

# Python : création d'un jeu vidéo (mini projet):

Pygame est une bibliothèque libre multiplate-forme qui facilite le développement de jeux vidéo en temps réel avec le langage de programmation Python.

Construite sur la bibliothèque SDL, elle permet de programmer la partie multimédia (graphismes, son et entrées au clavier, à la souris ou au joystick), sans se heurter aux difficultés des langages de bas niveaux. Cela se fonde sur la supposition que la partie multimédia, souvent la plus contraignante à programmer dans un jeu, est suffisamment indépendante de la logique même du jeu pour qu'on puisse utiliser un langage de haut niveau (en l'occurrence le Python) pour la structure du jeu.



Pygame, en plus d'adapter la SDL au Python, fournit également un petit nombre de fonctions spécifiques au développement de jeux. On peut aussi remarquer que Pygame n'est plus utilisée exclusivement pour des jeux vidéo, mais également pour des applications diverses nécessitant du graphisme.

## 1. Ouverture d'une fenêtre:

On s'intéresse au code suivant:

```
1 #Importation des bibliothèques nécessaires
2 import pygame
3 from pygame.locals import *
4
5 #Initialisation de la bibliothèque Pygame
6 pygame.init()
7
8 #Création de la fenêtre
9 fenetre = pygame.display.set_mode((640, 480))
10
11 #Variable qui continue la boucle si = 1, stoppe si = 0
12 continuer = 1
13
14 #Boucle infinie
15 while continuer:
16     continue #Je place continue ici pour pouvoir relancer la boucle infinie
17     |         | #mais il est d'habitude remplacé par une suite d'instructions
18
```

1.1 Expliquer ce code.

1.2 Est-il possible de fermer la fenêtre?

## 2. Affichage d'un fond et d'un personnage:

On s'intéresse au code suivant:

```
1 import pygame
2 from pygame.locals import *
3
4 pygame.init()
5
6 #Ouverture de la fenêtre Pygame
7 fenetre = pygame.display.set_mode((640, 480))
8
9 #Chargement et collage du fond
10 fond = pygame.image.load("background.jpg").convert()
11 fenetre.blit(fond, (0,0))
12
13 #Chargement et collage du personnage
14 perso = pygame.image.load("perso.png").convert()
15 fenetre.blit(perso, (200,300))
16
17 #Rafraîchissement de l'écran
18 pygame.display.flip()
19
20 #BOUCLE INFINIE
21 continuer = 1
22 while continuer:
23     continuer = int(input())
24
```

2.1 Expliquer ce code.

2.2 Gérer la transparence du personnage en remplaçant `convert()` par `convert_alpha()`.

## 3. Gestion des événements:

Un événement peut prendre plusieurs formes, il peut être amené par la pression ou le relâchement d'une touche du clavier, ou encore d'un bouton de la souris, un mouvement de la souris, du joystick, etc... Mais il peut aussi être un déplacement ou un redimensionnement de la fenêtre. Un événement est donc tout ce que le programme peut capter, de la part de l'utilisateur.

On s'intéresse au code suivant (le début reste identique aux codes précédents):

```
13 #Chargement et collage du personnage
14 perso = pygame.image.load("perso.png").convert_alpha()
15 position_perso = perso.get_rect()
16 fenetre.blit(perso, position_perso)
17
18 #Rafraîchissement de l'écran
19 pygame.display.flip()
20
21 #BOUCLE INFINIE
22 continuer = 1
23 while continuer:
24     for event in pygame.event.get(): #Attente des événements
25         if event.type == QUIT:
26             continuer = 0
27         if event.type == KEYDOWN:
28             if event.key == K_DOWN: #Si "flèche bas"
29                 #On descend le perso
30                 position_perso = position_perso.move(0,3)
31
32     #Re-collage
33     fenetre.blit(fond, (0,0))
34     fenetre.blit(perso, position_perso)
35     #Rafraîchissement
36     pygame.display.flip()
```

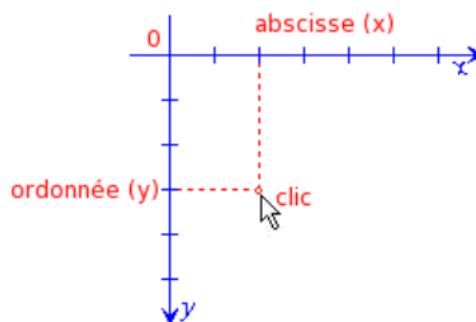
3.1 Expliquer ce code.

3.2 Placer avant la boucle principale la commande `pygame.key.set_repeat(400, 30)` afin de ne pas avoir à appuyer sur bas plusieurs fois pour déplacer le personnage.

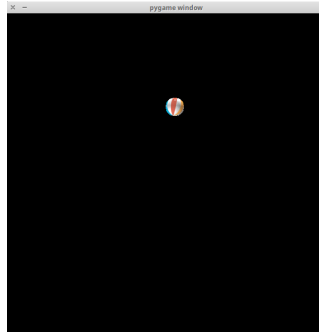
3.3 Modifier le code afin que le personnage puisse se déplacer dans les quatre directions.

3.4 Modifier encore le code afin que le personnage ne puisse pas sortir de la fenêtre.

NB: les coordonnées d'un point de la fenêtre sont données comme suit:

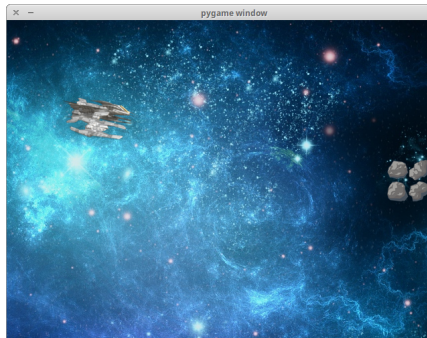


#### 4. Un premier jeu: Bouncing Ball:



- 4.1 Télécharger le code et les images correspondant à l'exercice Bouncing ball.
- 4.2 Modifier le code afin que la balle soit sombre lorsqu'elle descend et lumineuse lorsqu'elle monte.
- 4.3 Ajouter un mode pause via la touche espace.
- 4.4 Ajouter la possibilité de d'accélérer la balle via les touches haut/bas.

#### 5. Un second jeu: Starship:



- 5.1 Télécharger le code et les images correspondant à l'exercice Bouncing ball.
- 5.2 Modifier le code afin qu'une explosion apparaisse lors de la rencontre du vaisseau et de l'astéroïde.
- 5.3 Modifier le code afin que le fond défile.
- 5.4 Ajouter une musique au jeu.
- 5.5 Ajouter un laser tiré par le vaisseau. Ce dernier doit provoquer une explosion lors de sa rencontre avec l'astéroïde.