



**KAOUTAR  
JABRI**

# **Compte rendu du projet space game**



---

**BTS 2021-2023**

# Sommaire

---

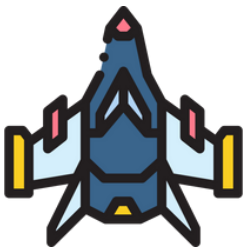
- 01.** Introduction
- 02.** Le résultat attendu du projet
- 03.** Présentation et explication  
du code
- 04.** Conclusion

## Introduction

Le projet space game est un jeu vidéo en 2D fait en python à l'aide de pygame une Framework python qui permet de créer des jeux vidéos

## Résultat attendu

Le but du projet est de créer un jeu vidéos ou le joueur principale est un vaisseau spatial qui devra éviter et attaquer des ennemies.



Le joueur principale



L'eclair



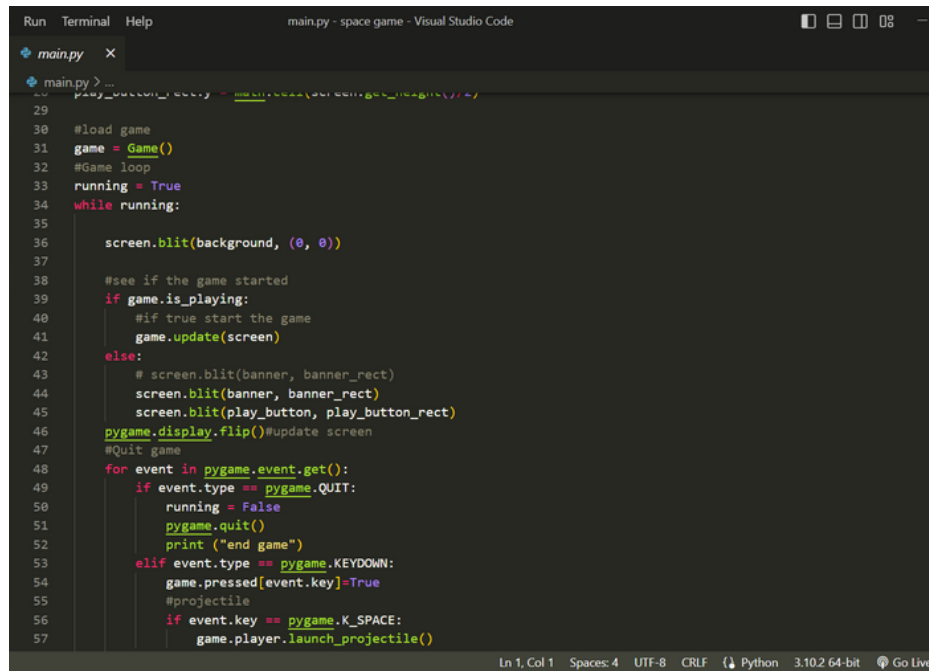
La flèche



Le diamon

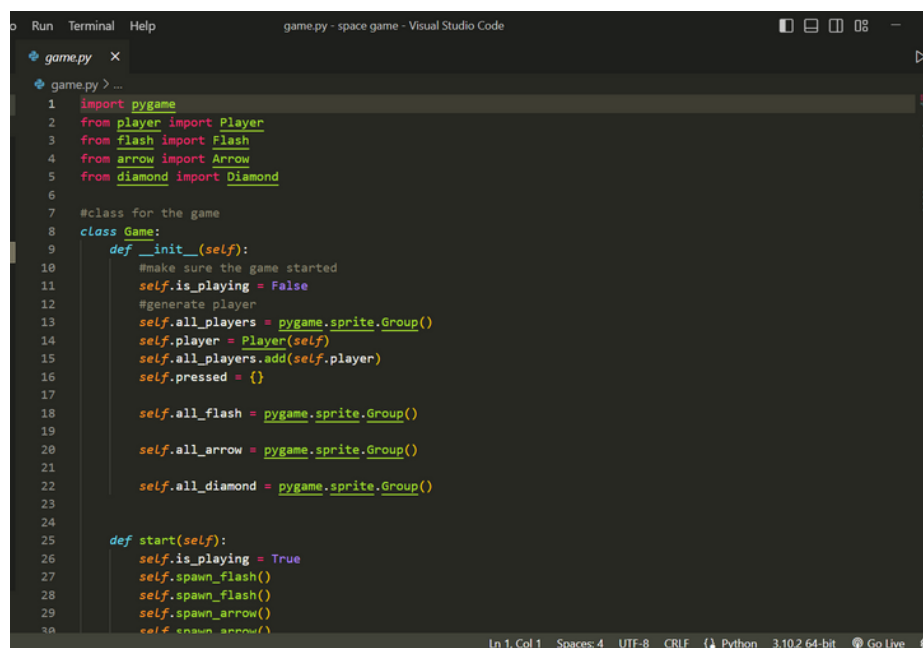
# Presentation et explication du code

Le premier fichier est le fichier main.py qui s'occupe du lancement du jeu



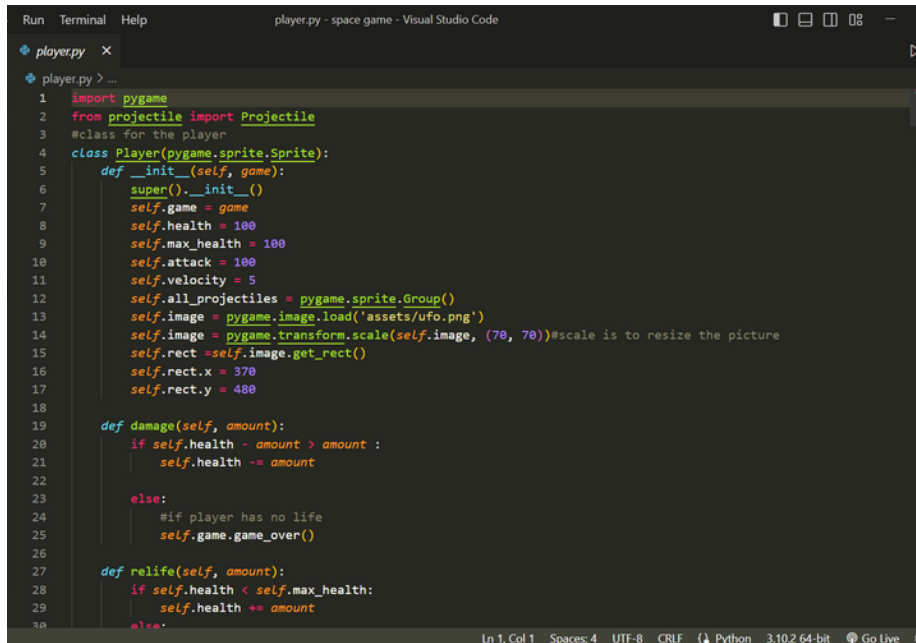
```
29
30 #load game
31 game = Game()
32 #Game loop
33 running = True
34 while running:
35
36     screen.blit(background, (0, 0))
37
38     #see if the game started
39     if game.is_playing:
40         #if true start the game
41         game.update(screen)
42     else:
43         # screen.blit(banner, banner_rect)
44         screen.blit(banner, banner_rect)
45         screen.blit(play_button, play_button_rect)
46         pygame.display.flip()#update screen
47     #Quit game
48     for event in pygame.event.get():
49         if event.type == pygame.QUIT:
50             running = False
51             pygame.quit()
52             print ("end game")
53         elif event.type == pygame.KEYDOWN:
54             game.pressed[event.key]=True
55             #projectile
56             if event.key == pygame.K_SPACE:
57                 game.player.launch_projectile()
```

le fichier game.py contient la classe Game qui se charge de contenir tout les composent du jeu



```
1 import pygame
2 from player import Player
3 from flash import Flash
4 from arrow import Arrow
5 from diamond import Diamond
6
7 #class for the game
8 class Game:
9     def __init__(self):
10         #make sure the game started
11         self.is_playing = False
12         #generate player
13         self.all_players = pygame.sprite.Group()
14         self.player = Player(self)
15         self.all_players.add(self.player)
16         self.pressed = {}
17
18         self.all_flash = pygame.sprite.Group()
19
20         self.all_arrow = pygame.sprite.Group()
21
22         self.all_diamond = pygame.sprite.Group()
23
24     def start(self):
25         self.is_playing = True
26         self.spawn_flash()
27         self.spawn_flash()
28         self.spawn_arrow()
29         self.spawn_arrow()
```

La classe Player contient tout les caractéristiques du joueur principale: l'image du joueur, les points de vie, sa force et sa vitesse



```
Run Terminal Help
player.py - space game - Visual Studio Code

player.py x
player.py > ...
1 import pygame
2 from projectile import Projectile
3 #class for the player
4 class Player(pygame.sprite.Sprite):
5     def __init__(self, game):
6         super().__init__()
7         self.game = game
8         self.health = 100
9         self.max_health = 100
10        self.attack = 100
11        self.velocity = 5
12        self.all_projectiles = pygame.sprite.Group()
13        self.image = pygame.image.load('assets/ufo.png')
14        self.image = pygame.transform.scale(self.image, (70, 70)) #scale is to resize the picture
15        self.rect = self.image.get_rect()
16        self.rect.x = 370
17        self.rect.y = 480
18
19    def damage(self, amount):
20        if self.health - amount > amount :
21            self.health -= amount
22
23        else:
24            #if player has no life
25            self.game.game_over()
26
27    def relife(self, amount):
28        if self.health < self.max_health:
29            self.health += amount
30        else:
31            pass
```

les autres composants du jeu c'est les classes Arrow et Flash, ceux qui représentent les ennemies du joueur principale



```
#create the monster
class Arrow(pygame.sprite.Sprite):
    def __init__(self, game):
        super().__init__()
        self.game = game
        self.health = 100
        self.max_health = 100
        self.attack = 10
        self.image = pygame.image.load('assets/arrow.png')
        self.image = pygame.transform.scale(self.image, (50, 50))
        self.rect = self.image.get_rect()
        self.rect.x = random.randint(20, 780)
        self.rect.y = random.randint(-50, 20)

        self.velocity = 2
```



```
#create the monster
class Flash(pygame.sprite.Sprite):
    def __init__(self, game):
        super().__init__()
        self.game = game
        self.health = 100
        self.max_health = 100
        self.attack = 10
        self.image = pygame.image.load('assets/flash.png')
        self.image = pygame.transform.scale(self.image, (50, 50))
        self.rect = self.image.get_rect()
        self.rect.x = random.randint(20, 780)
        self.rect.y = random.randint(-50, 20)

        self.velocity = 3
```

Le diamant est pour donner plus de vie au joueur principale



```
class Diamond(pygame.sprite.Sprite):
    def __init__(self, game):
        super().__init__()
        self.game = game
        self.health = 100
        self.max_health = 100
        self.relife = 20
        self.image = pygame.image.load('assets/diamond.png')
        self.image = pygame.transform.scale(self.image, (50, 50))
        self.rect = self.image.get_rect()
        self.rect.x = random.randint(50, 750)
        self.rect.y = random.randint(-50, 20)

        self.velocity = 1
```

# Conclusion

---

Le résultat final est le suivant :

