

# Rapport TP ANI-IA 4068

Claude Rene EZOM AIA 4

5<sup>th</sup> Mars 2024

## Contents

<b>1</b>	<b>INTRODUCTION</b>	<b>2</b>
<b>2</b>	<b>PRÉSENTATION</b>	<b>2</b>
<b>3</b>	<b>DÉMONSTRATION</b>	<b>2</b>
3.1	Création de la fenêtre et affichage . . . . .	2
3.2	Gestion des évènements et testes . . . . .	3

# 1 INTRODUCTION

La programmation informatique est une discipline indispensable dans la conception logicielle. Pour mieux asseoir ses bases dans cette discipline, il est capital de s'exercer à la programmation des jeux vidéos. Notre étude se focalisera uniquement sur la production d'une application événementielle base sur la bibliothèque Pygame du langage de programmation Python.

## 2 PRÉSENTATION

A travers cette étude, il est question pour nous de concevoir une application permettant de détecter une touche enfoncée sur le clavier d'un ordinateur puis afficher le résultat sur l'écran à travers une fenêtre générée par la bibliothèque PYGAME du langage de programmation Python.

## 3 DÉMONSTRATION

### 3.1 Création de la fenêtre et affichage

```
pygame.init()

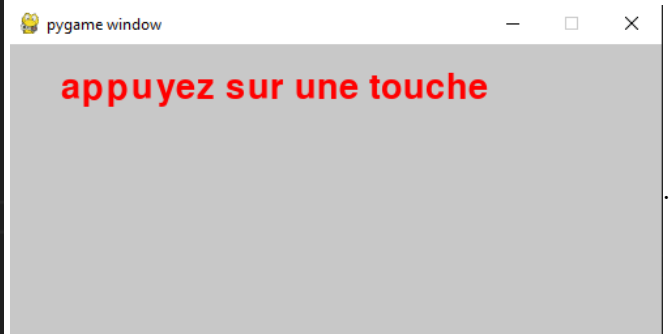
display = pygame.display.set_mode((500, 500))

#font = pygame.font.SysFont(None, 40)

text2 = 'appuyez sur une touche'
font = pygame.font.SysFont(None, 40)
img = font.render(text2, True, (255, 0, 0))
|
rect = img.get_rect()
rect.topleft = (20, 20)

display.fill((200, 200, 200))
display.blit(img, rect)

pygame.display.update()
```



résultat

Expérimentation

### 3.2 Gestion des évènements et testes

```
> stage > cs > 4AIA > SDL > tpvt1.py > ...  
30  
31     for event in pygame.event.get():  
32         if event.type == pygame.QUIT:  
33             running = False  
34  
35  
36  
37     if event.type == pygame.KEYDOWN:  
38  
39         text = 'Vous avez appuyé la touche ' + event.unicode  
40         text2 = 'appuyez sur une touche'  
41         font = pygame.font.SysFont(None, 40)  
42         img = font.render(text, True, (255, 0, 0))  
43         img2 = font.render(text2, True, (255, 0, 0))  
44         rect = img.get_rect()  
45         rect2 = img2.get_rect()  
46         rect2.topleft = (40, 20)  
47         rect2.topleft = (40, 250)  
48         display.fill((200, 200, 200))  
49         display.blit(img, rect)  
50         display.blit(img2, rect2)  
51  
52     pygame.display.update()  
53
```

Expérimentation2



teste