



UNIVERSITÉ
CAEN
NORMANDIE

- Soutenance - Conception d'un Tetris

ANDRÉ Lorada (21809742)
LEBRANCHU Paul (21403460)
LEVESQUE Willy (21808901)
LOPEZ-PARDO Hugues (21803489)

Université de Caen Normandie

April 5, 2020

- 1 Introduction
- 2 Gestion des écrans du menu
- 3 Explication des mécanismes du jeu
- 4 Pistes intelligence artificielle
- 5 Explication piste réseau
- 6 Démonstration
- 7 Conclusion

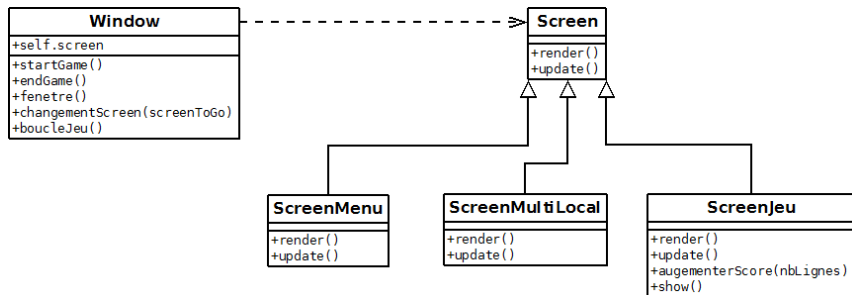


- Objectif: concevoir un jeu Tetris en python, avec la bibliothèque Pygame.
- D'après Wikipédia, le Tetris est "un jeu vidéo de puzzle conçu par Alekseï Pajitnov à partir de juin 1984".
- Fonctionnalités: Tetris basique, mode multijoueurs.

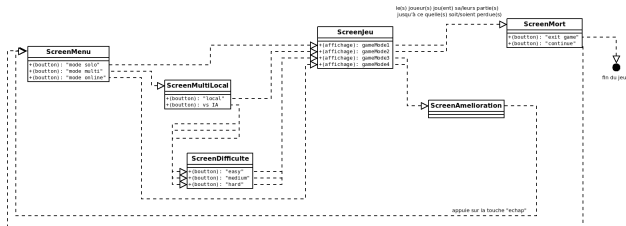
Pourquoi une gestion des écrans?

- plusieurs écrans dans le jeu
- meilleure lisibilité du code
- meilleure organisation

Structure



Le passage d'un écran à l'autre



```

if event.type == MOUSEBUTTONDOWN: #Si un de ces événements est de type MOUSEBUTTONDOWN (clic souris)
    if event.button == 1 and event.pos[0] >= 350 and event.pos[0] <= 650 and event.pos[1] <=400 and event.pos[1] >=300:
        self.gameMode = 1 #solo
        self.window.changeScreen(ScreenJeu(self.window, self.gameMode))
    if event.button == 1 and event.pos[0] >= 350 and event.pos[0] <= 650 and event.pos[1] <=525 and event.pos[1] >=425:
        self.window.changeScreen(ScreenMultiLocal(self.window))
    if event.button == 1 and event.pos[0] >= 350 and event.pos[0] <= 650 and event.pos[1] <=650 and event.pos[1] >=550:
        self.gameMode = 4 #online
        self.window.changeScreen(ScreenJeu(self.window, self.gameMode))
  
```

```

def changeScreen(self, screenToGo):
    self.screen = screenToGo
  
```

Création d'un tableau puis affichage du contenu du tableau:



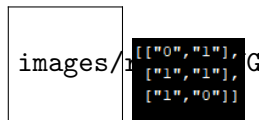
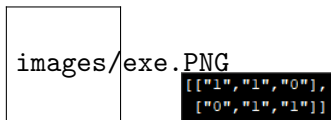
Gestion de la chute des pièces et des déplacements

1 Sélection des pièces



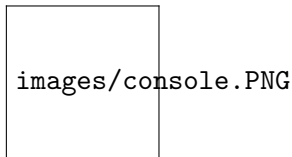
2 Déplacement vers la gauche, vers la droite, vers le bas et chute automatique

3 Rotation des pièces



Gestion des collisions, du score et des niveaux

- 1 Gestion de la collision des pièces (pygame: collidelistall)
- 2 Implémentation des pièces dans la grille































- 3 Suppression des lignes complètes, ligne malus, score et niveau

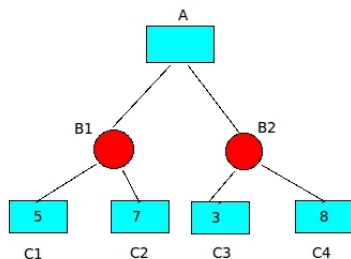
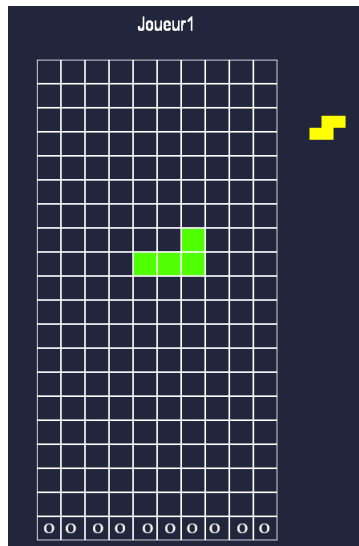


Calcul du score: $(25 * n + 100 * (n-1)) * \text{niveau}$

La logique du bot

I	O	T	L	J	Z	S	
							no rotation
							1 rotation
							2 rotations
							3 rotations

Application de l'algorithme dans le logiciel

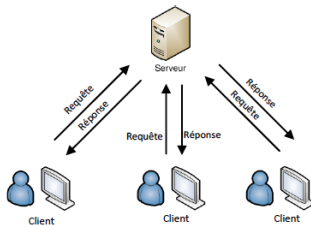


Gestion de l'audio

```
1  import pygame
2
3  def theme():
4      pygame.mixer.init()
5      pygame.mixer.music.load("assets/music/theme.mp3")
6      pygame.mixer.music.play(loops=-1, start=0.0)
7      pygame.mixer.music.set_volume(1)
```

```
1  import pygame
2  from files.Window import *
3  from gestionAssets.LoadingMusic import *
4
5
6  theme()
7  Window()
```

Fonctionnement du réseau



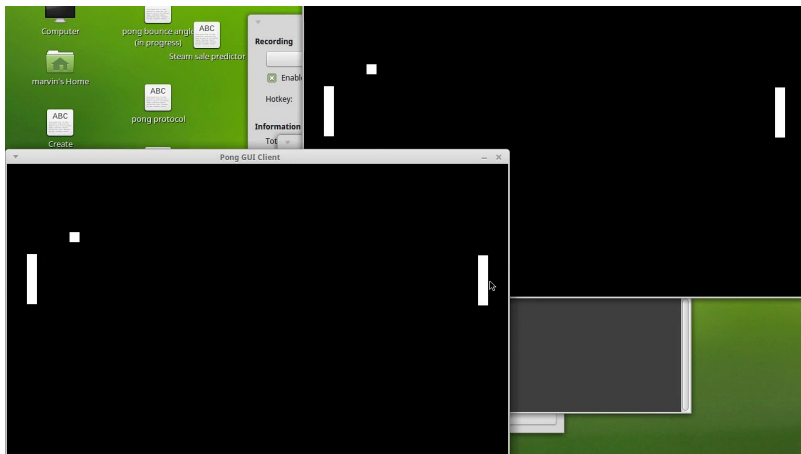
Lancement du serveur :

Le serveur est démarré

Un client vient de se connecter au serveur :



Application du client et du serveur avec pygame



Début de la démonstration!

- Un projet mené à bien malgré quelques bugs
- Piste d'amélioration: un mode puzzle et de nouvelles fonctionnalités multijoueurs

Merci de votre attention