

RAPPORT

**THEME : CREATION D'UNE APPLICATION
UTILISANT PYGAME POUR AFFICHER
LES CODES ET LES TOUCHES SAISIES AU
CLAVIER.**

REALISE PAR
EYA ESSAM ETIENNE ERNEST 22P126

PLAN

I. Introduction

II. Présentation de Pygame

1. Présentation de Pygame en tant que bibliothèque de développement de jeux
2. Explication des principales fonctionnalités de Pygame

III. Conception de l'application

1. Description de l'architecture globale de l'application
2. Explication des différentes étapes de développement

IV. Implémentation de l'application

1. Présentation du code source de l'application
2. Explication des principales fonctionnalités implémentées
3. Détails sur la gestion des événements clavier

V. Résultats et fonctionnalités

1. Présentation de l'interface utilisateur de l'application
2. Démonstration des fonctionnalités clés de l'application

VI. Difficultés rencontrées

1. Description des problèmes techniques rencontrés lors du développement
2. Solutions adoptées pour résoudre ces problèmes

VII. Conclusion

I. INTRODUCTION

Le présent rapport porte sur la création d'une application utilisant Pygame pour afficher les codes et les touches saisies au clavier. L'objectif de ce projet était de développer une interface interactive permettant de visualiser en temps réel les codes et les touches pressées par l'utilisateur. Cette application peut être utilisée dans divers contextes, tels que l'apprentissage de la programmation ou la démonstration de l'utilisation des touches du clavier dans un jeu. 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 rapport 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.

II. PRÉSENTATION DE PYGAME

1. Présentation de Pygame en tant que bibliothèque de développement de jeux

Pygame est une bibliothèque de développement de jeux en Python qui offre des fonctionnalités pour créer des graphismes, gérer les entrées utilisateur (clavier, souris), jouer des sons et gérer la physique des objets. Elle est largement utilisée dans l'industrie du jeu vidéo et est appréciée pour sa simplicité d'utilisation et sa portabilité. Pygame est basé sur la bibliothèque SDL (Simple DirectMedia Layer), qui fournit des fonctionnalités de bas niveau pour le rendu graphique, l'audio et les entrées utilisateur. Pygame abstrait ces fonctionnalités de bas niveau et offre une interface plus conviviale pour le développement de jeux.

2. Les principales fonctionnalités de Pygame comprennent :

- a) **Gestion des fenêtres et des surfaces** : Pygame permet de créer des fenêtres graphiques et de manipuler des surfaces pour dessiner des éléments graphiques.
- b) **Gestion des événements** : Pygame permet de gérer les événements utilis-

teur tels que les clics de souris et les pressions de touches clavier.

c) Rendu graphique : Pygame offre des fonctions pour dessiner des formes géométriques, des images et du texte à l'écran.

d) Gestion du son : Pygame permet de charger et de jouer des fichiers audio, offrant ainsi la possibilité d'ajouter des effets sonores et de la musique à un jeu ou une application.

e) 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.

Pygame est une bibliothèque open-source largement documentée, ce qui facilite son utilisation et son apprentissage pour les développeurs Python. Elle offre 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.

III. CONCEPTION DE L'APPLICATION

L'application que nous avons développée utilise Pygame pour afficher les codes et les touches saisies au clavier. Voici une description de l'architecture globale de l'application et des différentes étapes de développement :

1. Architecture globale de l'application :

- o Fenêtre graphique : L'application crée une fenêtre graphique à l'aide de Pygame, où les codes et les touches seront affichés.
- o Gestion des événements : L'application utilise les fonctionnalités de gestion des événements de Pygame pour détecter les pressions de touches clavier.
- o Affichage des codes et des touches : Lorsqu'une touche est pressée, l'application affiche le code correspondant et la touche elle-même à l'écran.

2. Étapes de développement

o Installation de Pygame : Nous avons commencé par installer Pygame en utilisant l'outil de gestion des packages de Python, tel que pip.

o Initialisation de Pygame : Nous avons créé une fenêtre graphique en utilisant la fonction `pygame.init()` et en définissant les dimensions et les paramètres de la fenêtre.

o Gestion des événements clavier : Nous avons utilisé la boucle princi-

pale de l'application pour détecter les événements clavier à l'aide de la fonction `pygame.event.get()`. Nous avons ensuite vérifié si les événements étaient des pressions de touches et avons récupéré les codes et les touches correspondantes.

o Affichage des codes et des touches : Nous avons utilisé la fonction `pygame.font.Font()` pour définir la police d'écriture à utiliser. Ensuite, nous avons utilisé la fonction `pygame.Surface.blit()` pour afficher les codes et les touches à l'écran.

La conception de l'application a été relativement simple, en utilisant les fonctionnalités de base de Pygame pour la création de la fenêtre graphique, la gestion des événements clavier et l'affichage des éléments à l'écran. La simplicité de Pygame a permis de développer rapidement cette application en se concentrant sur les fonctionnalités principales.

IV. IMPLÉMENTATION DE L'APPLICATION

1. Présentation du code source de l'application

L'implémentation de l'application s'est faite en utilisant la bibliothèque Pygame pour créer la fenêtre graphique, gérer les événements clavier et afficher les codes et les touches pressées. Voici une présentation du code source de l'application, des principales fonctionnalités implémentées et de la gestion des événements clavier :

Code source de l'application :

```
1  import pygame
2  from pygame.locals import *
3
4  pygame.init()
5
6  # Définition des dimensions de la fenêtre
7  WINDOW_WIDTH = 500
8  WINDOW_HEIGHT = 400
9
10 # Création de la fenêtre graphique
11 window = pygame.display.set_mode((WINDOW_WIDTH, WINDOW_HEIGHT))
12 pygame.display.set_caption("Affichage des codes et touches clavier")
13
14 # Définition de la police d'écriture
15 font = pygame.font.Font(None, 36)
16
17 # Liste pour stocker les codes et les touches
18 key_list = []
19
20 # Variables pour la gestion du défilement
21 scroll_position = 0
22 scroll_speed = 40 # Vitesse de défilement en pixels par pression de touche
23
24 # Boucle principale de l'application
25 running = True
26 while running:
27     # Gestion des événements
28     for event in pygame.event.get():
29         if event.type == QUIT:
30             running = False
```

```

32 elif event.type == KEYDOWN:
33     # Récupération du code et de la touche pressée
34     key_code = event.key
35     key_name = pygame.key.name(key_code)
36
37     # Ajout du code et de la touche à la liste
38     key_list.append(f"Code: {key_code} Touche: {key_name}")
39
40     # Limitation de la liste à 10 éléments pour éviter un affichage trop long
41     if len(key_list) > 10:
42         key_list.pop(0)
43
44     # Mise à jour de la position de défilement pour afficher la dernière ligne ajoutée
45     scroll_position = max(0, len(key_list) * 40 - WINDOW_HEIGHT)
46
47 elif event.type == KEYUP:
48     if event.key == K_UP:
49         # Défilement vers le haut lors de la relâche de la touche flèche haut
50         scroll_position -= scroll_speed
51     elif event.key == K_DOWN:
52         # Défilement vers le bas lors de la relâche de la touche flèche bas
53         scroll_position += scroll_speed
54
55 # Effacement de la fenêtre
56 window.fill((200, 200, 200))
57
58 # Création d'une surface plus grande que la fenêtre visible
59 surface = pygame.Surface((WINDOW_WIDTH, len(key_list) * 40))
60 surface.fill((200, 200, 200))
61
62 # Affichage des codes et des touches sur la surface
63 y = 0
64 for key_text in key_list:
65     text_surface = font.render(key_text, True, (5, 5, 5))
66     surface.blit(text_surface, (10, y))
67     y += 40
68
69 # Affichage de la portion de la surface correspondant à la fenêtre visible
70 window.blit(surface, (0, -scroll_position))
71
72 # Rafraîchissement de la fenêtre
73 pygame.display.flip()
74
75 pygame.quit()

```

2. Principales fonctionnalités implémentées

- Création d'une fenêtre graphique de dimensions 500x400 pixels.
- Définition d'une police d'écriture pour afficher les codes et les touches.
- Détection des événements clavier à l'aide de la boucle principale de l'application.
- Récupération du code et de la touche pressée lorsque l'événement KEYDOWN est détecté.
- Affichage du code et de la touche à l'écran à l'aide de la fonction `pygame.font.render()` et `pygame.Surface.blit()`.

3. Gestion des événements clavier

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 KEYDOWN, ce qui indique qu'une touche a été pressée. Nous récupérons alors le code de la touche à l'aide de `event.key` et son nom correspondant en utilisant `pygame.key.name()`. Enfin, nous affichons le code et la touche à l'écran en utilisant la fonction `pygame.font.render()` pour créer une surface de texte et la fonction `pygame.Surface.blit()` pour l'afficher à la position souhaitée. L'implémentation de l'application est relativement simple, mais elle permet d'obtenir un affichage en temps réel des codes et des touches pressées au clavier.

V. RÉSULTATS ET FONCTIONNALITÉS

L'application que nous avons développée pour afficher les codes et les touches saisies au clavier a été un succès. Voici les résultats obtenus et les fonctionnalités clés de l'application :

1. Interface utilisateur :

- o L'application crée une fenêtre graphique où les codes et les touches pressées sont affichés en temps réel.
- o Les codes et les touches sont affichés en haut à gauche de la fenêtre, dans une police d'écriture claire et lisible.

2. Affichage des codes et des touches :

- o Lorsqu'une touche est pressée, l'application récupère automatiquement le code correspondant à la touche et affiche les informations à l'écran.
- o Les codes et les touches sont affichés simultanément, ce qui permet à l'utilisateur de visualiser à la fois le code numérique et la représentation textuelle de la touche.

3. Mise à jour en temps réel :

- o L'application met à jour l'affichage à chaque pression de touche, offrant ainsi une expérience interactive en temps réel.
- o À chaque nouvelle pression de touche, les informations précédentes sont effacées et les nouvelles informations sont affichées à leur place.

4. Simplicité d'utilisation :

- o L'application est facile à utiliser, sans nécessiter de connaissances préalables

en programmation ou en utilisation de Pygame.

o Les codes et les touches sont affichés de manière claire et concise, ce qui facilite la compréhension pour les utilisateurs.

Grâce à l'utilisation de Pygame, nous avons pu développer une application simple mais efficace pour afficher les codes et les touches saisies au clavier. L'application offre une interface utilisateur conviviale et une mise à jour en temps réel, ce qui permet à l'utilisateur de visualiser facilement les informations clés liées aux pressions de touches clavier.



resultat de l'application

VI. DIFFICULTÉS RENCONTRÉES

Lors du développement de l'application, nous avons rencontré quelques difficultés spécifiques. Voici les principaux défis auxquels nous avons été confrontés :

1. Gestion des événements clavier : La gestion des événements clavier avec Pygame peut être un peu complexe, car il existe plusieurs types d'événements clavier et il faut les différencier correctement. Nous avons dû comprendre comment détecter l'événement KEYDOWN spécifiquement pour récupérer les pressions de touches.

2. Rafraîchissement de la fenêtre : Pour que les changements apportés à la fenêtre graphique soient visibles, nous devons rafraîchir la fenêtre à chaque itération de la boucle principale. Cependant, si le rafraîchissement est mal géré,

cela peut entraîner des problèmes d’affichage ou de performances. Nous avons dû nous assurer de rafraîchir la fenêtre au bon moment et de manière efficace.

3. Gestion des polices d’écriture : L’affichage des codes et des touches nécessite l’utilisation d’une police d’écriture appropriée. Nous avons dû trouver une police adaptée et nous assurer qu’elle était correctement chargée et utilisée dans l’application.

4. Compatibilité entre les versions de Pygame : Pygame peut être sujet à des mises à jour et des changements de comportement entre les versions. Cela peut entraîner des problèmes de compatibilité si le code est écrit pour une version spécifique de Pygame et est exécuté sur une autre version. Nous avons dû nous assurer que notre code fonctionnait correctement avec la version de Pygame que nous utilisions.

Ces difficultés ont été surmontées grâce à la documentation détaillée de Pygame, à la recherche en ligne et à la persévérance. En fin de compte, nous avons pu résoudre ces problèmes et achever le développement de l’application avec succès.

VII. CONCLUSION

En conclusion, nous avons développé une application utilisant Pygame pour afficher les codes et les touches saisies au clavier. L’application offre une interface utilisateur conviviale, avec un affichage en temps réel des informations clés liées aux pressions de touches. Les principales fonctionnalités de l’application ont été implémentées avec succès, permettant aux utilisateurs de visualiser les codes et les touches pressées de manière claire et concise. Nous avons rencontré quelques difficultés lors du développement, notamment dans la gestion des événements clavier, le rafraîchissement de la fenêtre, la gestion des polices d’écriture et la compatibilité entre les versions de Pygame. Cependant, grâce à la documentation de Pygame et à notre persévérance, nous avons pu surmonter ces obstacles et achever le projet. Cette application peut être utile dans de nombreux scénarios, tels que la création de jeux, l’enregistrement des entrées clavier pour des analyses ou des simulations, ou simplement pour permettre aux utilisateurs de visualiser les codes correspondants aux touches pressées. Elle peut également servir de point de départ pour des développements plus avancés impliquant la gestion des entrées clavier.