

## Projet SIMON

### 1. Objectifs du projet.

- Utilisation de différentes structures de données (p-uplets, listes, dictionnaires, ...)
- Prise en main d'une bibliothèque : Pygame
- Construction d'une IHM

### 2. Descriptif du projet

Nous allons programmer en Python le jeu SIMON : <https://www.youtube.com/watch?v=1Yqj76Q4jJ4>

Le projet sera découpé en étapes et chaque semaine je vous proposerai d'avancer dans ce projet.

### 3. Semaine 1.

#### 3.1 Installation de l'environnement de travail.

Vous avez besoin d'un IDE pour programmer en Python 3 et de la bibliothèque Pygame. Plusieurs propositions :

- Si vous avez installé Edupython, normalement la bibliothèque Pygame est installée avec, pour le vérifier, taper `import pygame` dans la console :

```
Console Python
*** Python 3.4.5 |Continuum Analytics, Inc.| (default, Jul 5 2016, 14:56:50)
*** Distant Python engine is active ***
>>> import pygame
pygame 1.9.4
Hello from the pygame community. https://www.pygame.org/contribute.html
>>> |
```

Si vous n'avez pas de message d'erreur, c'est que l'installation est correcte.

- Si vous n'avez pas d'environnement correct pour programmer en Python, je vous propose d'installer Thonny, puis la bibliothèque Pygame en suivant les 2 vidéos : `install_thonny.mp4` et `install_pygame.mp4`

#### 3.2 Préparation de la fenêtre.

Pour cette semaine, nous allons nous contenter de préparer la fenêtre avec les blocs de couleurs sur lesquels le joueur devra cliquer et les fichiers son correspondant à chaque couleur.

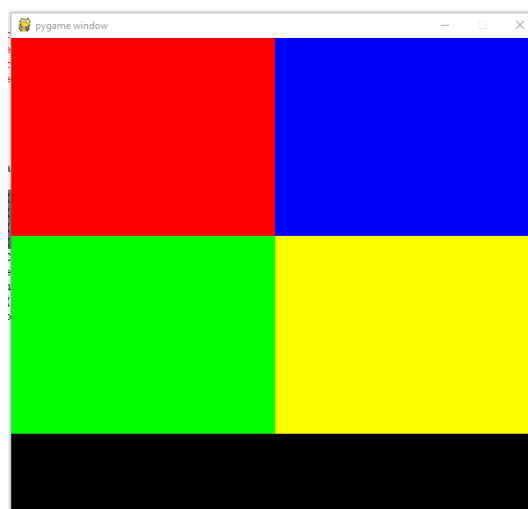
Voici le code du début du jeu :

```

simonV0.py ×
1  """Projet NSI
2  Jeu de mémoire SIMON
3  Codé par ..... le ....."""
4
5  import pygame,sys
6  import pygame.event as GAME_EVENTS
7  import pygame.locals as GAME_GLOBALS
8
9  pygame.init()
10
11  jaune=(255,255,0)
12  rouge=(255,0,0)
13  vert=(0,255,0)
14  bleu=(0,0,255)
15  blanc=(255,255,255)
16  noir=(0,0,0)
17
18  fenetre=pygame.display.set_mode((640,580))
19
20
21  while True:
22      pygame.draw.rect(fenetre,rouge,(0,0,320,240))
23      pygame.draw.rect(fenetre,vert,(0,240,320,240))
24      pygame.draw.rect(fenetre,bleu,(320,0,320,240))
25      pygame.draw.rect(fenetre,jaune,(320,240,320,240))
26      for event in GAME_EVENTS.get():
27          if event.type == GAME_GLOBALS.QUIT:
28              pygame.quit()
29              sys.exit()
30
31      pygame.display.update()

```

Et le rendu attendu :



### 3.3 Préparation des sons

Lorsqu'on clique sur un rectangle, il se produit un son, différent pour chaque couleur.

Cherchez 4 fichiers son courts (1 s max), au format .wav, que vous attribuerez aux quatre couleurs.

Vous nommerez ces fichiers sonRouge.wav, sonBleu.wav, sonJaune.wav et sonVert.wav.

### 3.4 Approfondissement

L'interface avec les rectangles ci-dessus permet de jouer, maintenant, si vous voulez approfondir et personnaliser le jeu vous pouvez explorer les options de la méthode draw sur la page suivante :

<https://www.pygame.org/docs/ref/draw.html>