

Création d'un jeu vidéo à l'aide de Pygame.

Etape 1 :

- Regarder les 4 premières vidéos expliquant le codage du jeu.
[PyGame Tile Based Platformer Game Beginner Tutorial in Python - PART 1 | Creating the World](#)
- Récupérer le code du jeu ainsi que le fichier « image » pour tester le code.
- Choisir des images pour modifier votre jeu.
- Créer votre propre arrière-plan

Etape 2 :

- Ajouter des plateformes qui se déplacent horizontalement et d'autres qui se déplacent verticalement.

Etape 3 :

- Rajouter des objets qui disparaissent au contact du personnage.
- Les objets doivent être animés.
- Une animation de « disparition » doit être prévue.

Etape 4 :

- Faire apparaître des ennemis qui se déplacent automatiquement dont le contact fait perdre des points de vie.
- Ajouter des ennemis fixes dont le contact fait perdre des points de vie.
- Prévoir pour chaque type d'ennemis des animations.

Etape 5 :

- Un des ennemis doit « tirer » de manière automatique.

Etape 6 :

- Le héros doit être en mesure de tirer (limiter le nombre de munitions) pour détruire une des catégories d'ennemis.

Etape 7 : Faire afficher le score, les points de vie, le nombre de munitions ...

- Dans la classe world, créer la méthode `ecrire_message`.

```
#Affichage du score et des vies
def ecrire_message(self,message, message_rectangle, couleur):
    font = pygame.font.SysFont('lato',40,False)
    message = font.render(message, True, couleur)
    screen.blit(message,message_rectangle)
```

- Dans la boucle principale, faire afficher le score et les vies.

```

world.ecrire_message('Vies :',[350,20,25,25],(255,0,0))
world.ecrire_message('{}'.format(player.pointdevie),[450,20,25,25],(255,0,0))
world.ecrire_message('Missiles :',[600,20,25,25],(255,0,0))
world.ecrire_message('{}'.format(player.munitions),[730,20,25,25],(255,0,0))

```

Etape 8: Créer une animation de fin de jeu si l'on gagne puis une autre si l'on perd.

```

pygame.display.update()
if player.pointdevie < 0:
    run = 3
if world.etoiles < 1:
    run = 4
if run == 3:
# ----- Fin Boucle principale -----
#Fin du jeu, début de l'animation si on a perdu
bg = pygame.image.load('image/gameOver.png')
bg = pygame.transform.scale(bg,(450,180))
pygame.mixer.music.stop()
pygame.mixer.init()
avion = "avion.ogg"
pygame.mixer.music.load(avion)
#On joue 1 seule fois le morceau en commençant à la seconde 1
pygame.mixer.music.play(1,1.0)
animationFin = 0
player.rect.y = 200
player.rect.x = 300
while animationFin < 40:
    screen.blit(bg_img1,(0,0))
    screen.blit(bg,(400,150))
    player.image = animation[animationFin%6]
    screen.blit(player.image,player.rect)
    player.rect.x += 8
    player.rect.y += 2
    animationFin += 1
    pygame.display.update()
    # --- on demande d'attendre pour un FPS de 30
    clock.tick(5)
animationFin = 0
while animationFin < 15:
    screen.blit(bg_img1,(0,0))
    screen.blit(bg,(400,150))
    player.image = animation[animationFin//3+6]
    screen.blit(player.image,player.rect)
    player.rect.x += 8
    player.rect.y += 2
    animationFin += 1
    pygame.display.update()
    # --- on demande d'attendre pour un FPS de 30
    clock.tick(5)
explosion.play()
score = 0
pygame.quit()

```

Etape 9 : Rajouter un second niveau.

- S'inspirer du code des animations de fin de partie.
- Ne pas oublier de vider les listes des « groupes ».

```

#Mise à zéro de tous les éléments du jeu
world.tile_list = []
asc_group = pygame.sprite.Group()
asc1_group = pygame.sprite.Group()
Etoiles_group = pygame.sprite.Group()
Oiseaux_group = pygame.sprite.Group()
Crottes_group = pygame.sprite.Group()
Boules_group = pygame.sprite.Group()
Lave_group = pygame.sprite.Group()
world = World(world_data2)
player.rect.x = 50
player.rect.y = 550
world.etoiles = 15

```

J'efface les données du world1, je charge celles du world2 et je modifie les coordonnées du player ainsi que le nombre d'étoiles à attraper.

Etape 10 : Enregistrer le score et le pseudo dans une base de données et faire afficher les meilleurs résultats. Modifier le code pour que les 5 meilleurs scores apparaissent.

A rajouter au début du programme.

```

import sqlite3
from tkinter import *

#Enregistrement des donnees dans une base de donnees
def quitter():
    fen.quit()

def enregistrer():
    global texteSurnom
    texteSurnom = E.get()
    nouvelle_personne=(texteSurnom,score)
    connexion=sqlite3.connect('basedonneesjeu.db')
    curseur=connexion.cursor()
    curseur.execute("CREATE TABLE IF NOT EXISTS PERSONNE(id INTEGER PRIMARY KEY AUTOINCREMENT UNIQUE,surnom TEXT, score INT)")
    curseur.execute("INSERT INTO PERSONNE(surnom,score) VALUES(?,?)",nouvelle_personne)
    connexion.commit()
    curseur.close()
    connexion.close()
    fen.quit()

```

A rajouter après l'animation de fin de jeu. L'ensemble du fichier se trouve dans module1.py

Penser à enlever la ligne score = 183

```

#Saisie du pseudo pour enregistrer le score dans la base de donnees
if score>0:
    fen =Tk()
    fen.title('Enregistrer les scores.')
    fen.geometry('500x120')
    L=Label(fen,text="Entrez votre pseudo:",width=35,fg="blue",bg="yellow")
    L.place(x=20,y=20)
    E=Entry(fen)
    E.place(x=280,y=20)
    E.focus()
    B1=Button(fen,text="Valider",fg="green",bg="white",command=enregistrer)
    B1.place(x=20,y=70)
    fen.protocol("WM_DELETE_WINDOW",quitter)
    fen.mainloop()

    fen.destroy()

```

```

#Affichage des meilleurs scores
fen1 =Tk()
fen1.title('MEILLEURS SCORES.')
fen1.geometry('450x300')
L=Label(fen1,text="Entrez vous dans la légende ?",width=25,fg="blue",bg="yellow")
L.place(x=20,y=20)
L1=Label(fen1,text="Surnom",width=10,fg="blue",bg="yellow")
L1.place(x=210,y=20)
L2=Label(fen1,text="temps",width=8,fg="blue",bg="yellow")
L2.place(x=300,y=20)
L1.config(text=texteSurnom)
L2.config(text=score)
L3=Label(fen1,text="TOP Scores",width=22,fg="blue",bg="yellow")
L3.place(x=20,y=50)
L4=Label(fen1,text="Vos meilleurs scores.",width=22,fg="blue",bg="yellow")
L4.place(x=250,y=50)
#Affichage des Top scores
S1=Label(fen1,text="1",width=3,fg="yellow",bg="blue")
S1.place(x=20,y=80)
S2=Label(fen1,text="2",width=3,fg="yellow",bg="blue")
S2.place(x=20,y=110)
N1=Label(fen1,text="",width=12,fg="yellow",bg="blue")
N1.place(x=50,y=80)
N2=Label(fen1,text="",width=12,fg="yellow",bg="blue")
N2.place(x=50,y=110)
T1=Label(fen1,text="",width=4,fg="yellow",bg="blue")
T1.place(x=143,y=80)
T2=Label(fen1,text="",width=4,fg="yellow",bg="blue")
T2.place(x=143,y=110)

#Affichage des meilleurs scores du joueur
S11=Label(fen1,text="1",width=3,fg="yellow",bg="blue")
S11.place(x=250,y=80)
S12=Label(fen1,text="2",width=3,fg="yellow",bg="blue")
S12.place(x=250,y=110)
N11=Label(fen1,text="",width=12,fg="yellow",bg="blue")
N11.place(x=280,y=80)
N12=Label(fen1,text="",width=12,fg="yellow",bg="blue")
N12.place(x=280,y=110)
T11=Label(fen1,text="",width=4,fg="yellow",bg="blue")
T11.place(x=373,y=80)
T12=Label(fen1,text="",width=4,fg="yellow",bg="blue")
T12.place(x=373,y=110)
#Recherche des meilleurs scores
connexion=sqlite3.connect('basedonneesjeu.db')
curseur=connexion.cursor()
curseur.execute('SELECT surnom,score FROM PERSONNE ORDER BY score DESC limit 6')
connexion.commit() #Obligatoire pour que la requête SQL soit effectuée
liste=curseur.fetchall() #Les réponses sont stockées dans liste
n=len(liste)
if n>0:
    N1.config(text=liste[0][0])
    T1.config(text=liste[0][1])
if n>1:
    N2.config(text=liste[1][0])
    T2.config(text=liste[1][1])
curseur.close()
connexion.close()

```

```

#Recherche des meilleurs scores du joueur
connexion=sqlite3.connect('basedonneesjeu.db')
curseur=connexion.cursor()
recherche=(texteSurnom,0)
curseur.execute('SELECT surnom,score FROM PERSONNE WHERE surnom= ? AND score> ? ORDER BY score DESC LIMIT 2',recherche)
connexion.commit()      #Obligatoire pour que la requête SQL soit effectuée
liste=curseur.fetchall() #Les réponses sont stockées dans liste
n=len(liste)
if n>0:
    N11.config(text=liste[0][0])
    T11.config(text=liste[0][1])
if n>1:
    N12.config(text=liste[1][0])
    T12.config(text=liste[1][1])
curseur.close()
connexion.close()
B2=Button(fen1,text="quitter",fg="green",bg="white",command=fen.quit)
B2.place(x=180,y=260)
fen1.mainloop()

fen1.destroy()

```

Barème

Etape 1 : création du monde et des contacts	3 points
Etape 2 : Plateformes horizontales et verticales .	2 points
Etape 3 : Bonus animé qui disparaît au contact du héros	2 points
Etape 4 : Ennemis fixes + ennemis qui se déplacent.	2 point
Etape 5 : un des ennemis doit tirer.	2 points
Etape 6 : Faire tirer le héros.	2 points
Etape 7 : Faire afficher le score ...	1 point
Etape 8 : Animations de fin de partie.	2 points
Etape 9 : création d'un second niveau.	2 points
Etape 10 : enregistrer le score et faire afficher les meilleurs scores.	2 points

Le travail devra être rendu au plus tard le dimanche 2 juin. Le fichier python, ainsi que les fichiers sons et les différentes images devront être regroupés dans un répertoire.

Votre jeu doit pouvoir fonctionner sur n'importe quel ordinateur muni de edupython. Attention à vos chemins d'accès pour charger les images et les sons.