



# Mon Petit Python

Coding Club by Epitech

3 juillet 2017



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Une fenêtre</b>	<b>5</b>
2.1	La base . . . . .	5
2.2	Regardons la fenêtre . . . . .	6
2.3	Votre première image . . . . .	7
<b>3</b>	<b>Du Déplacement</b>	<b>8</b>
<b>4</b>	<b>Une carte</b>	<b>9</b>
<b>5</b>	<b>Liberté</b>	<b>10</b>



# 1 Introduction

Pygame est un module graphique en python permettant de faire des jeux.

Aujourd'hui, nous allons faire un jeu d'action dans le type des vieux Zelda.



## 2 Une fenêtre

### 2.1 La base

On va ouvrir un nouveau fichier python dans lequel on va commencer par importer la bibliothèque.  
Pour cela, écrivez la ligne :

```
1 import pygame
```

Plutôt simple et explicité, n'est-ce pas ?! M

Ensuite, il faut ajouter :

```
1 from pygame.locals import *
```

Cela permet d'importer les variables constantes de pygame.  
Et pour finir, il faut initialiser pygame avec :

```
1 pygame.init()
```

## 2.2 Regardons la fenêtre

Pour créer une fenêtre, il faut appeler la fonction "pygame.display.setmode". Elle prend en paramètre la largeur et la hauteur de la fenêtre dans un tuple.

```
1 fenetre = pygame.display.set_mode((640, 480))
```



Tips : Le tuple est un type de variable que l'on déclare en mettant entre parenthèse une suite de valeurs séparées par des virgules. Dans l'exemple précédent, il s'agissait de (640,480).

Maintenant, faites comme moi et créez votre fenêtre que vous stockez dans la variable window.

Lancez votre script et vous avez une fenêtre !

Mais elle se ferme tout de suite.

Pour empêcher cela, mettez une boucle dans votre code et voilà !

## 2.3 Votre première image

Créez une classe Personnage qui va contenir une variable avec le chemin vers une image, ainsi que deux variables pour la position X et Y du joueur.

Initialisez ces trois variables dans le constructeur. Vous trouverez l'image du joueur dans le dossier fourni. Ensuite, ajoutez une variable image et vous allez l'initialiser avec la fonction suivante :

```
1 pygame.image.load(chemin_vers_l_image).convert_alpha()
```

La fonction load permet de charger l'image et la fonction convertalpha permet de la rendre compatible avec pygame.

Pour finir, créez une fonction show prenant en paramètre la fenêtre et affichez le joueur en fonction de sa position X et Y.

La fonction pour cela est :

```
1 fenetre.blit(image, (X, Y))
```

Python

Créez le joueur dans le code et affichez-le avec la fonction.

Relancez votre programme et c'est bon !



Tips : Affichez votre joueur dans votre boucle infinie !

N'oubliez pas d'appeler la fonction pygame.display.flip() pour rafraichir l'écran. Sinon, vous ne verrez jamais votre joueur.



### 3 Du Déplacement

Nous allons maintenant ajouter de la gestion d'événement à notre jeu. C'est-à-dire que nous allons faire en sorte d'associer les touches du clavier à un événement tel que le déplacement du personnage. Je ne sais pas comment vous avez fait pour que votre fenêtre ne se ferme pas mais maintenant, nous allons utiliser une boucle while. Pour cela, créez une variable continuer que vous initialisez à 1. Ensuite, faite :

```
1 while continuer :
```

Python

Et dans cette boucle, on va rajouter une autre boucle pour récupérer les événements.

```
1 while continuer :  
2     for event in pygame.event.get() :  
3         if event.type == QUIT :  
4             continuer = 0
```

On va récupérer chaque événement qui survient dans le jeu, le stocker dans la variable event et regarder si l'événement correspond à ce que l'on veut.

Exemple : event.type == QUIT où l'on regarde si on ferme la fenêtre.

Bien, maintenant, je veux que l'on quitte le jeu si on appuie sur "Echap". Pour cela, on va détecter que le type de l'événement est un pressage de touche avec event.type == KEYDOWN. Puis avec l'élément event.key, on va vérifier que la touche Echap est pressée.

On va donc avoir :

```
1 if event.type == KEYDOWN && event.key == K_ESCAPE :
```

Si c'est bon, on met continuer à zéro pour quitter la boucle. Maintenant, faites déplacer votre joueur sur l'écran avec les touches 'zqsd'.



## 4 Une carte

Bien, nous allons faire une carte pour rendre ce jeu plus sympa et plus beau !

Pour cela, créez un objet Tile qui aura deux variables : - une image - est-ce un obstacle ou non ?

Et maintenant, créez un tableau à deux dimensions de cet objet. Votre but : créer la carte en mettant des obstacles sur les bords de la map et quelques-uns où vous le souhaitez. La première dimension représente la position X de la tuile et la deuxième, la dimension Y.

Une fois cela fini, faites une boucle d’affichage et voilà ! Vous devriez voir la carte et votre personnage dessus.



Tips : Les images se dessinent au fur et à mesure de l’appel des fonctions.

Vous pouvez maintenant rajouter de la collision personnage/obstacle en comparant la position du personnage à celle de la tuile si c’est un obstacle.





## 5 Liberté

Maintenant que vous en êtes arrivé là, vous avez la base d'un jeu !  
Amusez-vous à le modifier, à ajouter des choses pour avoir un jeu qui vous plaît !  
Et n'hésitez pas à poser des questions aux Cobras !