



Figure 1: clavier avec identification ID et nom touche .

M.tp_rodolphe

Assouгна Onambele CM

March 2024

1 Introduction

ce code python permet d'identifier chaque bouton appuyer par un user et affiche l'ID du bouton

2 Initialisation de Pygame et des couleurs

Le code commence par initialiser Pygame et définir les couleurs utilisées dans l'interface.

3 Définition de la fenêtre et des dimensions des rectangles

Ensuite, la fenêtre Pygame est créée avec les dimensions spécifiées. Les dimensions des rectangles pour les lettres sont également définies.

4 Création des rectangles et de la police

Les rectangles pour les lettres de A à Z sont créés, ainsi qu'une police pour afficher le texte.

5 Fonction pour afficher le texte

Une fonction `draw_text()` est définie pour afficher du texte sur la fenêtre Pygame.

6 Fonction principale

La fonction principale `main()` est définie. Elle gère les événements et met à jour l'affichage en conséquence.

7 Gestion des événements

Les événements de clavier et de souris sont gérés dans la boucle principale. Lorsqu'une touche est pressée ou qu'un clic de souris est détecté, la fonction `handle_event()` est appelée pour traiter l'événement.

8 Fonction pour traiter les événements

La fonction `handle_event()` vérifie si la touche pressée correspond à une lettre. Si c'est le cas, elle affiche des informations dans la console et change la couleur du rectangle si celui-ci est cliqué avec la souris.

9 Explication du code Python

```
# Importation des modules nécessaires
import pygame
import sys

# Initialisation de Pygame
pygame.init()

# Définition des couleurs
WHITE = (255, 255, 255)
RED = (255, 0, 0)
BLACK = (0, 0, 0)

# Définition des dimensions de la fenêtre
WINDOW_WIDTH = 800
WINDOW_HEIGHT = 600

# Définition de la taille et de l'espacement des rectangles
RECT_WIDTH = 60
RECT_HEIGHT = 60
GAP = 20

# Création de la fenêtre
window = pygame.display.set_mode((WINDOW_WIDTH, WINDOW_HEIGHT))
pygame.display.set_caption("Interface Pygame")

# Création de la police
font = pygame.font.SysFont(None, 40)

# Fonction pour afficher le texte sur la fenêtre
def draw_text(text, font, color, surface, x, y):
```

```

textobj = font.render(text, 1, color)
textrect = textobj.get_rect()
textrect.center = (x, y)
surface.blit(textobj, textrect)

# Fonction principale
def main():
    running = True
    clock = pygame.time.Clock()

    # Cr ation des rectangles pour les lettres de A      Z
    letters = [chr(i) for i in range(65, 91)]
    letter_rects = []
    for i, letter in enumerate(letters):
        x = (i % 7) * (RECT_WIDTH + GAP) + 100
        y = (i // 7) * (RECT_HEIGHT + GAP) + 100
        letter_rects.append(pygame.Rect(x, y, RECT_WIDTH, RECT_HEIGHT))

    while running:
        window.fill(WHITE)

        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                running = False

        # Obtention de l' tat des touches du clavier
        keys = pygame.key.get_pressed()

        # V rification des touches du clavier
        for i, key in enumerate(keys):
            if key:
                if pygame.K_a <= i <= pygame.K_z:
                    handle_event(pygame.K_a + i, letters, letter_rects)

        # Dessiner les rectangles avec les lettres
        for i, rect in enumerate(letter_rects):
            pygame.draw.rect(window, BLACK, rect, 2) # Dessiner les contours
            draw_text(letters[i], font, BLACK, window, rect.x + RECT_WIDTH / 2)

        pygame.display.update()
        clock.tick(30)

    pygame.quit()
    sys.exit()

def handle_event(key, letters, letter_rects):

```

```

# V rifier si la touche press e correspond une lettre
for i, letter_key in enumerate(range(pygame.K_a, pygame.K_z + 1)):
    if key == letter_key:
        print("Touche press e:", letters[i], "(ID:", key, ")")
        for rect in letter_rects:
            if rect.collidepoint(pygame.mouse.get_pos()):
                pygame.draw.rect(window, RED, rect)
        pygame.display.update()
        pygame.time.delay(100) # Pour que la couleur rouge reste visible

if __name__ == "__main__":
    main()

```