

R1.01 - Séance spécifique pour la SAÉ

Notions supplémentaires

1 pygame



A l'instar de Tk, **pygame** est une bibliothèque facilitant le développement d'applications graphiques. Elle est orientée jeu vidéo et permet d'afficher facilement des éléments graphiques tels que des formes diverses colorées ou des images. Elle facilite également la gestion du clavier et de la souris mais aussi les joysticks.

Vous pouvez consulter le site web de la bibliothèque à l'adresse suivante :
<https://www.pygame.org>

Installation

Comme tout package python, vous pouvez installer **pygame** simplement à l'aide d'un terminal :

```
pip install pygame
```

(ou pip3, en fonction de votre environnement de travail)

Fonctionnement

Un programme développé à l'aide de **pygame** respecte généralement le modèle suivant :

Création d'une fenêtre

Boucle de jeu

Gestion des interactions clavier/souris

Modification du contexte de jeu

Mise à jour de l'affichage

Création d'une fenêtre et initialisation

Pour utiliser `pygame` dans un programme, vous devrez, avant tout, ne pas oublier d'importer la bibliothèque :

```
import pygame
```

La création et initialisation d'un programme utilisant cette bibliothèque nécessite l'exécution de plusieurs fonctions :

- `pygame.init()` : initialise la bibliothèque.
- `pygame.time.Clock()` : récupère une horloge qui sera utilisée pour limiter le nombre d'images par seconde.
- `pygame.display.set_mode(...)` : crée la fenêtre, donne sa taille et renvoie la variable à utiliser pour utiliser la fenêtre.
- `pygame.display.set_caption(...)` : modifie le nom de la fenêtre.

Boucle de jeu

La boucle de jeu permet la gestion du jeu. C'est elle qui va relancer le jeu pour l'éternité! (ou jusqu'à ce que la fin du jeu soit détecté).

La boucle va relancer la gestion du clavier/souris et leur conséquence sur le jeu. L'affichage est ensuite mis à jour. L'instruction `tick(...)` permettra de faire les pauses nécessaires pour limiter le nombre d'images par seconde.

affichage de primitives

`pygame` propose de nombreuses primitives à afficher grâce, par exemple, aux instructions : `pygame.draw.rect(...)`, `pygame.draw.circle(...)` ou encore `pygame.draw.polygon(...)`. Il est également possible d'afficher facilement des images.

Gestion clavier/souris

L'instruction `pygame.event.get()` permet de récupérer toutes les informations sur les entrées (clavier/souris) détectées par le programme. A partir de cette instruction, il est possible de retrouver le type d'entrée détectée ainsi

que la touche appuyée ou le clic pressé. Le code suivant permet d'utiliser ces deux périphériques :

```
for event in pygame.event.get():
    if event.type == pygame.QUIT:
        stop = True
    elif event.type == pygame.MOUSEBUTTONDOWN:
        if pygame.mouse.get_pressed()[0] == True:
            print('Vous avez appuyé sur le clic gauche')
    elif event.type == pygame.KEYDOWN:
        if event.key == pygame.K_UP:
            print('Vous avez appuyé sur la flèche du haut')
```

2 Le projet

L'existant

Le projet sur lequel vous allez travailler est un jeu de tir. Au lancement du programme, une fenêtre de jeu apparaît ainsi que des cibles vertes qui disparaissent petit à petit. Le but du jeu est de cliquer sur le plus de cibles possibles en un temps limité.

TODO

Vous allez faire en sorte que ce jeu devienne jouable à deux. Le premier joueur continue à tirer sur les cibles tandis que le second joueur va contrôler l'ensemble des cibles pour empêcher le premier joueur de les toucher. Le second joueur devra pouvoir déplacer les cibles dans une des quatre directions.

Pour ce faire, suivez les instructions suivantes. Les questions

1. Lancez le jeu pour le tester puis étudiez le code.
2. Faites en sorte que la fonction `manage_events` affiche l'ensemble des informations des cibles **dans le terminal**.
3. Ajoutez une procédure qui sera appelée lorsqu'une touche du clavier sera pressée. Cette fonction affichera l'ensemble des informations des cibles dans le terminal.
4. Modifiez cette procédure pour faire en sorte que les cibles soient déplacées en accord avec la touche pressée. Seules les flèches de directions produiront un déplacement. Utilisez la constante `TARGET_SPEED` présente dans le fichier `constants.py`
5. Testez le jeu avec une autre personne!