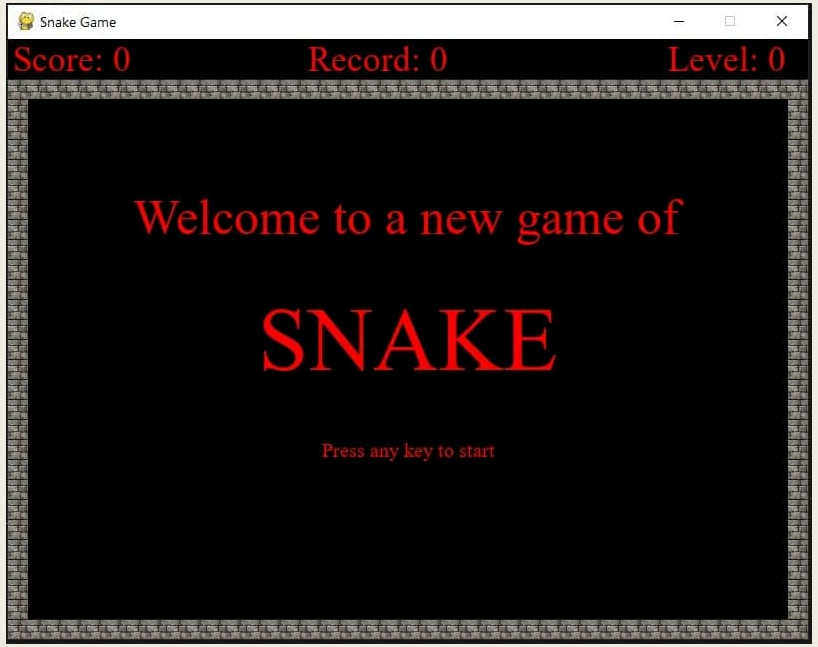
- Des fonctions tests basiques pour parvenir à un code qui marche
- Premier test localement pour chaque nouvelle fonctionnalité
- Le plus souvent, demande d'aide et réflexion avec d'autres membres pour résoudre les problèmes
- Nouveaux tests
- Si concluant, PUSH sur Git et PULL pour les autres membres
- Revue du code par les autres membres avec d'éventuelles correction : exemple de la fonctionnalité pause
- Test à l'aide de Pytest

```
(base) PS C:\Users\Fondation\Desktop\CW\g18_snake> pytest --cov= g18_snake --cov-report html g18_snake/test.py
============================================= test session starts =============================================
platform win32 -- Python 3.8.3, pytest-5.4.3, py-1.9.0, pluggy-0.13.1
rootdir: C:\Users\Fondation\Desktop\CW\g18_snake
plugins: cov-2.10.1, mock-3.3.1
collected 34 items

g18_snake\test.py ................................Coverage.py warning: Module  was never imported. (module-not-imported)
Coverage.py warning: No data was collected. (no-data-collected)
WARNING: Failed to generate report: No data to report.

C:\ProgramData\Anaconda3\lib\site-packages\pytest_cov\plugin.py:271: PytestWarning: Failed to generate report: No data to report.

  self.cov_controller.finish()


============================================= warnings summary ==============================================
C:\ProgramData\Anaconda3\lib\site-packages\win32\lib\pywintypes.py:2
  C:\ProgramData\Anaconda3\lib\site-packages\win32\lib\pywintypes.py:2: DeprecationWarning: the imp module is deprecated in favour o
ntation for alternative uses
    import imp, sys, os

-- Docs: https://docs.pytest.org/en/latest/warnings.html

----------- coverage: platform win32, python 3.8.3-final-0 -----------

========================================== 34 passed, 1 warning in 0.90s ==========================================
(base) PS C:\Users\Fondation\Desktop\CW\g18_snake>
```

Screenshot du passage des différents tests

AJOUTS

- Après avoir un MVP, développer de nouvelles fonctionnalités et améliorer l'expérience utilisateur
- Ajout de différentes pommes bonus → Nouvelles fonctions
- Mettre en forme et commenter le code, *refactoring*
- Ajout d'un écran de fin avec le meilleur score et d'un écran d'accueil
- Développer une interface plus plaisante :
  - Sélection de nouveaux motifs
  - Implémentation dans le jeu



Écran d'accueil du jeu

DIFFICULTÉS ET ENSEIGNEMENTS

- Des fonctionnalités non prévues à régler dans le code : le serpent pouvait revenir sur lui-même
- Des difficultés à appréhender les nouveaux outils : affichage des pommes dans l'interface, accélération et ralentissement qui ne fonctionnent pas avec des clocks, etc.
- Des problèmes dans le code : progression en tâtonnant jusqu'à ce que le code fonctionne
- Mais des enseignements :
  - Préparation et adaptations du projet nécessaires
  - Communication primordiale pour le travail de groupe
  - Méthode de développement MVP efficace pour avoir un premier produit rapidement

BIBLIOGRAPHIE

- Tuto Pygame : https://www.youtube.com/watch?v=gx4yVcJqBaI
- Installation de pip : https://www.youtube.com/watch?v=UTUlp6L2zkw
- Résolutions des problèmes : Stack overflow
- Tuto timer sur Pygame : https://www.youtube.com/watch?v=gx4yVcJqBaI
- Syntaxe pour insérer des images : https://www.geeksforgeeks.org/python-display-images-with-pygame/?fbclid=IwAR0kAYYFXDj8Rp6EaYpx3nCkJr1LsCwuD1QUj1yAsGpu9NjfeMgJMS0bbQ4

MERCI DE VOTRE ÉCOUTE !