



CentraleSupélec

CODING WEEKS

WEEK 2: JEUX EN PYTHON

FLAPPY BRIRD



Pinheiro Terashima, Rafael Seigi
Souto Rodriguez, Thales Augusto
Candido Athanasio, Victor Alexandre
Tokumoto, Yuichi

<https://gitlab-cw1.centralesupelec.fr/thales.sr/flappy-brirds>

SOMMAIRE EXECUTIVE

Introduction

Planification du Projet

Fonctionnalités & Structuration

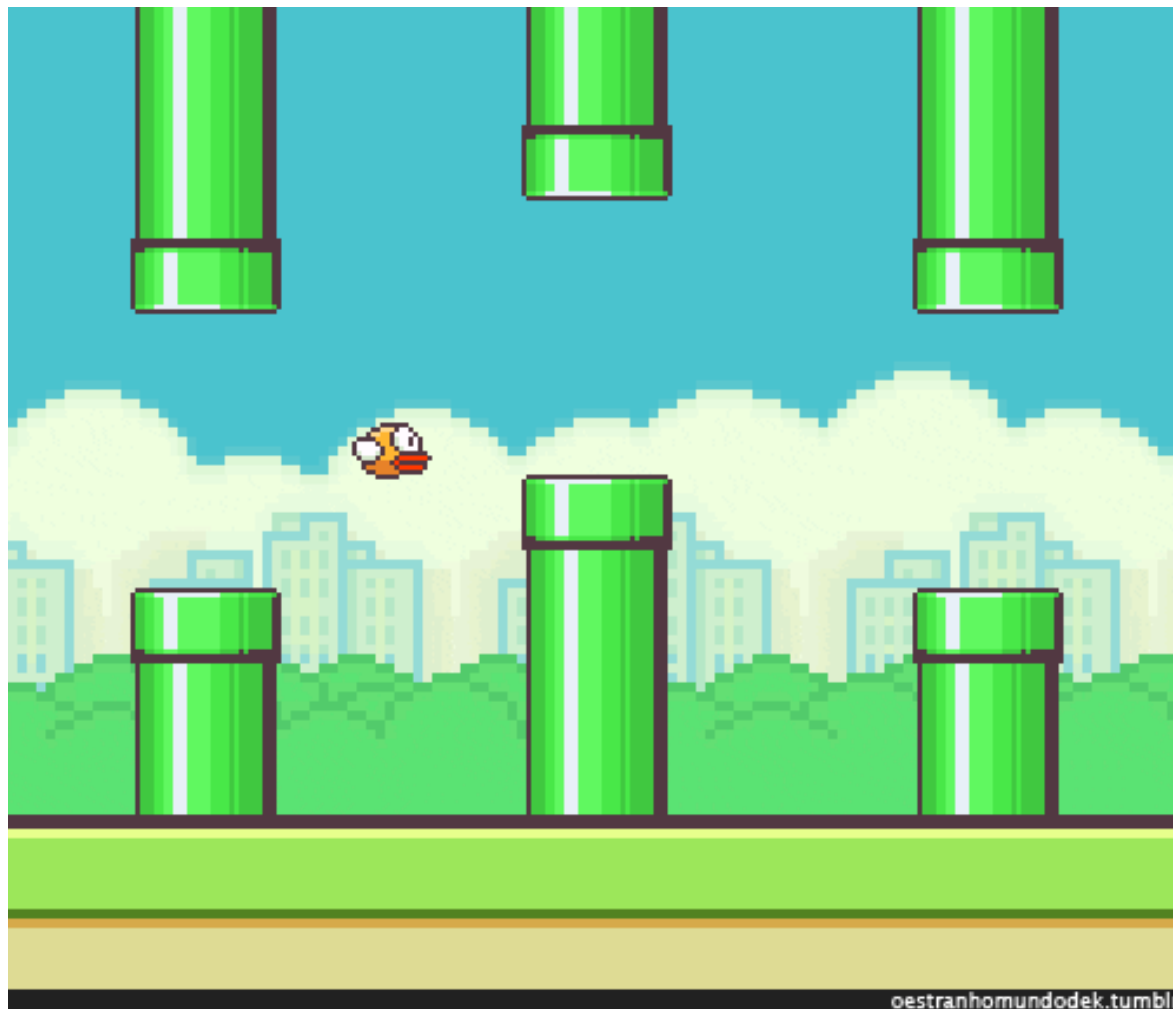
Pytesting & Coverage



INTRODUCTION



FLAPPY BIRD – L'OISEAU QUI SAUTE



DESCRIPTION GLOBAL DU PRODUIT

Quoi ?

- Un jeu pareil à le célèbre jeu mobile de Flappy Bird
- Un jeu développé en Python

Pourquoi ?

- Distraindre pendant le Lockdown
- S'amuser
- Avoir de la nostalgie

Pour Qui ?

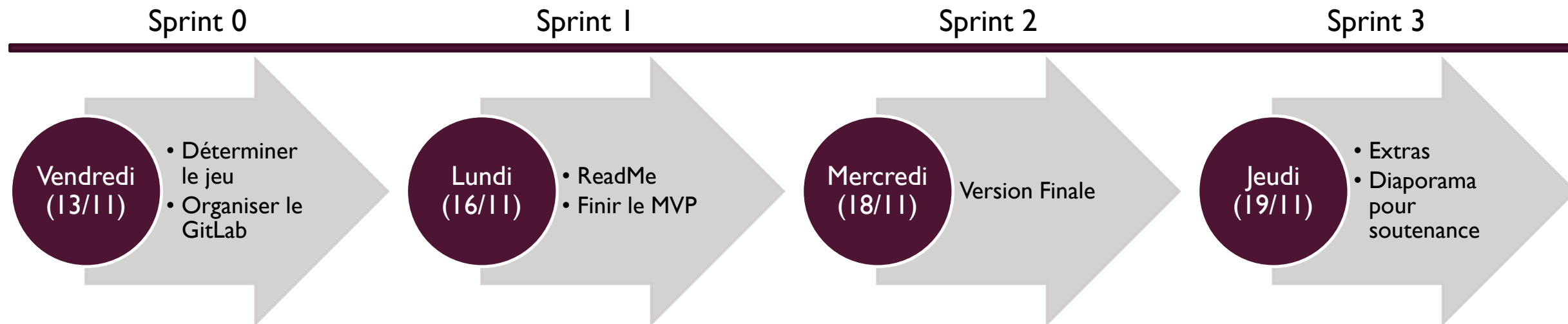
- Pour les personnes qui ont déjà joué, et vont avoir de la nostalgie
- Des personnes intéressés en jouer le Flappy Bird avec des changements qui le rendent plus intéressant

SPRINT 0

- Brainstorming (Tout le monde)
 - Définition du MVP
 - Définition de la Version Finale
 - Définition des fonctionnalités extras
- Définition d'un calendrier (Victor)
- Mettre en œuvre du GitLab (Thales)
- Structuration du projet (Tout le monde)
 - Objectés nécessaires
 - Fonctions nécessaires
 - Structure

PLANIFICATION





FONCTIONNALITÉS

MVP

1) Créer l'environnement

- Créer l'esquelette du jeu

2) Bird

- Implementer la classe Bird

3) Ground

- Implementer la classe Ground

4) Pipe

- Implementer la classe Pipe et la classe Pipes

VERSION FINALE

5) Design

- Creation du design pour le jeu

6) Animations

- Creation de las animations du oiseau e du sol

7) Sons

- Ajouter les sons du jeu

8) Score

- Ajouter le score
- Garder le meilleur score

9) Menu

- Ajouter le menu du jeu a là dynamique

EXTRAS

I0) Auto Play

- Création d'AI pour jouer

I1) Ajouter de publicité

- Création de publicités

I2) Différentes niveaux de difficulté

- Ajouter des tubes que se tubes mobiles
- Ajouter des tubes avec lacune de taille dynamique
- Vitesse variable des tubes

I3) Thèmes

- Ajouter des autres thèmes

I4) Fin du jeu

- Implementer le menu de fin du jeu

GITLAB – FLAPPY BRIRD

■ <https://gitlab-cwl.centralesupelec.fr/thales.sr/flappy-brirds>

The screenshot shows the GitLab web interface for a project named 'Flappy-BRirds'. The left sidebar contains navigation links: Project overview, Details (selected), Activity, Releases, Repository, Issues (0), Merge Requests (0), CI / CD, Operations, Packages & Registries, Analytics, Wiki, Snippets, Members, and Settings. The main content area displays the project details for 'Flappy-BRirds' (Project ID: 112). It shows 2 commits, 1 branch, 0 tags, 174 KB files, and 174 KB storage. The 'master' branch is selected, showing the 'flappy-brirds' directory. Below this, a table lists the files in the repository:

Name	Last commit	Last update
classes	First commit	22 minutes ago
.gitignore	First commit 2	21 minutes ago
README.md	First commit	22 minutes ago
main.py	First commit 2	21 minutes ago

Below the table, the 'README.md' content is displayed, showing the project name 'Project Flappy Bird' and the commit message 'First commit'.

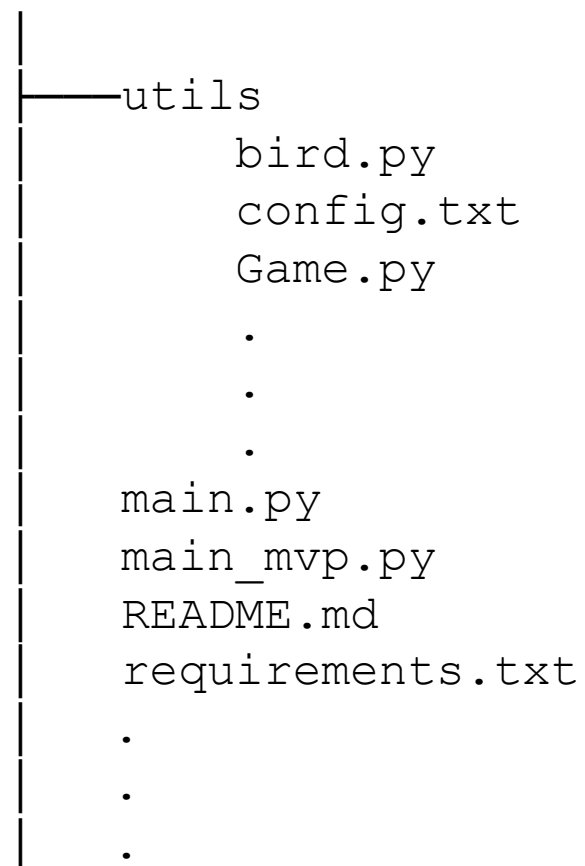
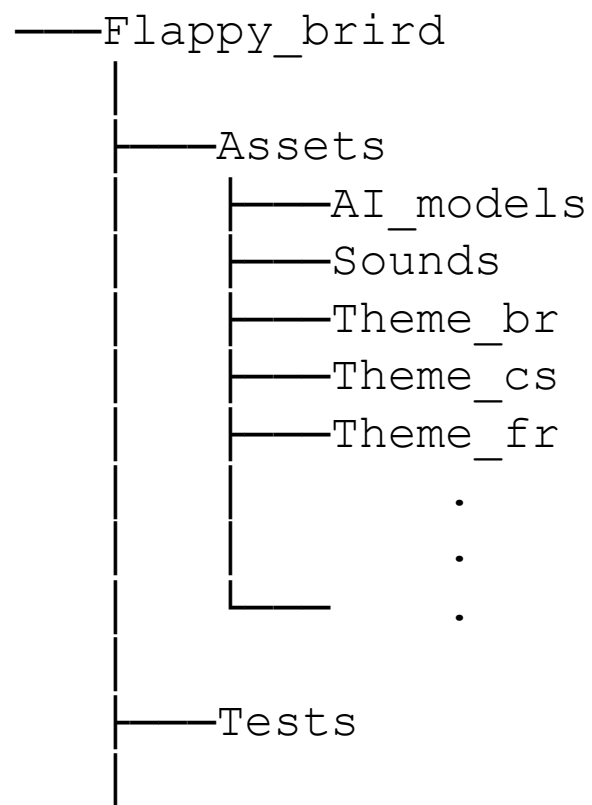
STRUCTURATION



PACKAGES

- Essentiel
 - Pygame
 - pygame-menu
 - neat-python
- Testing
 - Pytest
 - Pytest-cov
 - Coverage

STRUCTURE DU PROJET



OBJETS

```

├── Bird
│   ├── init
│   ├── move
│   ├── .
│   ├── .
│   └── draw
├── Score
│   ├── init
│   ├── count
│   ├── save_score
│   └── draw
├── gamesettings.py
│   ├── init
│   ├── set_difficulty
│   ├── .
│   ├── .
│   └── set_speed

```

```

├── Pipe
│   ├── init
│   ├── move
│   ├── create_pipe
│   ├── .
│   ├── .
│   └── draw
├── Ground
│   ├── init
│   ├── move
│   ├── collide
│   └── draw

```

```

├── radio
│   ├── init
│   ├── die_sound
│   ├── .
│   ├── .
│   └── wing_sound
├── Score_display
│   ├── init
│   ├── restart_button
│   ├── .
│   ├── .
│   └── draw

```



FONCTIONS

```
└─ NEAT
   ├── draw_AI_info
   ├── get_AI_input
   ├── run_ai
   ├── .
   ├── .
   └── ai_gameloop
```

```
└─ Game
   └─ GameLoop
```



SPRINT I

- Fonctionnalité 1 – Créer l'environnement (Thales)
 - Implémenter un main.py et un squelette de code pour aider à la division des tâches et des tests.
- Fonctionnalité 2 – Bird (Yuichi et Thales)
 - Implémenter et tester la classe Bird.
- Fonctionnalité 3 – Ground (Rafael)
 - Implémenter et tester la classe Ground.
- Fonctionnalité 4 – Pipe (Victor)
 - Implémenter et tester la classe Pipe et Pipes.

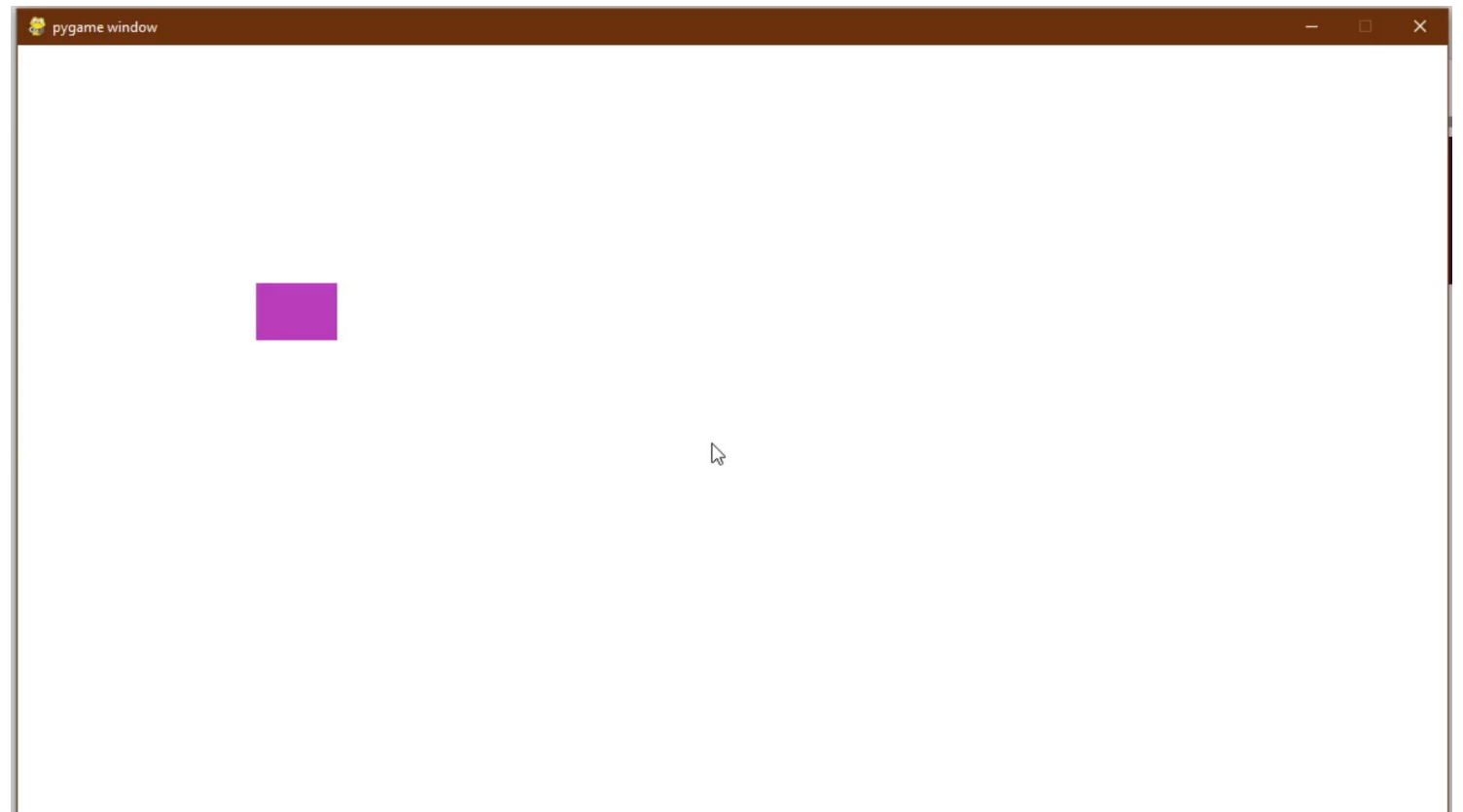
FONCTIONNALITÉ I



Fonctionnalité 1: Implémenter un fichier avec le squelette du code pour faciliter l'implémentation des objets

FONCTIONNALITÉ 2

Fonctionnalité 2: Implémenter la classe bird en considérant l'analyse de besoin



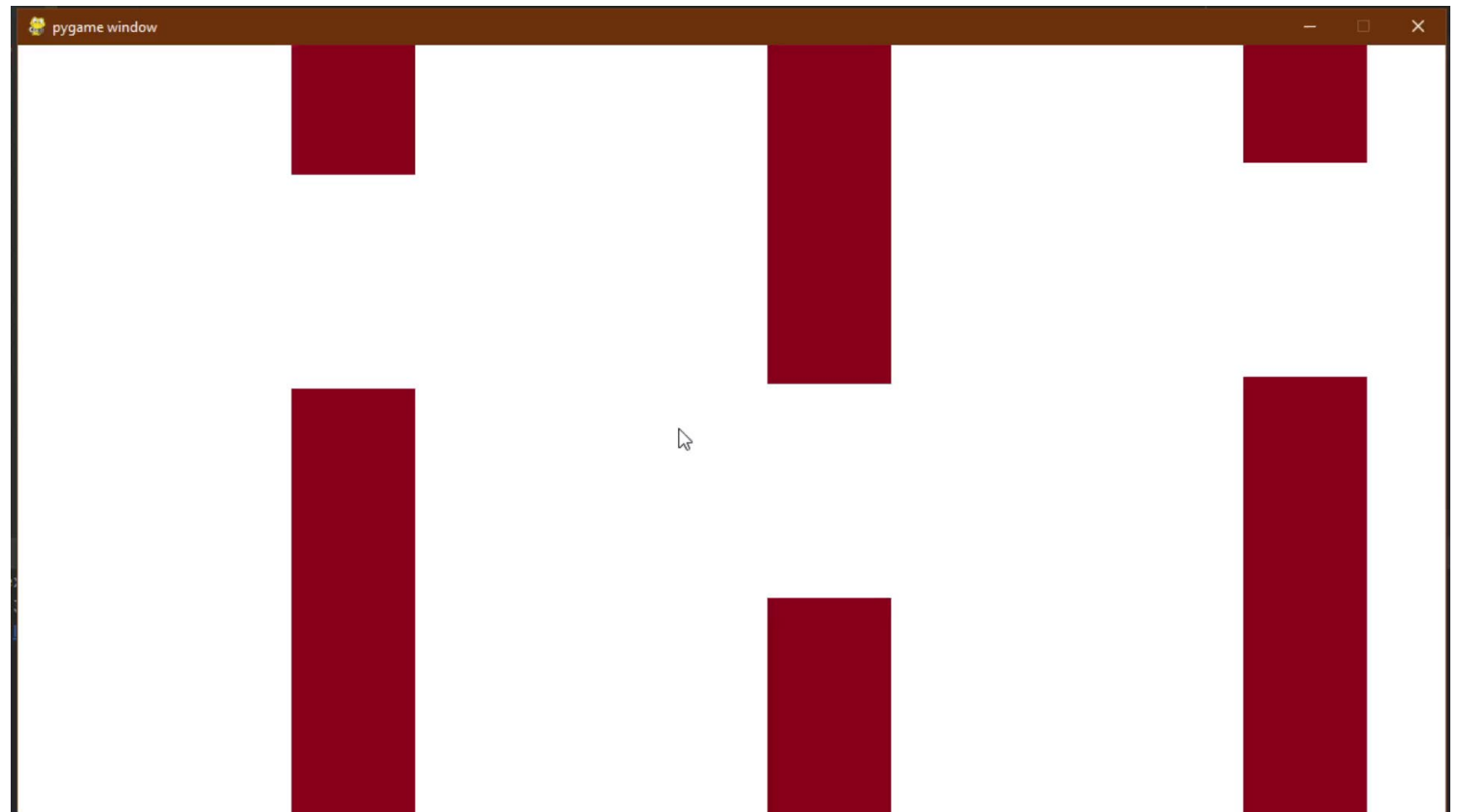
FONCTIONNALITÉ 3



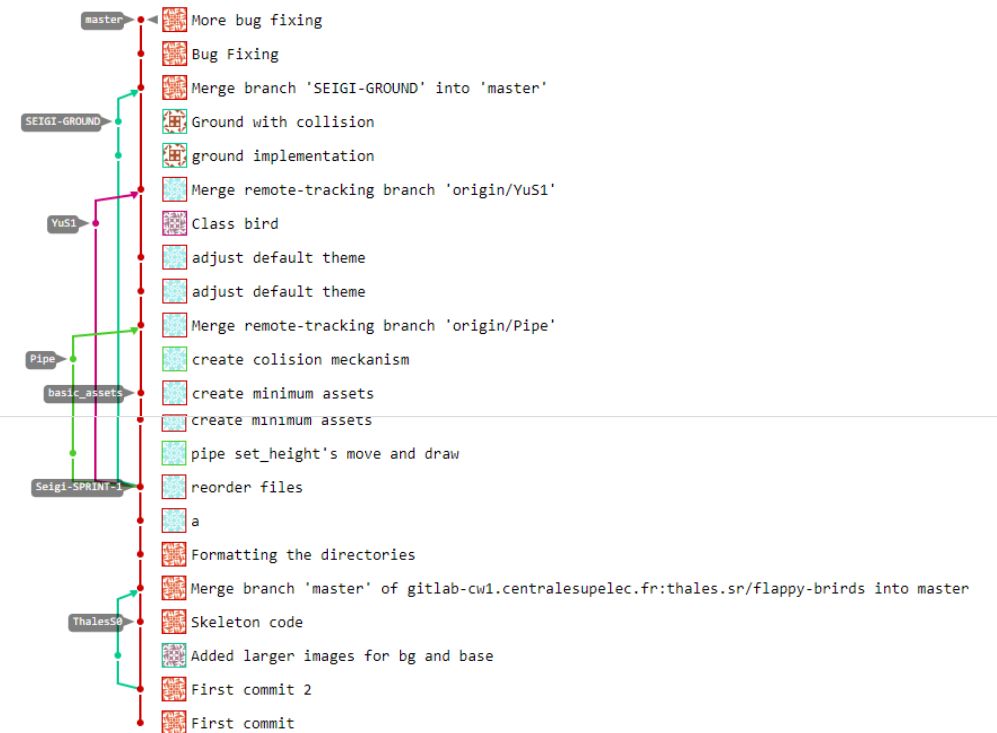
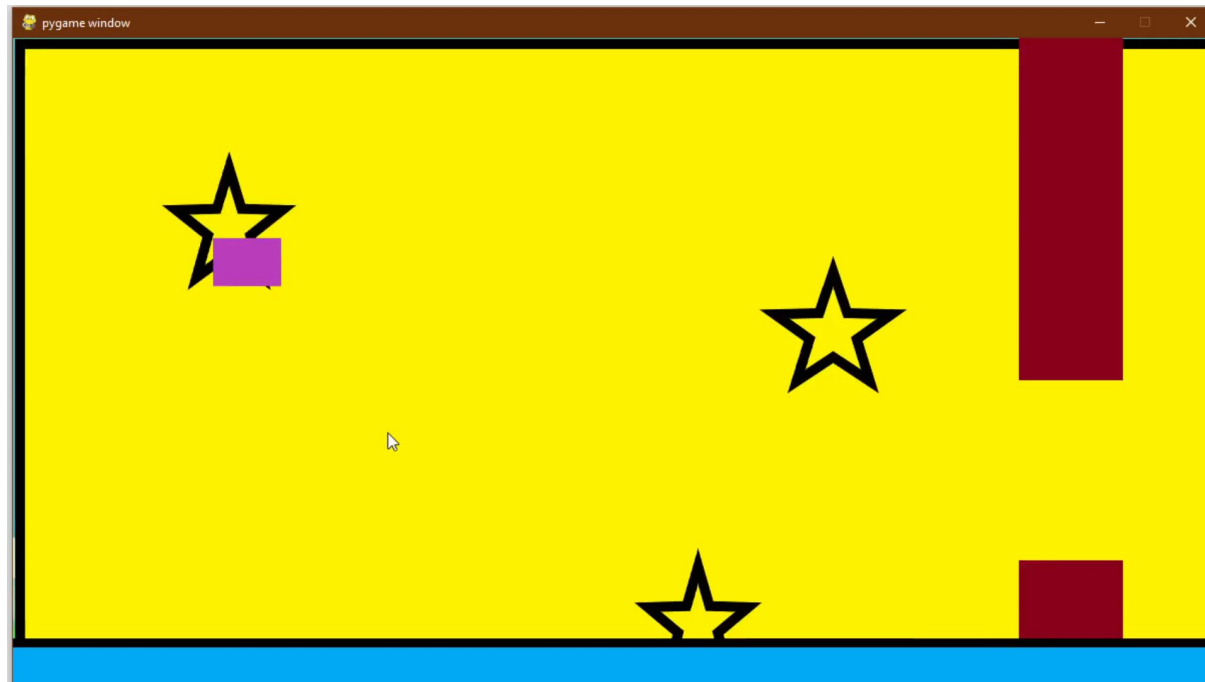
Fonctionnalité 3: Implémenter la classe ground en considérant l'analyse de besoin

FONCTIONNNALITÉ 4

Fonctionnalité 4: Implémenter la classe pipe en considérant l'analyse de besoin



SPRINT I

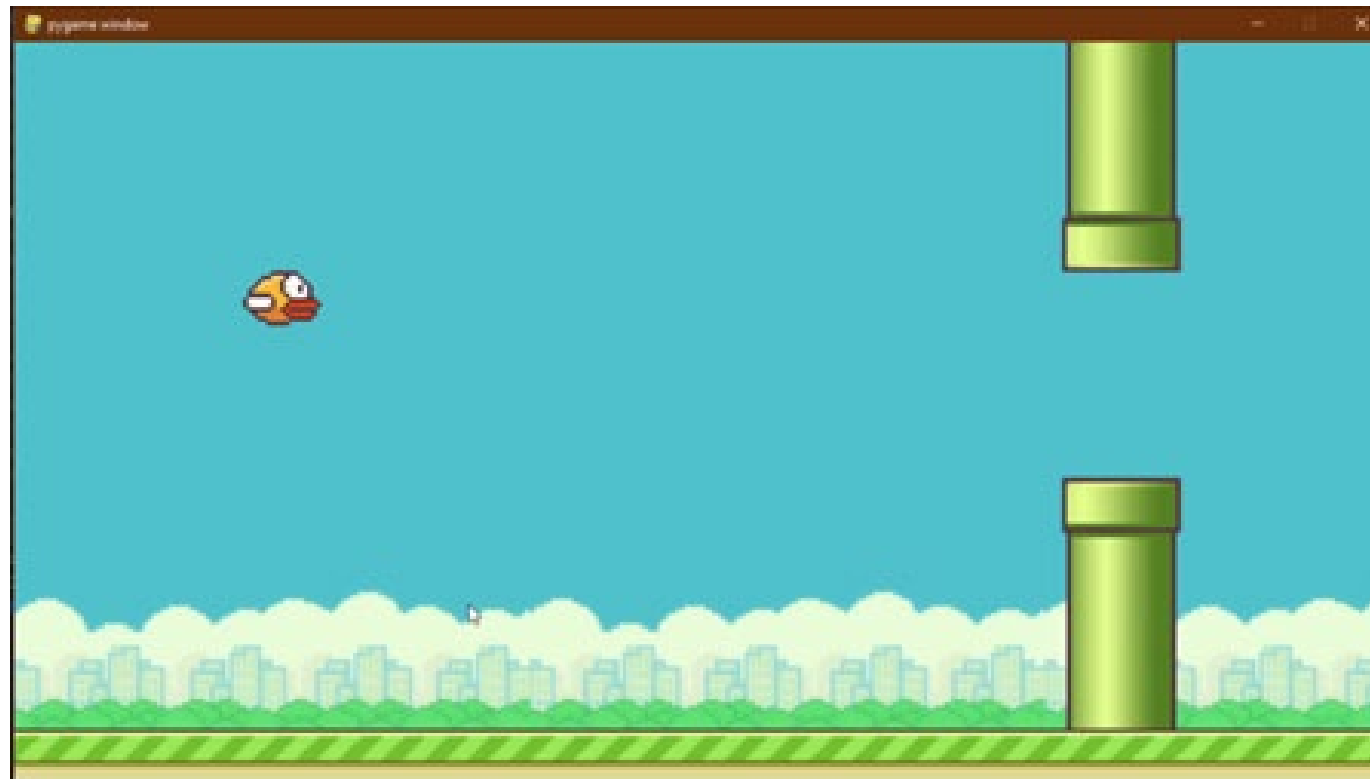


SPRINT 2

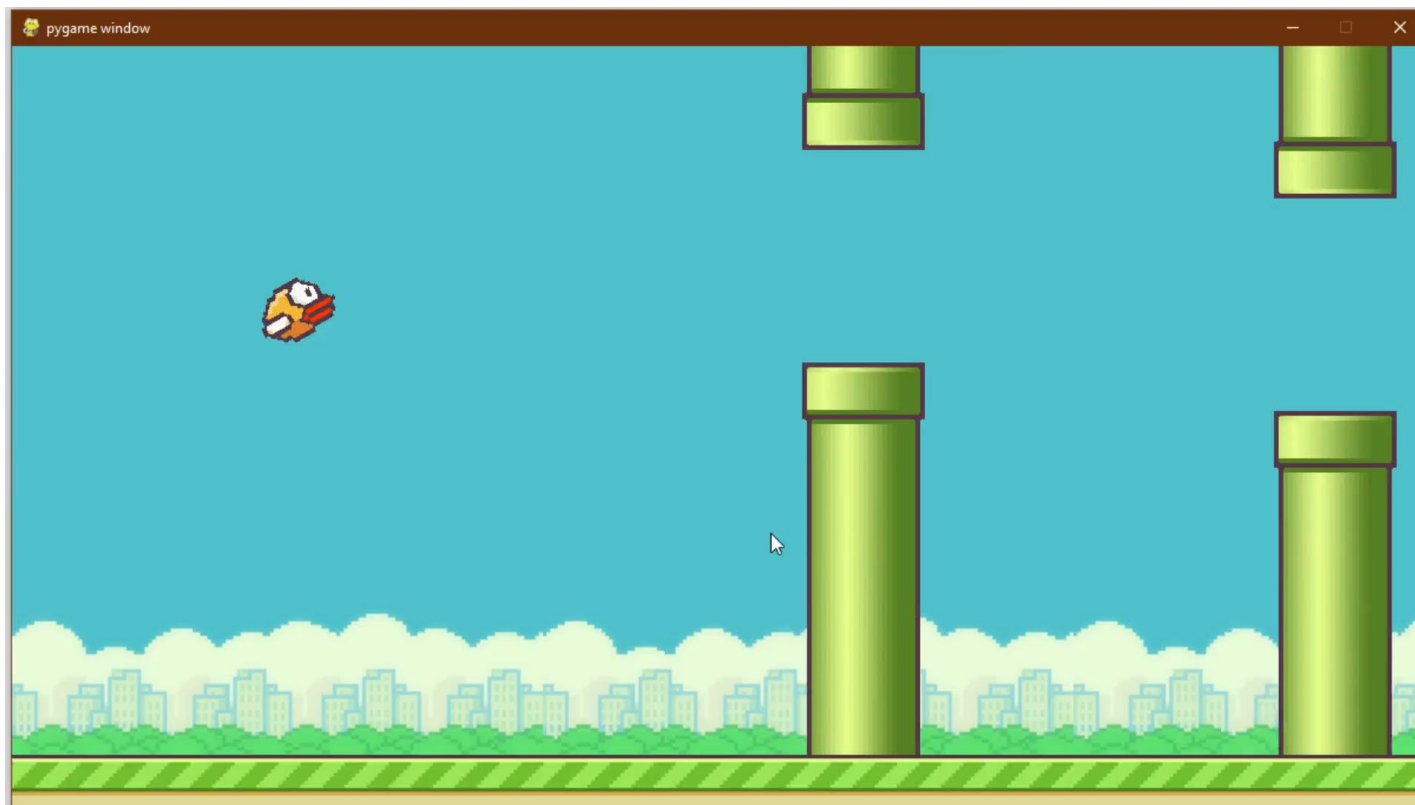
- Fonctionnalité 5 – Obtention et utilisation des images réel (Victor)
 - Mettre en œuvre un jeu avec ces outils obtenu
- Fonctionnalité 6 – Créer les animations d'oiseau (Rafael et Thales)
 - Ajouter a la définition Draw d'oiseau les paramètres nécessaires pour une propre animation
- Fonctionnalité 7 – Ajouter des sons (Thales)
 - Le son d'oiseau
 - Le son quand on perd le jeu
 - Le son pour démarrer le jeu
- Fonctionnalité 8 – Score (Yuichi)
 - Ajouter le score
 - Garder le meilleur score
- Fonctionnalité 9 – Menu (Victor)
 - Ajouter le menu du jeu a là dynamique

FONCTIONNALITÉ 5

Fonctionnalité 5: Implémenter le design original du jeu



FONCTIONNALITÉ 6



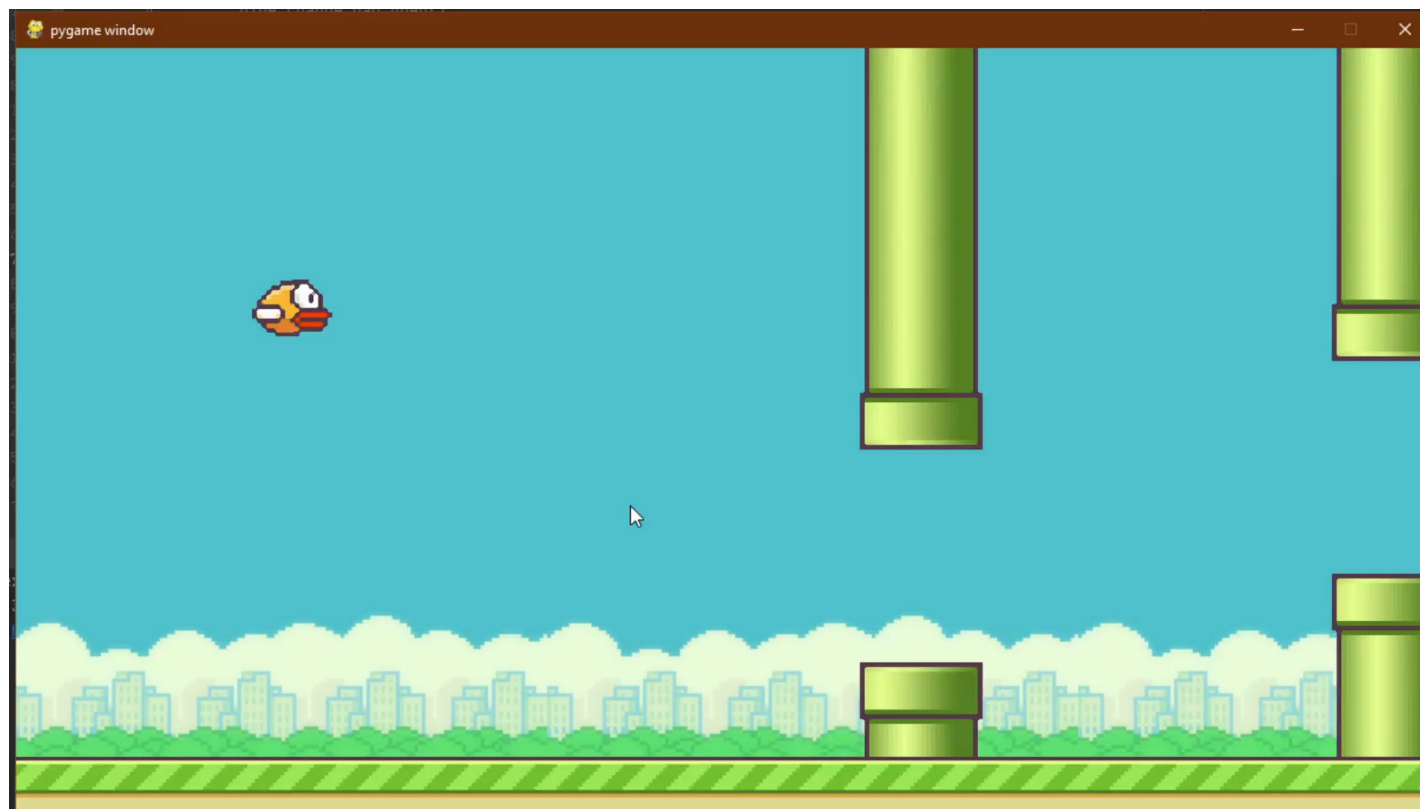
Fonctionnalité 6: Créer les animations du oiseau comme le jeu original



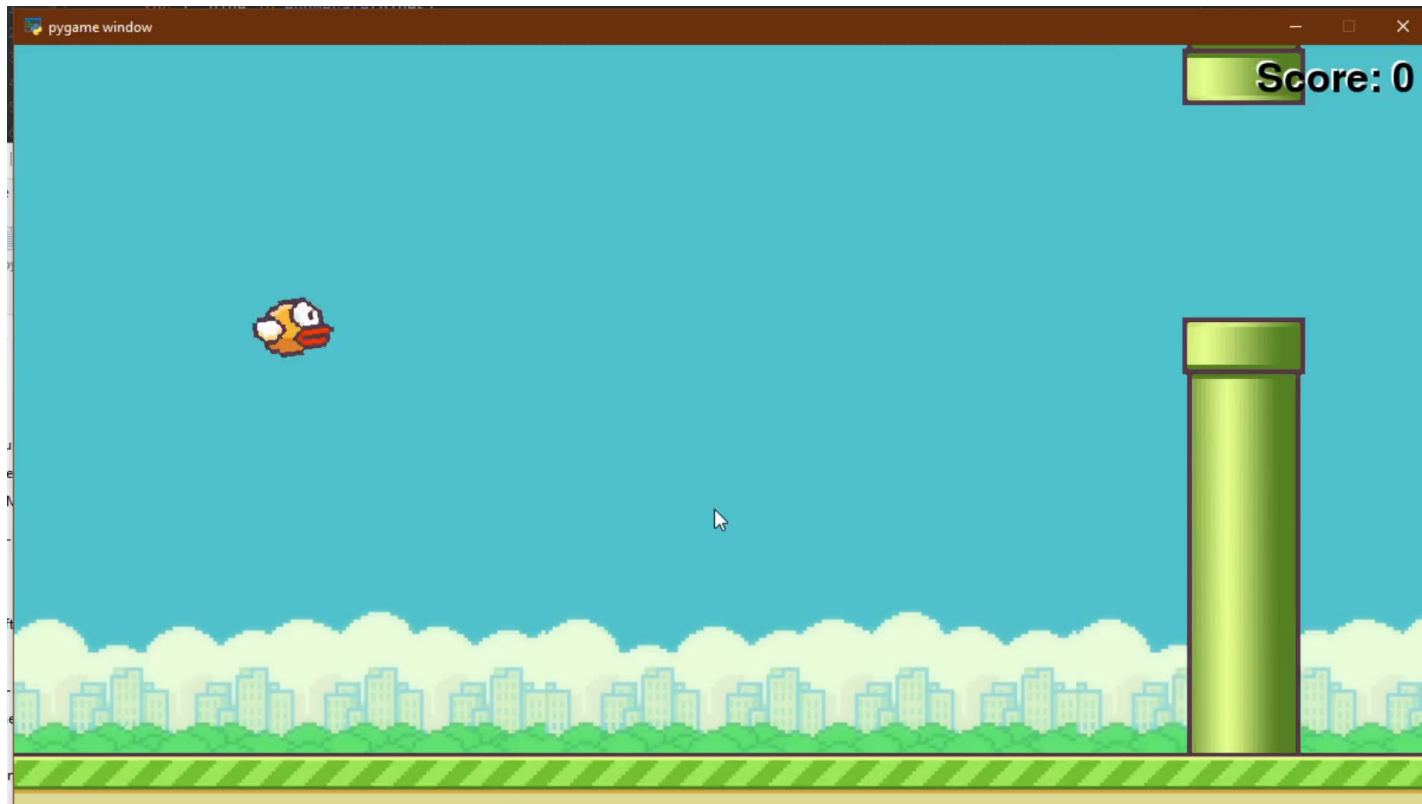
CentraleSupélec

FONCTIONNALITÉ 7

Fonctionnalité 7: Ajouter des sons



FONCTIONNALITÉ 8



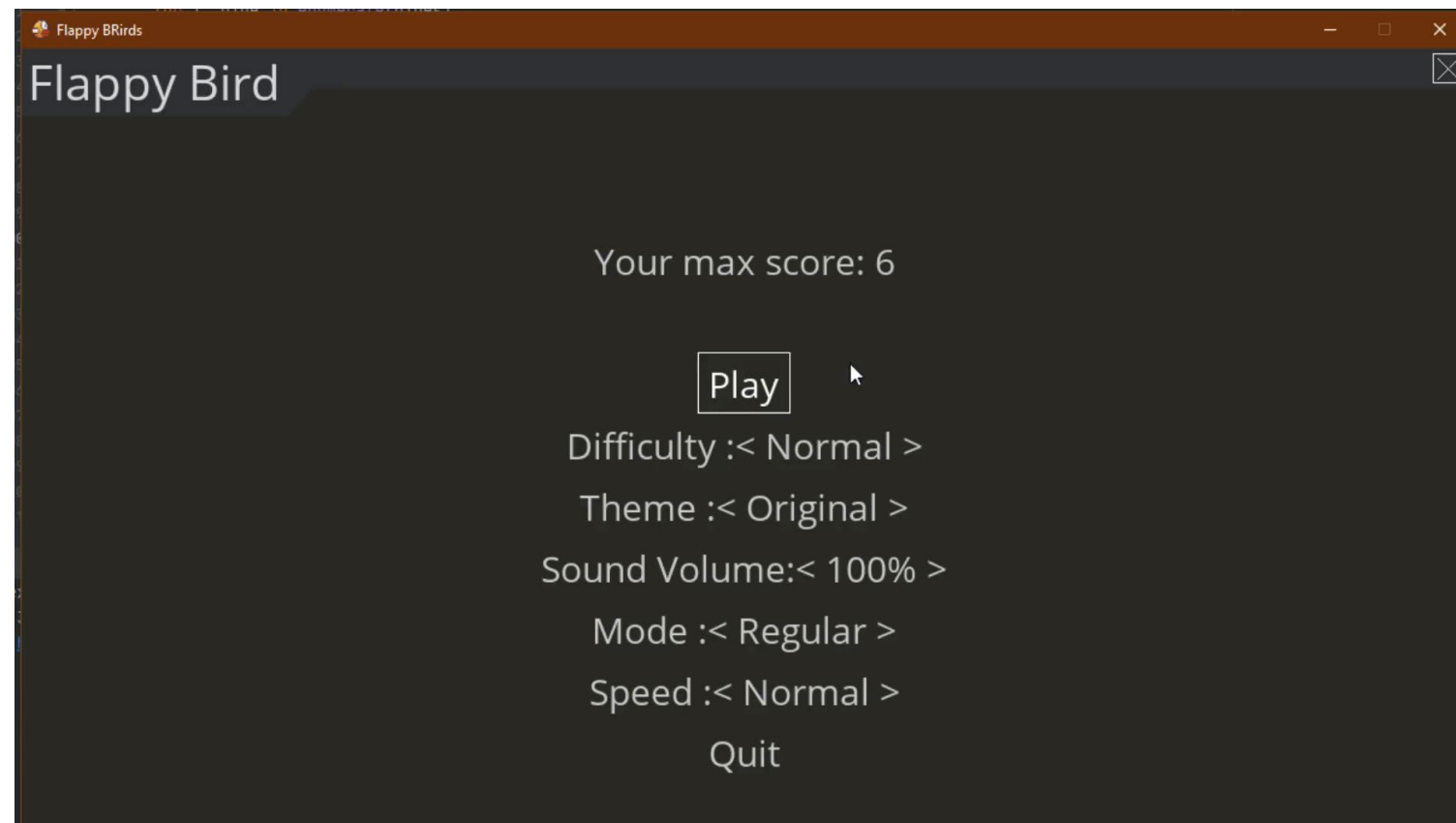
Fonctionnalité 8: Ajouter le score et garder le meilleur score

FONCTIONNALITÉ 9

Fonctionnalité 9: Implémenter un menu



CentraleSupélec

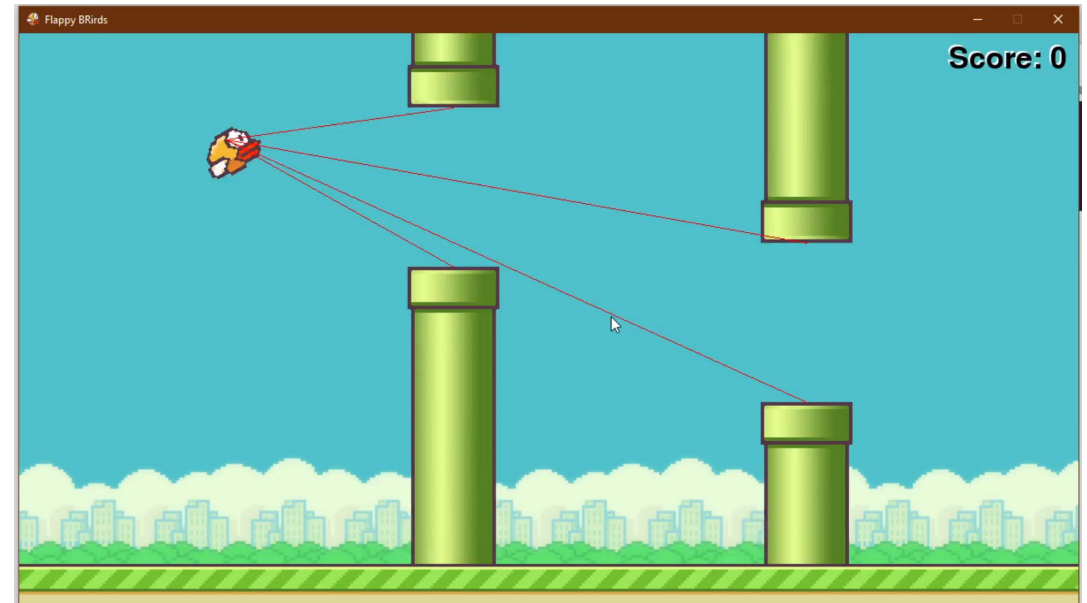
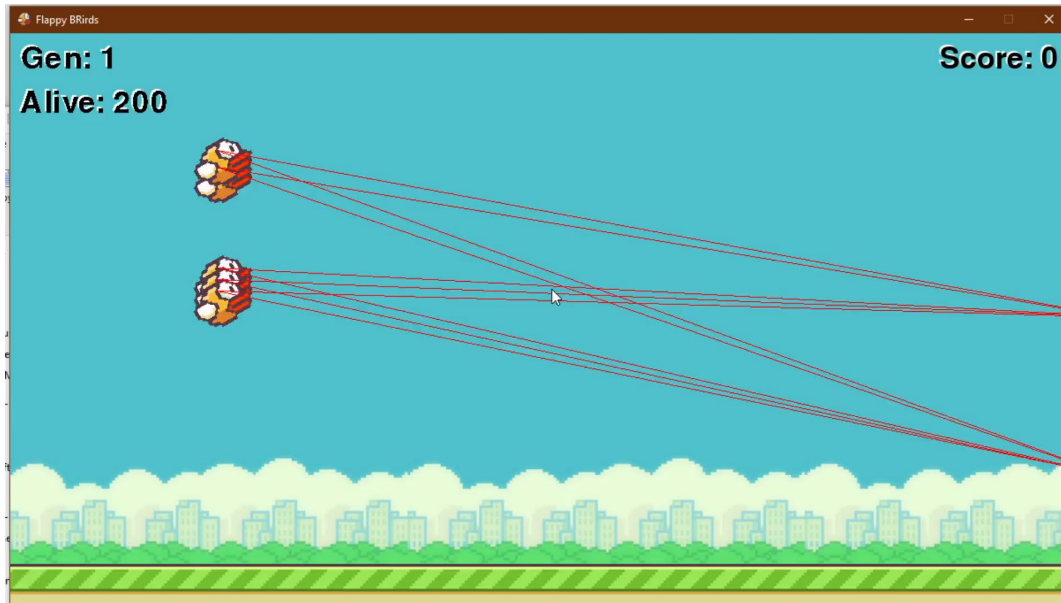


SPRINT 3

- Fonctionnalité I0 – Création d'AI pour jouer (Victor)
 - AI basée dans l'algorithme NEAT (Algorithme génétique)
- Fonctionnalité I1 – Ajouter des publicités (ne pas mis en œuvre)
 - Utilisation des services google pour avoir un flux de recettes
- Fonctionnalité I2 – Différents niveaux de difficulté (Yuichi)
 - Ajouter des tubes que sont mobiles
 - Vitesse variable des tubes
 - Ajouter des tubes avec lacune de taille dynamique
- Fonctionnalité I3 – Thèmes (Yuichi)
 - Ajouter des options de thème
- Fonctionnalité I4 – Fin de jeu (Rafael)
 - Créer le menu de fin du jeu



FONCTIONNNALITÉ 10 – INTELLIGENCE ARTIFICIELLE

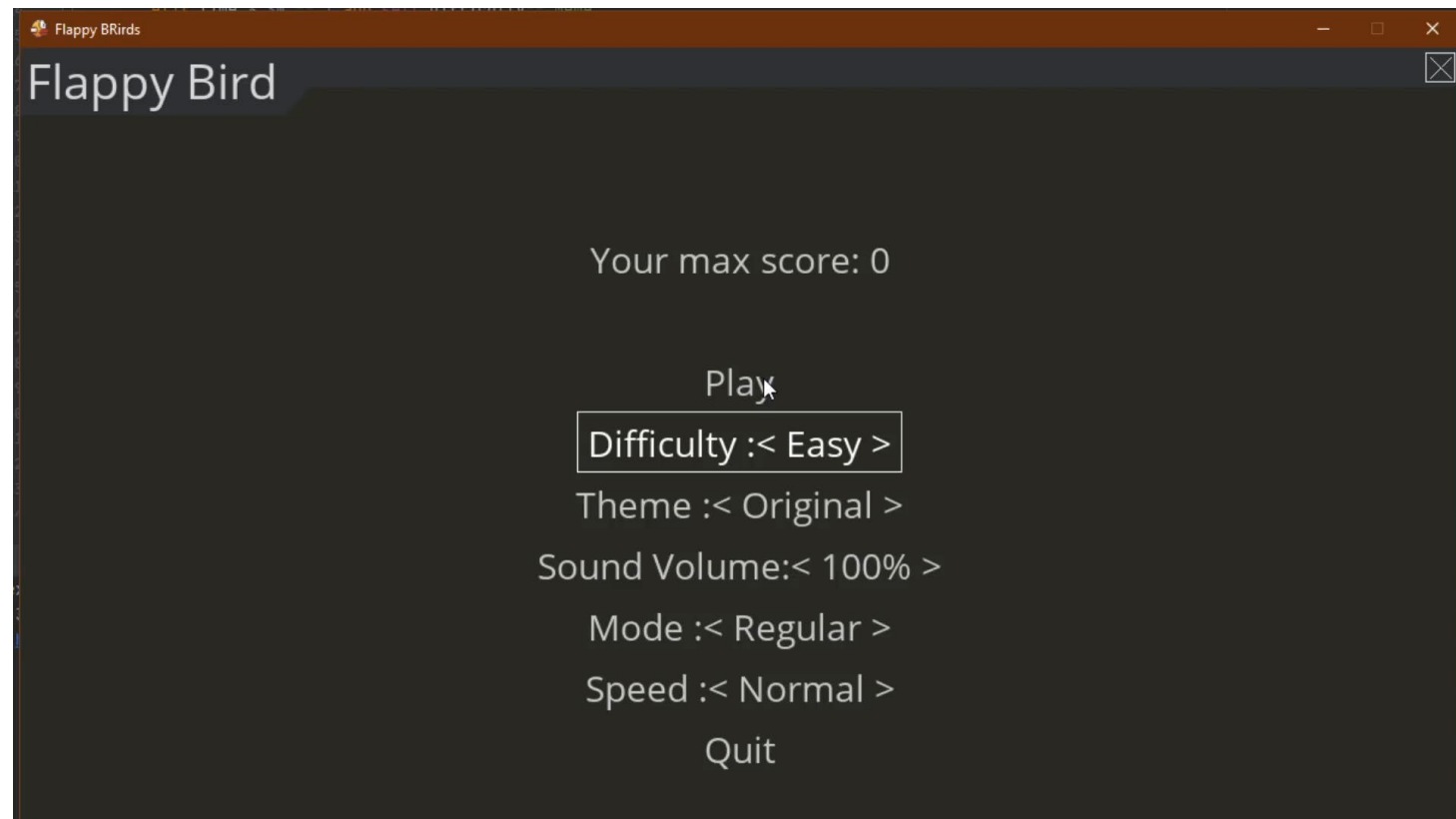


FONCTIONNalité 12

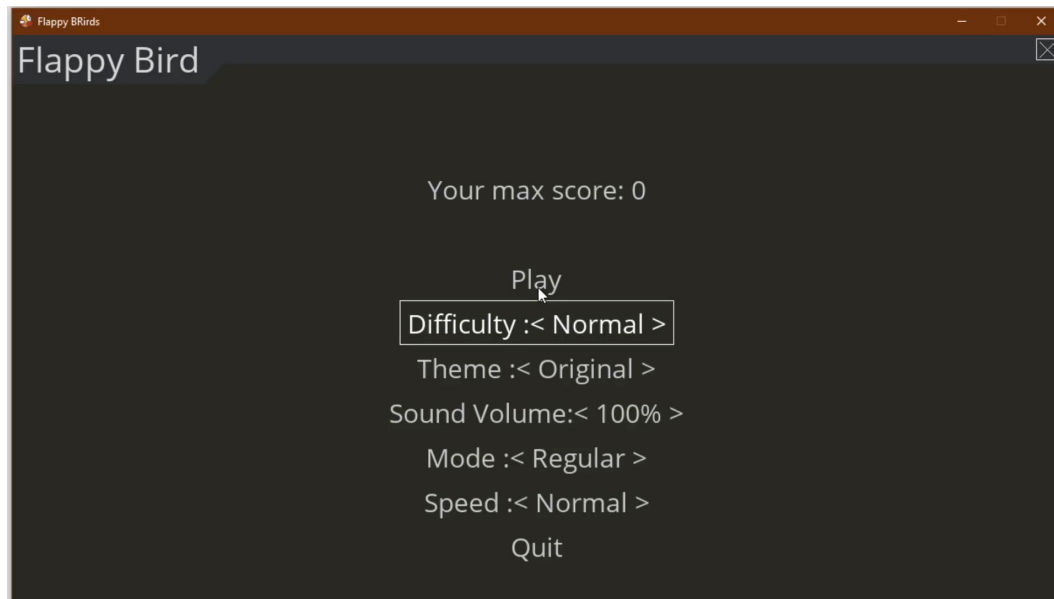
Fonctionnalité 12: Ajouter différents difficultés pour le jeu



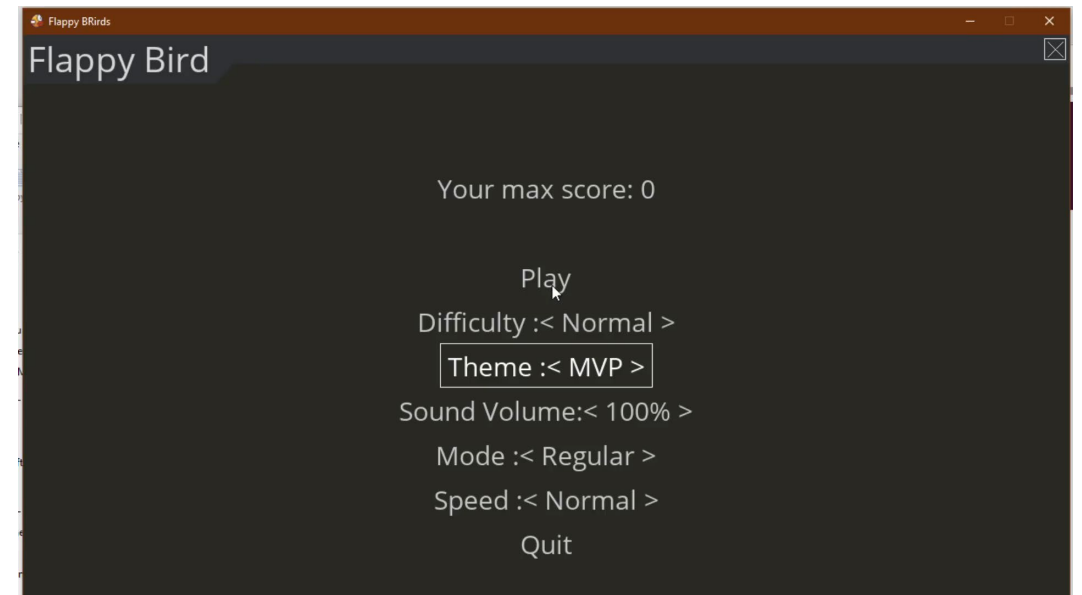
CentraleSupélec



FONCTIONNalité 13 – THÈMES

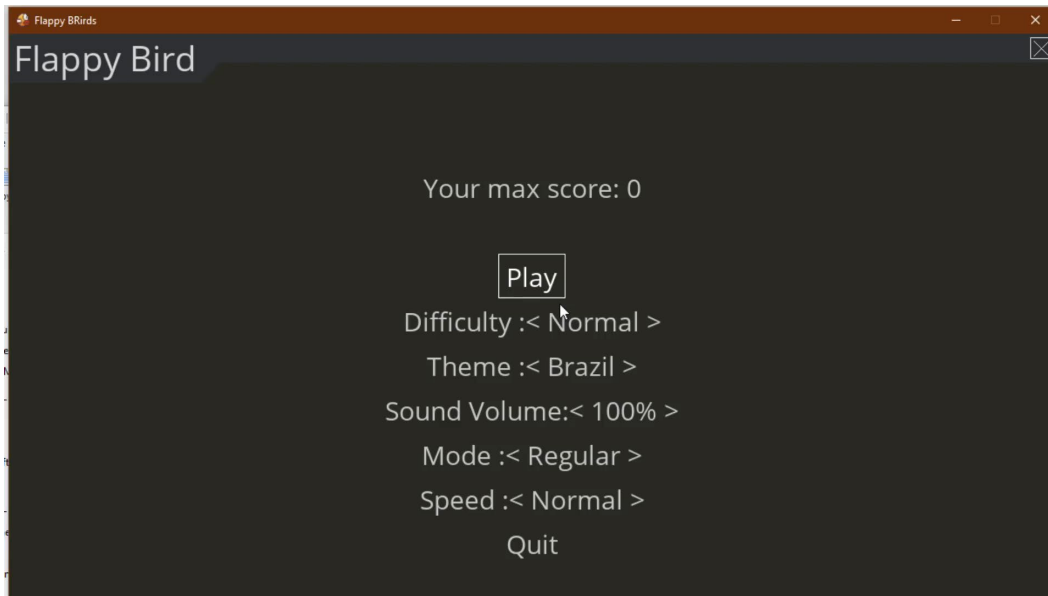


Original

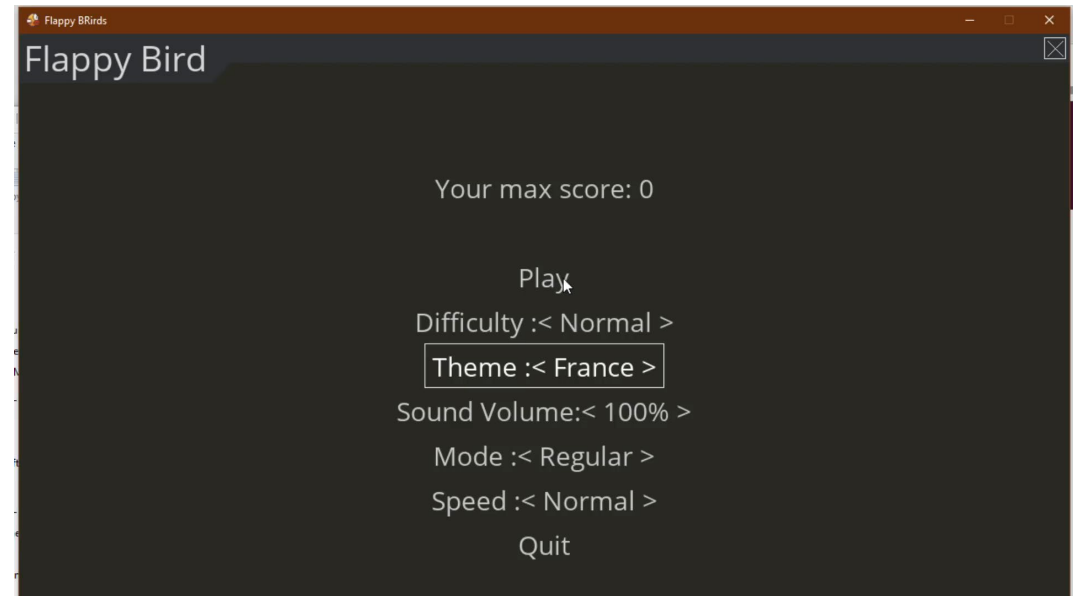


MVP

FONCTIONNalité 13 – THÈMES

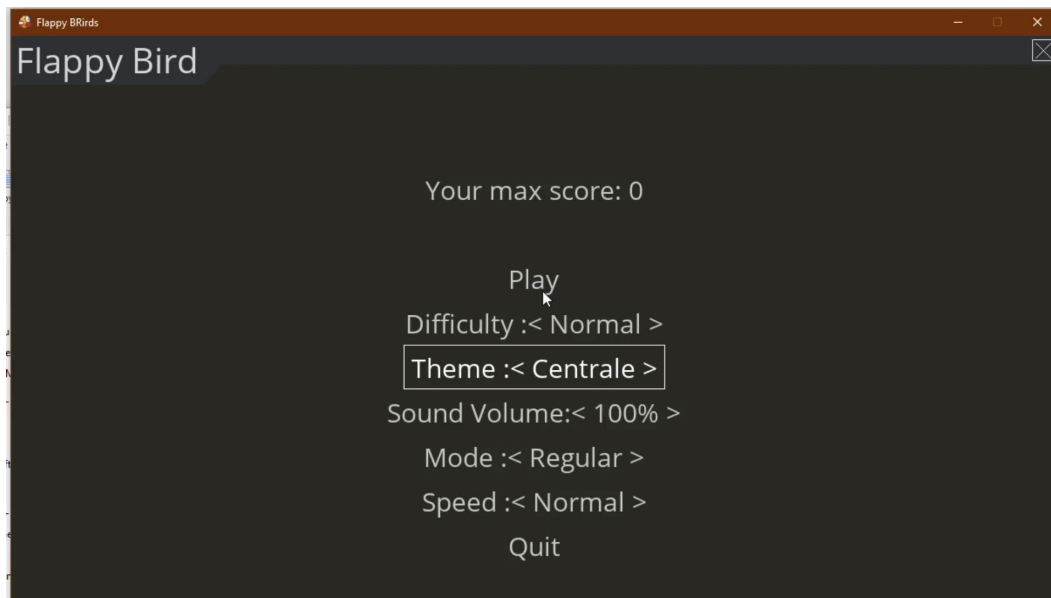


Brésil

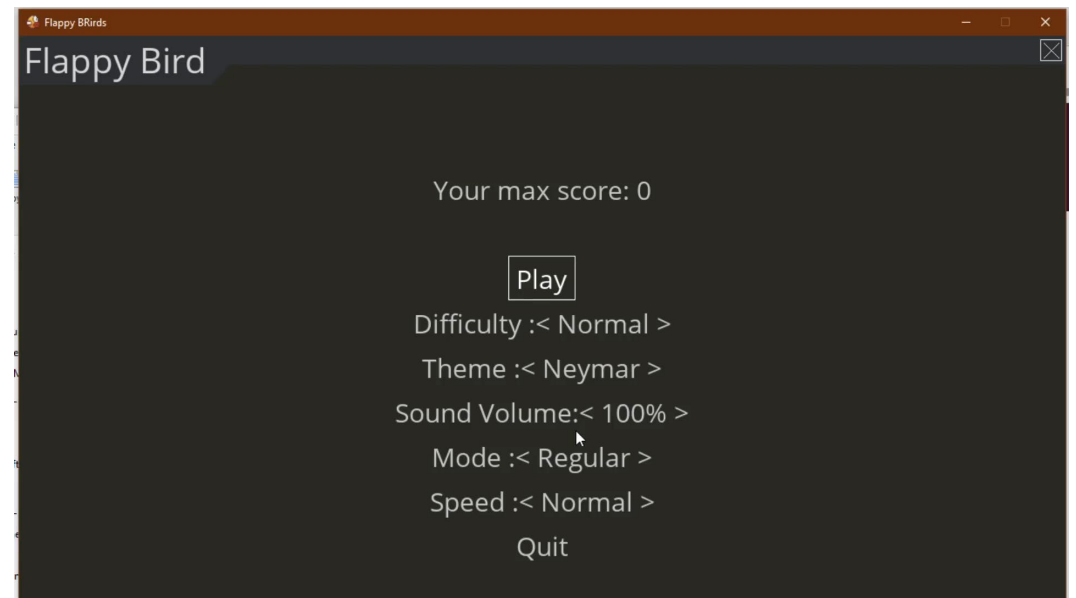


France

FONCTIONNalité 13 – THÈMES



Centrale



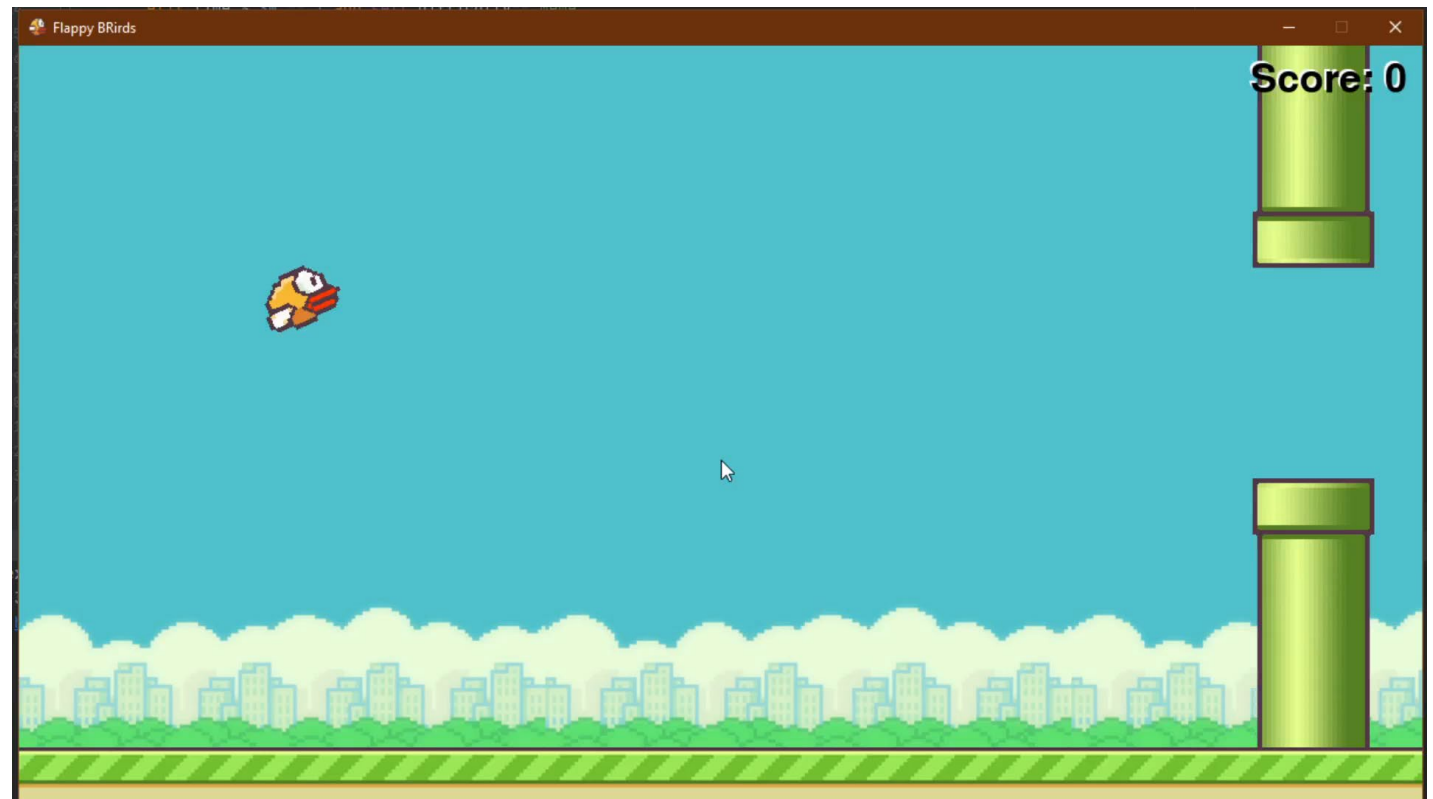
Neymar

FONCTIONNALITÉ 14

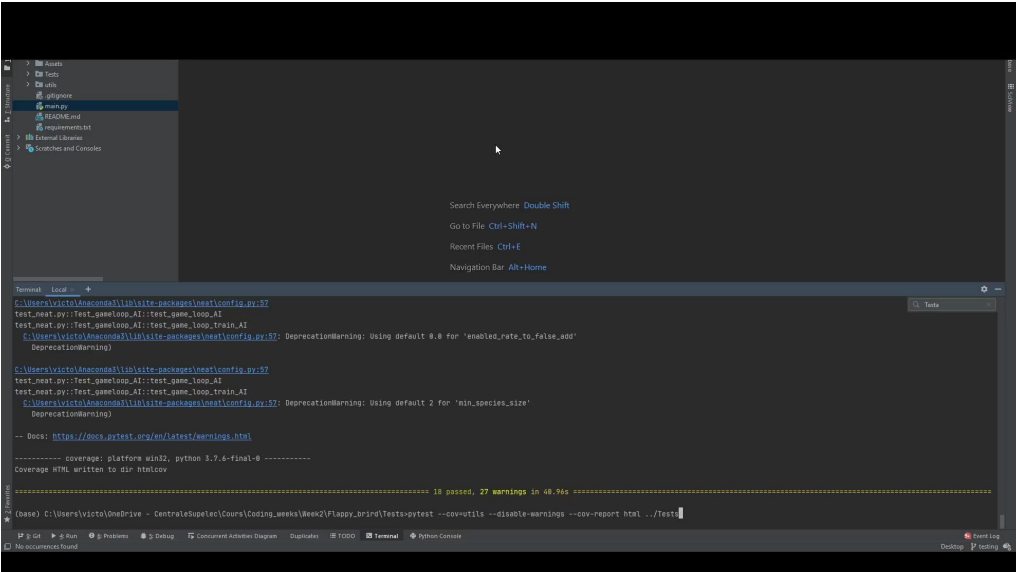
Fonctionnalité 14: Créer un menu pour la fin du jeu



CentraleSupélec



PYTESTING & COVERAGE



PYTESTING & COVERAGE

Coverage report: 90%

Module ↑	statements	missing	excluded	coverage
C:\Users\victo\OneDrive - CentraleSupélec\Cours\Coding_weeks\Week2\Flappy_bird\utils\Game.py	106	18	0	83%
C:\Users\victo\OneDrive - CentraleSupélec\Cours\Coding_weeks\Week2\Flappy_bird\utils\NEAT.py	211	21	0	90%
C:\Users\victo\OneDrive - CentraleSupélec\Cours\Coding_weeks\Week2\Flappy_bird\utils__init__.py	0	0	0	100%
C:\Users\victo\OneDrive - CentraleSupélec\Cours\Coding_weeks\Week2\Flappy_bird\utils\bird.py	87	1	0	99%
C:\Users\victo\OneDrive - CentraleSupélec\Cours\Coding_weeks\Week2\Flappy_bird\utils\gamesettings.py	22	6	0	73%
C:\Users\victo\OneDrive - CentraleSupélec\Cours\Coding_weeks\Week2\Flappy_bird\utils\ground.py	28	0	0	100%
C:\Users\victo\OneDrive - CentraleSupélec\Cours\Coding_weeks\Week2\Flappy_bird\utils\pipe.py	122	9	0	93%
C:\Users\victo\OneDrive - CentraleSupélec\Cours\Coding_weeks\Week2\Flappy_bird\utils\radio.py	27	0	0	100%
C:\Users\victo\OneDrive - CentraleSupélec\Cours\Coding_weeks\Week2\Flappy_bird\utils\score.py	34	4	0	88%
C:\Users\victo\OneDrive - CentraleSupélec\Cours\Coding_weeks\Week2\Flappy_bird\utils\score_display.py	50	8	0	84%
Total	687	67	0	90%

coverage.py v5.3, created at 2020-11-19 22:24 +0100

README

<https://hackmd.io/yM9TaIqYR5er-HbKt0fpBQ?view>

**Vous pouvez trouver
dans le Readme**

*Disponible aussi sur le gitlab

- Description
- Progress
- Visuels
- Installation
- Usage
- Authors & contributions
- Testing
- License



FLAPPY **BR**IRD



MERCI BEAUCOUP !

