



**FILIERE : BTS Services informatiques aux organisations**

**- SIO SLAM -**

**RAPPORT DE FIN DE STAGE**

**Acquérir des compétences de développement des jeux  
vidéo.**

**Réalisé par : MOUBTAKIR Mouad**

**Encadré par : Mr. Zied LOUKIL**

*Année Universitaire : 2021 / 2022*

# Table des matières

## **Chapitre 1 : Contexte général du projet**

I.	Présentation de l'organisme d'accueil .....	5
II.	Présentation des modules .....	5
1.	Turtle : .....	5
2.	Pygame : .....	5
III.	Déroulement prévisionnel .....	5

## **Chapitre 2 : Présentation des plateformes Turtle et Pygame**

I.	Exemples d'application .....	7
II.	2ème Application : .....	9
<b>Conclusion .....</b>		<b>11</b>

## ***REMERCIEMENTS***

Au terme de ce stage, je tiens à remercier Mr **Zied LOUKIL** pour son aimable accueil qu'il m'a réservé.

Je tiens à le remercier, tout particulièrement, pour l'intérêt qu'il a porté à mon projet, pour ses conseils fructueux et pour m'avoir fourni tous les moyens nécessaires au bon déroulement de mon stage.

J'exprime tout particulièrement ma reconnaissance à **Mr Anis BENAMMAR**, le coordonnateur de la filière BTS Services informatiques aux organisations, qui a fait preuve de responsabilité et engagement.

Je remercie vivement le corps professoral de l'Ecole ESTYA pour ses efforts visant le maintien et le développement de la qualité de la formation au sein de notre école.

Je remercie toute personne ayant contribué de près ou de loin à la réalisation de ce travail.

## *Liste des abréviations*

**ERP** : Enterprise Resource Planning

**JSON**: JavaScript Object Notation

**Turtle** : Le module `Turtle` est une version étendue du module homonyme appartenant à la distribution standard de Python jusqu'à la version 2.5.

**Pygame** : Simple DirectMedia Layer

## Chapitre 1

### *Contexte général du projet*

Au cours de ce chapitre, j'introduis le contexte général ou se situe mon projet.

Je commence par une présentation de turtle et pygame j'enchaîne ensuite par la présentation du sujet de mes projet et leur principaux axes.

# I. Présentation de l'organisme d'accueil

La société ELITE LABS est une société qui intervient dans le marché des nouvelles technologies. Elle propose à ses clients des services autour des technologies suivantes :

- Intégration de jeux turtle
- Intégration de jeux pygame

Son portefeuille est essentiellement composé des clients en France.

## II. Présentation des modules

### 1. Turtle :

*Turtle* permet d'utiliser des primitives graphiques en utilisant un style de programmation orienté objet ou procédural. Du fait qu'il utilise la bibliothèque graphique [tkinter](#), *Turtle* a besoin d'une version de python implémentant *Tk*.

### 2. Pygame :

pygame est un wrapper Python pour la [bibliothèque SDL](#), qui signifie **Simple DirectMedia Layer**. SDL fournit un accès multiplateforme aux composants matériels multimédias sous-jacents de votre système, tels que le son, la vidéo, la souris, le clavier et le joystick. pygame a commencé sa vie en remplacement du [projet bloqué PySDL](#).

## III. Déroulement prévisionnel

Liste des tâches	Délai
Prise de connaissance avec le contexte et installation des logiciels	7 jours
Trouver le sujet auquel se baser pour développer l'application	10 jours

Développement de première application	8 jours
Développement de la deuxième application	8 jours

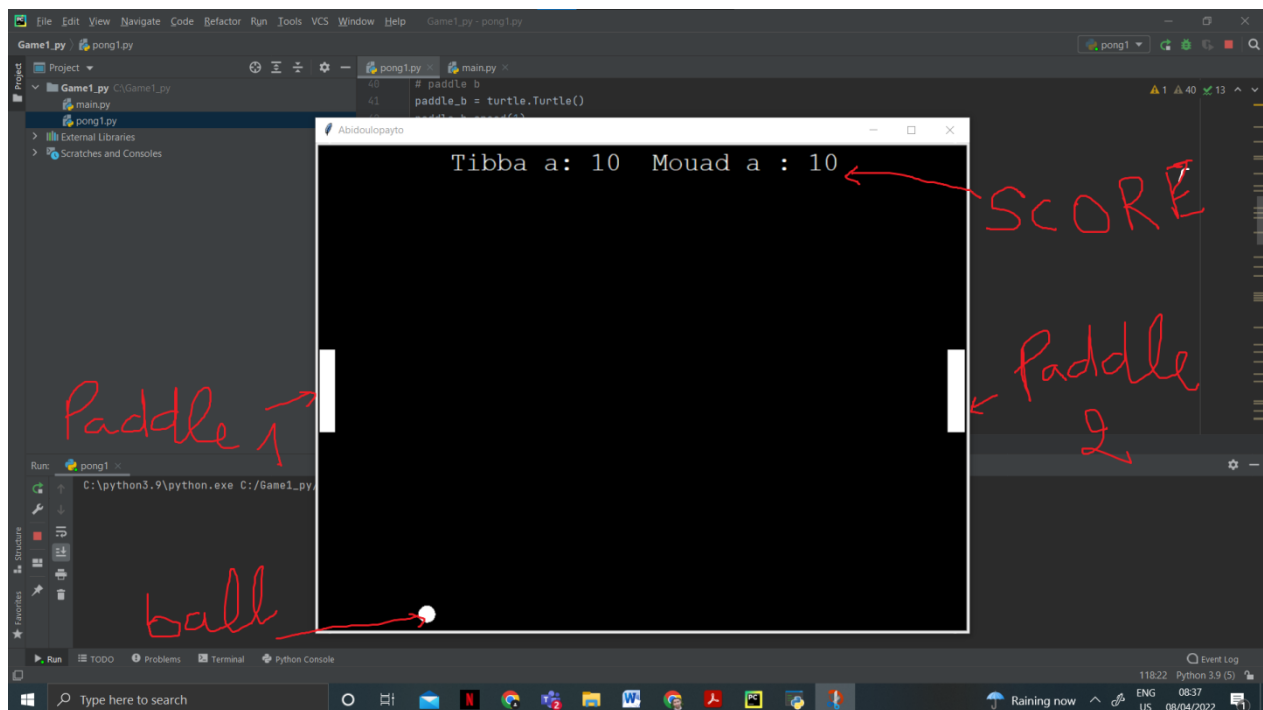
## Chapitre 2

### *Présentation des plateformes Turtle et Pygame*

Dans ce chapitre on présentera la plateforme Turtle, pygame qui a été utilisée pour développer notre application.

#### **I. Exemples d'application**

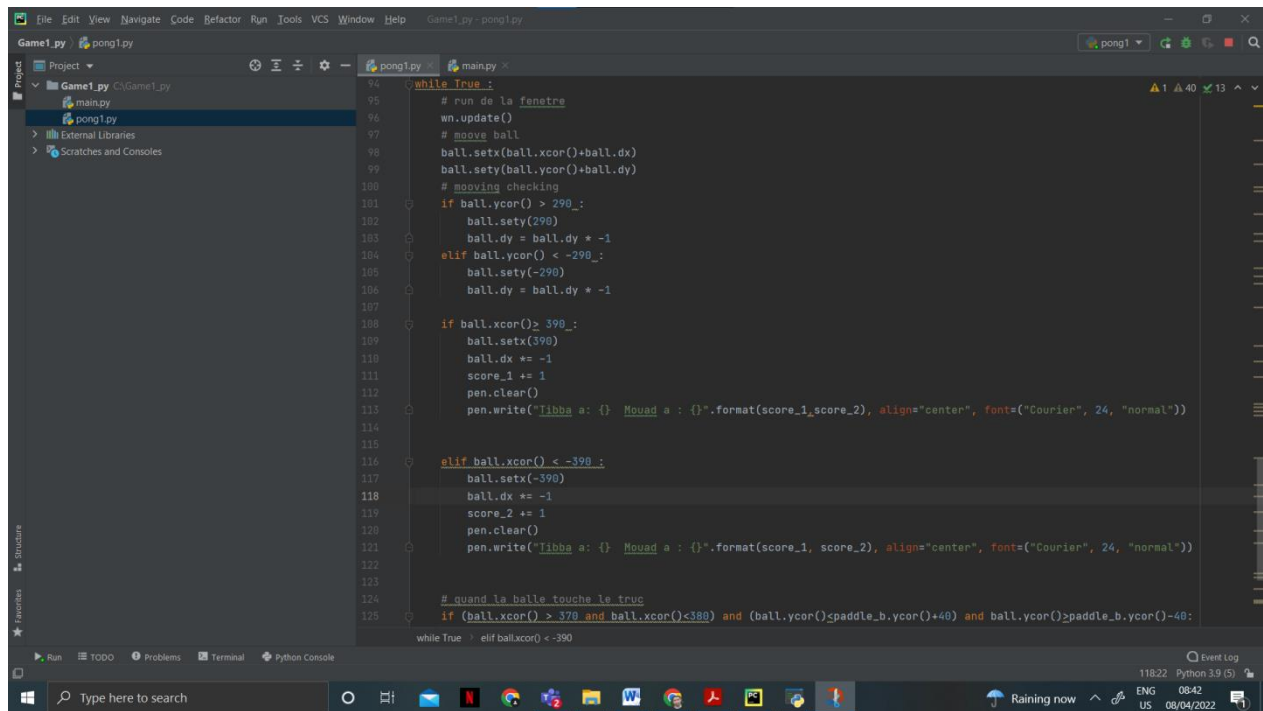
Dans le site officiel du pycharm, on peut accéder aux exemples ou démos d'applications turtle, pygame. On peut aussi télécharger le code de ces applications



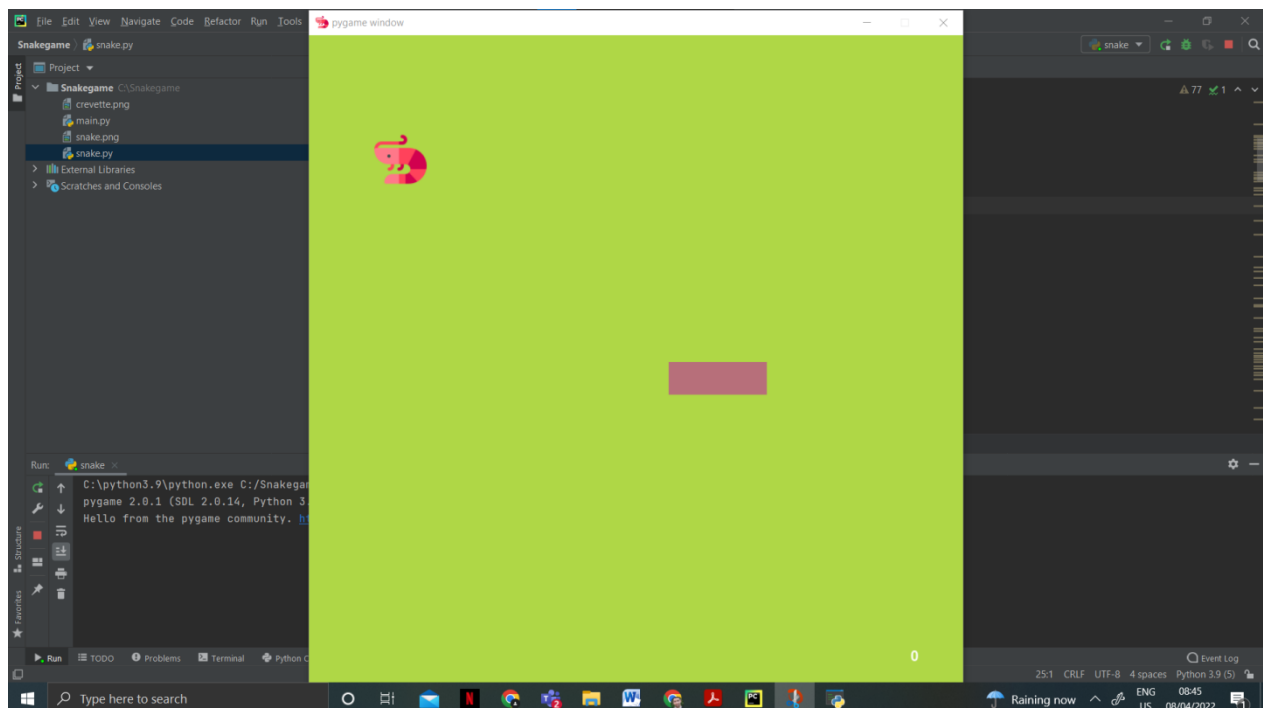
```
1 #
2 import turtle
3 import winsound
4
5 score_1 = 0
6 score_2 = 0
7
8
9 wn = turtle.Screen()
10 wn.title("Abidouloupayto")
11 wn.bgcolor("black")
12 wn.setup(width=800,height=600)
13 wn.tracer(0)
14
15
16
17
18 #pen(player)
19 pen = turtle.Turtle()
20 pen.speed(0)
21 pen.color("white")
22 pen.penup()
23 pen.hideturtle()
24 pen.goto(0,260)
25 pen.write("Iibba a : 0 Mouad a : 0", align="center", font=("Courier", 24, "normal"))
26
27
28
29
30
31 # paddle a
32 paddle_a = turtle.Turtle()
33 while True : elif ball.xcor() < -390
```

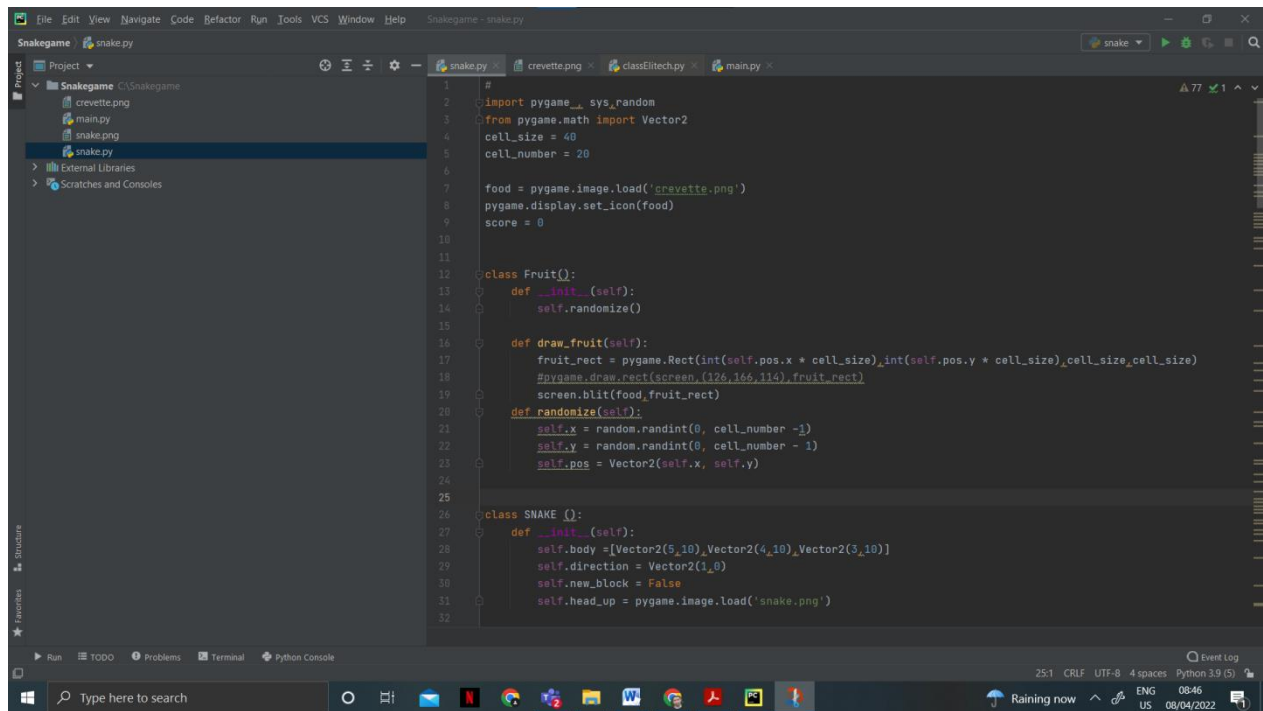
```
49 #ball
50 ball = turtle.Turtle()
51 ball.speed(0)
52 ball.shape("circle")
53 ball.color("white")
54 ball.penup()
55 ball.goto(0,0)
56 ball.dx = 0.3
57 ball.dy = 0.3
58
59
60 def paddle_a_up():
61     x = paddle_a.ycor()
62     x += 20
63     paddle_a.sety(x)
64
65 def paddle_a_down():
66     x = paddle_a.ycor()
67     x -= 20
68     paddle_a.sety(x)
69
70 def paddle_b_up():
71     x = paddle_b.ycor()
72     x += 20
73     paddle_b.sety(x)
74
75 def paddle_b_down():
76     x = paddle_b.ycor()
77     x -= 20
78     paddle_b.sety(x)
79
80
81 wn.listen()
82 wn.onkeypress(paddle_a_up,"w")
83
84 while True : elif ball.xcor() < -390
```



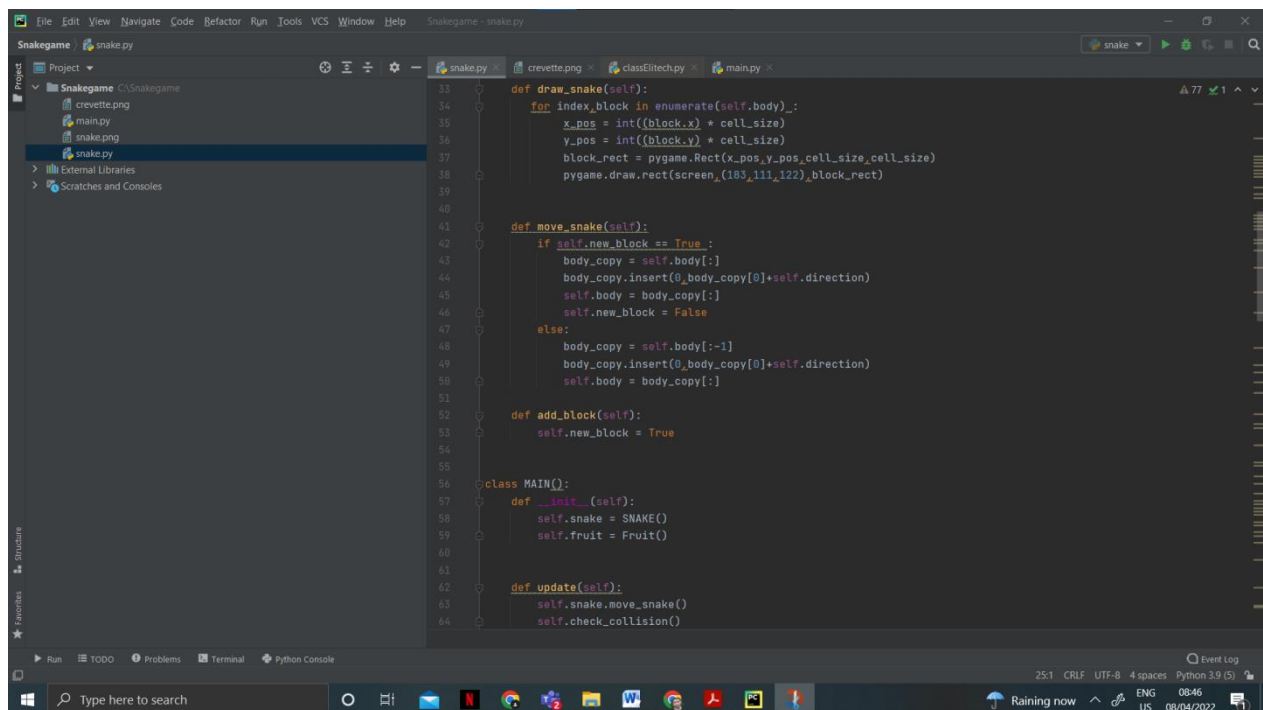


## II. 2ème Application :



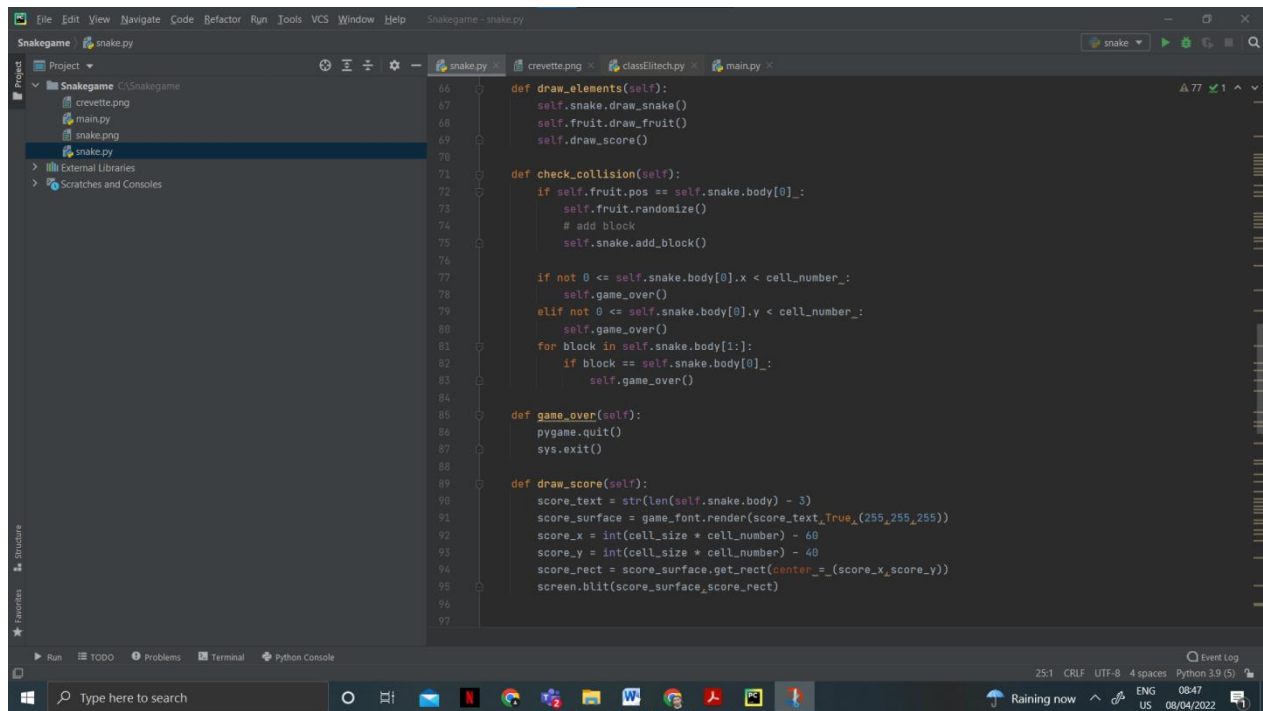


```
1 #
2 import pygame, sys, random
3 from pygame.math import Vector2
4 cell_size = 40
5 cell_number = 20
6
7 food = pygame.image.load('crevette.png')
8 pygame.display.set_icon(food)
9 score = 0
10
11
12 class Fruit():
13     def __init__(self):
14         self.randomize()
15
16     def draw_fruit(self):
17         fruit_rect = pygame.Rect(int(self.pos.x * cell_size), int(self.pos.y * cell_size), cell_size, cell_size)
18         pygame.draw.rect(screen, (126, 166, 114), fruit_rect)
19         screen.blit(food, fruit_rect)
20
21     def randomize(self):
22         self.x = random.randint(0, cell_number - 1)
23         self.y = random.randint(0, cell_number - 1)
24         self.pos = Vector2(self.x, self.y)
25
26 class SNAKE():
27     def __init__(self):
28         self.body = [Vector2(5, 10), Vector2(4, 10), Vector2(3, 10)]
29         self.direction = Vector2(1, 0)
30         self.new_block = False
31         self.head_up = pygame.image.load('snake.png')
32
```

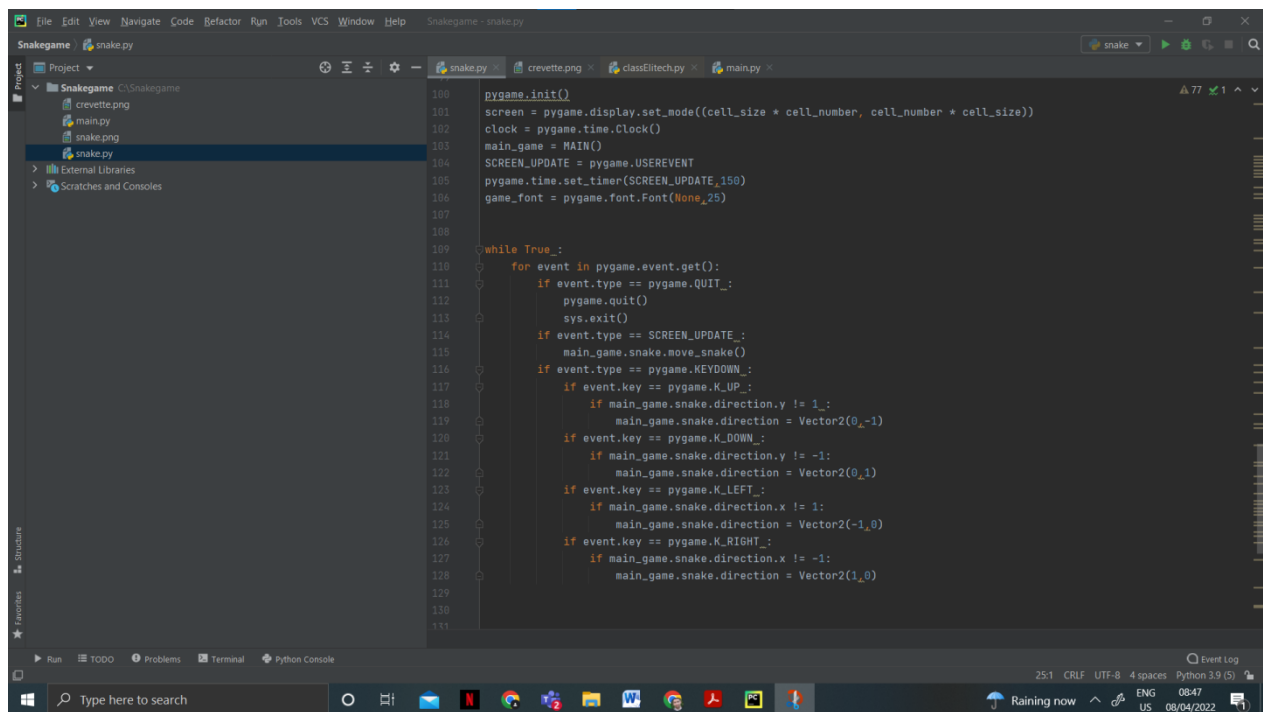


```
33
34 def draw_snake(self):
35     for index_block in enumerate(self.body):
36         x_pos = int((block.x) * cell_size)
37         y_pos = int((block.y) * cell_size)
38         block_rect = pygame.Rect(x_pos, y_pos, cell_size, cell_size)
39         pygame.draw.rect(screen, (183, 111, 122), block_rect)
40
41     def move_snake(self):
42         if self.new_block == True:
43             body_copy = self.body[:]
44             body_copy.insert(0, body_copy[0] + self.direction)
45             self.body = body_copy[:]
46             self.new_block = False
47         else:
48             body_copy = self.body[:-1]
49             body_copy.insert(0, body_copy[0] + self.direction)
50             self.body = body_copy[:]
51
52     def add_block(self):
53         self.new_block = True
54
55
56 class MAIN():
57     def __init__(self):
58         self.snake = SNAKE()
59         self.fruit = Fruit()
60
61
62     def update(self):
63         self.snake.move_snake()
64         self.check_collision()

```



```
46 def draw_elements(self):
47     self.snake.draw_snake()
48     self.fruit.draw_fruit()
49     self.draw_score()
50
51 def check_collision(self):
52     if self.fruit.pos == self.snake.body[0]:
53         self.fruit.randomize()
54         # add block
55         self.snake.add_block()
56
57     if not 0 <= self.snake.body[0].x < cell_number_:
58         self.game_over()
59     elif not 0 <= self.snake.body[0].y < cell_number_:
60         self.game_over()
61     for block in self.snake.body[1:]:
62         if block == self.snake.body[0]:
63             self.game_over()
64
65 def game_over(self):
66     pygame.quit()
67     sys.exit()
68
69 def draw_score(self):
70     score_text = str(len(self.snake.body) - 3)
71     score_surface = game_font.render(score_text, True, (255, 255, 255))
72     score_x = int(cell_size * cell_number) - 60
73     score_y = int(cell_size * cell_number) - 40
74     score_rect = score_surface.get_rect(center=(score_x, score_y))
75     screen.blit(score_surface, score_rect)
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
```



```
100 pygame.init()
101 screen = pygame.display.set_mode((cell_size * cell_number, cell_number * cell_size))
102 clock = pygame.time.Clock()
103 main_game = MAIN()
104 SCREEN_UPDATE = pygame.USEREVENT
105 pygame.time.set_timer(SCREEN_UPDATE, 150)
106 game_font = pygame.font.Font(None, 25)
107
108
109 while True:
110     for event in pygame.event.get():
111         if event.type == pygame.QUIT:
112             pygame.quit()
113             sys.exit()
114         if event.type == SCREEN_UPDATE:
115             main_game.snake.move_snake()
116         if event.type == pygame.KEYDOWN:
117             if event.key == pygame.K_UP:
118                 if main_game.snake.direction.y != 1:
119                     main_game.snake.direction = Vector2(0, -1)
120             if event.key == pygame.K_DOWN:
121                 if main_game.snake.direction.y != -1:
122                     main_game.snake.direction = Vector2(0, 1)
123             if event.key == pygame.K_LEFT:
124                 if main_game.snake.direction.x != 1:
125                     main_game.snake.direction = Vector2(-1, 0)
126             if event.key == pygame.K_RIGHT:
127                 if main_game.snake.direction.x != -1:
128                     main_game.snake.direction = Vector2(1, 0)
129
130
131
```

# Conclusion

Ce stage d'application consistait en la découverte et l'initiation avec le logiciel ainsi que son langage de programmation associés : Python

Sur le plan personnel, cette expérience a été très enrichissante. L'encadrant a été accueillant dès le début du stage. Dans le cadre de la communication en entreprise, j'ai pu échanger, partager et aussi apprécier les avantages que présente le travail au sein d'un établissement

Sur le plan professionnel, ce stage m'a permis d'apprendre de nouvelles connaissances et acquérir de nouvelles compétences, notamment le système de fonctionnement de Python qui est très intéressante ainsi que pygame, turtle qui était un nouveau langage pour moi et encore manier de nouveaux outils malgré les difficultés rencontrées , ainsi que les notions basique pour utiliser la plateforme pycharm dans le but d'une simplification de l'expérience utilisateurs pour les scénarios d'utilisation les plus courants.