

# RAPPORT

**THEME : CREATION D'UNE APPLICATION  
UTILISANT PYGAME POUR AFFICHER  
EN TEMPS REEL LES DEPLACEMENTS DU  
CURSEUR DE LA SOURIS.**

REALISE PAR :

SIPA Franck Ronald 22P130

Lien github : <https://github.com/franckronaldsipa/HID-detection-Souris>

# Table des matières

<b>1</b>	<b>INTRODUCTION</b>	<b>3</b>
<b>2</b>	<b>Présentation de Pygame</b>	<b>3</b>
2.1	Présentation de Pygame en tant que bibliothèque de développement de jeux . . . . .	3
2.2	Principales fonctionnalités de Pygame . . . . .	3
<b>3</b>	<b>Conception de l'application</b>	<b>3</b>
<b>4</b>	<b>Implémentation de l'application</b>	<b>4</b>
4.1	Présentation du code source . . . . .	4
4.2	Explications . . . . .	4
<b>5</b>	<b>Résultats et fonctionnalités</b>	<b>5</b>
<b>6</b>	<b>Difficultés rencontrées</b>	<b>5</b>
<b>7</b>	<b>CONCLUSION</b>	<b>5</b>

# 1 INTRODUCTION

Dans ce rapport, nous allons présenter un projet portant sur la création d'une application utilisant Pygame pour afficher en temps réel les déplacements de la souris. L'objectif de ce projet était de développer une interface interactive permettant de visualiser en temps réel les interactions d'un utilisateur avec la souris. Dans ce rapport, nous présenterons tout d'abord une introduction à Pygame en tant que bibliothèque de développement de jeux. Nous expliquerons ensuite la conception de l'application, en décrivant son architecture globale et les différentes étapes de développement. Nous aborderons également l'implémentation de l'application, en présentant le code source et en expliquant les principales fonctionnalités mises en place. Enfin, nous discuterons des résultats obtenus, des difficultés rencontrées et des perspectives d'amélioration de l'application. L'objectif de ce document est de fournir une vue d'ensemble complète de la création de cette application utilisant Pygame, en mettant l'accent sur les aspects clés du développement et des fonctionnalités implémentées.

## 2 Présentation de Pygame

### 2.1 Présentation de Pygame en tant que bibliothèque de développement de jeux

Pygame est une bibliothèque de développement de jeux en Python. Elle est open-source et largement documentée, ce qui facilite son utilisation et son apprentissage pour les développeurs Python. Elle offre de nombreuses fonctionnalités, une bonne flexibilité et peut être utilisée pour créer des jeux simples ou plus complexes, ainsi que des applications interactives nécessitant des fonctionnalités graphiques avancées.

### 2.2 Principales fonctionnalités de Pygame

Les principales fonctions de Pygame sont :

- La gestion des fenêtres et des surfaces : Pygame permet de créer des fenêtres graphiques et de manipuler des surfaces pour dessiner des éléments graphiques.
- La gestion des événements : Pygame permet de gérer les événements utilisateur tels que les clics de souris et les pressions de touches clavier.
- Les rendus graphiques : Pygame offre des fonctions pour dessiner des formes géométriques, des images et du texte à l'écran.
- La gestion des événements et des mouvements de la souris.
- La gestion du son : Pygame permet de charger et de jouer des fichiers audios, offrant ainsi la possibilité d'ajouter des effets sonores et de la musique à un jeu ou une application.
- La gestion de la physique : Bien que Pygame ne soit pas une bibliothèque de physique complète, elle offre des fonctionnalités de base pour détecter les collisions entre des objets et gérer la gravité.

## 3 Conception de l'application

L'architecture globale de l'application se compose de plusieurs étapes de développement. Tout d'abord, il faut initialiser Pygame et créer une fenêtre graphique. Ensuite, il faut gérer les événements de la souris pour détecter les clics et les relâchements. Lorsqu'un clic est détecté, les coordonnées du point de départ sont enregistrées. Lorsque le relâchement est détecté, les coordonnées du point d'arrivée sont enregistrées et une ligne est dessinée entre les deux points.

## 4 Implémentation de l'application

### 4.1 Présentation du code source

```
7
8 import pygame
9
10 # Initialisation de Pygame
11 pygame.init()
12
13 # Création de la fenêtre
14 window = pygame.display.set_mode((800, 600))
15
16 # Boucle principale du jeu
17 running = True
18 while running:
19     # Gestion des événements
20     for event in pygame.event.get():
21         # Si l'utilisateur ferme la fenêtre
22         if event.type == pygame.QUIT:
23             running = False
24
25         # Si l'utilisateur clique avec la souris
26         if event.type == pygame.MOUSEBUTTONDOWN:
27             # Récupérer les coordonnées du clic
28             previous_mouse_pos = pygame.mouse.get_pos()
29             print("Clic effectué à la position", previous_mouse_pos)
30
31             # Dessin d'un point vert au point où la souris est cliquée
32             pygame.draw.circle(window, (0, 255, 0), previous_mouse_pos, 3)
33
34         # Si l'utilisateur relâche le clic de souris
35         if event.type == pygame.MOUSEBUTTONUP:
36             # Récupérer les coordonnées du relâchement du clic
37             current_mouse_pos = pygame.mouse.get_pos()
38             print("Clic relâché à la position ", current_mouse_pos)
39
40             # dessin d'un point rouge au point où la souris a été relâché
41             pygame.draw.circle(window, (255, 0, 0), current_mouse_pos, 3)
42
43             if previous_mouse_pos is not None:
44                 # Comparer les coordonnées pour déterminer la direction
45                 if current_mouse_pos[0] > previous_mouse_pos[0]:
46                     print("La souris a été déplacée vers la droite")
47                 elif current_mouse_pos[0] < previous_mouse_pos[0]:
48                     print("La souris a été déplacée vers la gauche")
49
50                 if current_mouse_pos[1] > previous_mouse_pos[1]:
51                     print("La souris a été déplacée vers le bas")
52                 elif current_mouse_pos[1] < previous_mouse_pos[1]:
53                     print("La souris a été déplacée vers le haut")
54
55             pygame.draw.line(window, (255, 255, 255), previous_mouse_pos, current_mouse_pos)
56             previous_mouse_pos = None
57
58         # Mise à jour de l'affichage
59         pygame.display.flip()
60
61 # Fermeture de Pygame
62 pygame.quit()
63
64
```

Le code source de l'application utilise la bibliothèque Pygame pour créer une fenêtre graphique et gérer les événements de la souris. Les principales fonctionnalités implémentées sont la détection des clics et des relâchements de la souris, l'enregistrement des coordonnées des points de départ et d'arrivée, et le dessin d'une ligne entre les deux points.

### 4.2 Explications

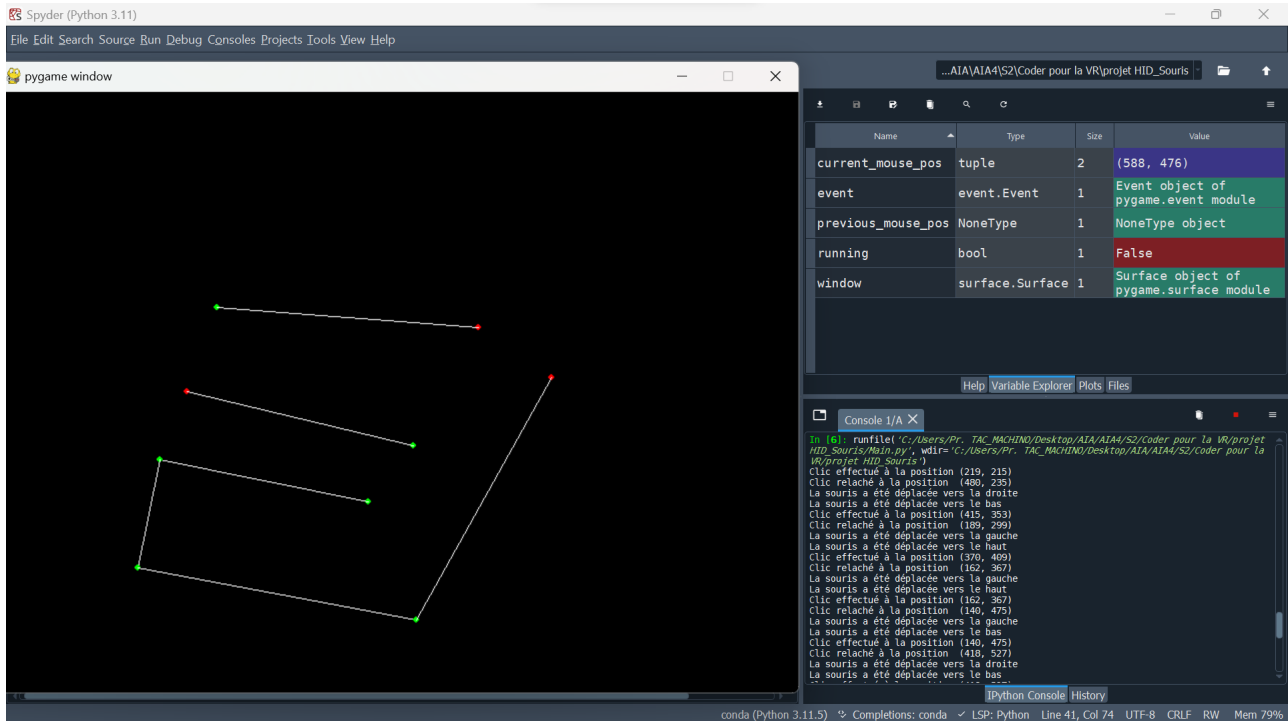
Dans la boucle principale de l'application, nous utilisons la fonction "pygame.event.get()" pour récupérer tous les événements survenus depuis la dernière itération de la boucle. Nous vérifions ensuite si l'événement est de type MOUSEBUTTONDOWN, ce qui indique qu'un a été pressée. Nous récupérons alors les coordonnées du point de départ à l'aide de "event.key" et son nom correspondant en utilisant "pygame.mouse.get-pos()". Ensuite nous vérifions si l'évènement est de type MOUSEBUTTONUP ce qui signifie que le clic a été relâché. Dans ce cas on récupère la position du point d'arriver toujours grâce à la commande "pygame.mouse.get-pos()". Enfin, nous traçons un trait liant le point de départ au point d'arriver ceci grâce à la commande "pygame.draw.line(window, (255, 255, 255), previous-mouse-pos, current-mouse-pos)" qui prend en argument la position de départ de la souris et celle d'arriver.

Une fois le clic relâché, un message informe l'utilisateur des directions prise par le curseur (le curseur a été déplacé vers la gauche, la droite, le bas ou le haut) et ceci es temps réel.

## 5 Résultats et fonctionnalités

L'interface utilisateur de l'application est simple et intuitive. L'utilisateur peut cliquer et relâcher la souris pour dessiner une ligne à l'écran.

La démonstration des fonctionnalités clés de l'application montre la fluidité du dessin de la ligne en temps réel. L'utilisateur peut dessiner des lignes de différentes longueurs et formes en utilisant la souris.



## 6 Difficultés rencontrées

Lors du développement de l'application, plusieurs problèmes techniques ont été rencontrés. Par exemple, la gestion des événements de la souris qui peut être complexe; il a fallu trouver des solutions pour détecter les clics et les relâchements de la souris.

Les solutions adoptées pour résoudre ces problèmes ont été de consulter la documentation de Pygame, de rechercher des exemples de code et de faire des tests et des ajustements pour obtenir le comportement souhaité de l'application.

## 7 CONCLUSION

En conclusion, ce projet Pygame permet à l'utilisateur de dessiner une ligne avec la souris en cliquant et relâchant les boutons de la souris. L'application offre une interface utilisateur simple et intuitive, et les fonctionnalités clés ont été implémentées avec succès. Malgré quelques difficultés techniques, des solutions ont été trouvées pour résoudre les problèmes rencontrés. Ce projet peut être amélioré et approfondi en ajoutant des fonctionnalités supplémentaires, telles que la possibilité de dessiner des formes plus complexes ou d'ajouter des couleurs à la ligne dessinée et plus encore.