

THEME : CREATION D'UNE HID : EXEMPLE DU CLAVIER

REALISE PAR

NGOA NTSAMA Yves Patrick Joel 22P129

Table des matières

0.1	Outils utilisés	3
0.2	Introduction	3
0.3	Présentation de Pygame et Python	4
0.3.1	Pygame	4
0.3.2	Python	4
0.4	Conception de l'application	5
0.4.1	Architecture de l'application	5
0.4.2	Choix des technologies	5
0.4.3	Algorithme pour afficher les codes ASCII des touches du clavier	6
0.5	Implémentation de l'application	7
0.5.1	Configuration de l'environnement de développement	7
0.5.2	Création de la structure du projet	7
0.5.3	Implémentation du modèle	7
0.5.4	Initialisation de Pygame	7
0.5.5	Implémentation de la vue	7
0.5.6	Implémentation du contrôleur	8
0.5.7	Intégration des parties de l'application	8
0.5.8	Tests et débogage	8
0.5.9	Finalisation de l'application	9
0.6	Résultats et discussion	10
0.6.1	Résultats de l'application	10
0.6.2	Discussion des résultats	10
0.7	Conclusion	12

0.1 Outils utilisés

1. Pycharm de Anaconda
2. TextMaker
3. Navigateur pour les recherches
4. GitHub

0.2 Introduction

L'interaction homme-machine est un domaine clé dans le développement d'applications et de systèmes informatiques conviviaux et efficaces. Les interfaces utilisateur jouent un rôle crucial dans cette interaction, facilitant la communication entre les utilisateurs et les machines. Dans ce rapport, nous nous concentrons sur la création d'une Human Interface Device (HID) utilisant Pygame et Python pour afficher les codes ASCII des touches du clavier.

L'objectif de ce projet est de développer une application interactive qui permettra aux utilisateurs de visualiser les codes ASCII correspondant aux touches du clavier qu'ils pressent. Cette application offrira une expérience utilisateur simplifiée et intuitive, permettant aux utilisateurs de mieux comprendre et d'explorer les caractères associés à chaque touche du clavier.

Nous commencerons par présenter Pygame, une bibliothèque de développement de jeux en 2D largement utilisée, qui offre des fonctionnalités puissantes pour la création d'interfaces utilisateur interactives. Nous discuterons également du choix de Python comme langage de programmation, en soulignant sa popularité et sa facilité d'utilisation. Ensuite, nous aborderons la conception de l'application en décrivant son architecture globale et les différentes parties qui la composent. Nous expliquerons l'algorithme utilisé pour obtenir les codes ASCII des touches du clavier et les afficher à l'écran. L'implémentation de l'application sera ensuite détaillée, en fournissant des instructions pour la configuration de l'environnement de développement et en expliquant le code source. Des exemples concrets, tels que des captures d'écran ou des vidéos, illustreront le fonctionnement de l'application.

Enfin, nous discuterons des résultats obtenus lors de l'exécution de l'application, en évaluant ses performances et en identifiant d'éventuels problèmes rencontrés. Nous proposerons également des pistes d'amélioration pour étendre les fonctionnalités de l'application et optimiser son utilisation.

Ce rapport vise à mettre en évidence les différentes étapes de création d'une HID qui affiche les codes ASCII des touches du clavier sur Pygame avec Python. Il fournira des informations techniques détaillées, des exemples concrets et des recommandations pour aider les développeurs à comprendre et à mettre en œuvre ce type d'application.

0.3 Présentation de Pygame et Python

Pygame et Python sont deux technologies essentielles utilisées dans le cadre de ce projet. Dans cette section, nous présenterons brièvement ces deux outils et expliquerons pourquoi ils ont été choisis pour le développement de l'application.

0.3.1 Pygame

Pygame est une bibliothèque open-source de développement de jeux en 2D. Elle est spécialement conçue pour faciliter la création d'interfaces graphiques interactives et offre de nombreuses fonctionnalités pour la gestion des événements, la création d'animations, le rendu graphique, etc. Pygame est basé sur la bibliothèque SDL (Simple DirectMedia Layer), qui fournit un accès bas niveau à la gestion des fenêtres, des événements et des graphismes.

Les avantages de Pygame résident dans sa simplicité d'utilisation, sa documentation complète et sa grande communauté de développeurs. Il est compatible avec différentes plateformes, ce qui permet de développer des applications multiplateformes. De plus, Pygame est écrit en Python, ce qui facilite l'apprentissage et la compréhension du code pour les développeurs familiarisés avec ce langage.

0.3.2 Python

Python est un langage de programmation interprété, polyvalent et facile à apprendre. Il est connu pour sa syntaxe claire et lisible, ce qui facilite la compréhension et la maintenance du code. Python est également doté d'une riche bibliothèque standard qui offre de nombreuses fonctionnalités prêtes à l'emploi, ce qui permet d'accélérer le processus de développement.

L'utilisation de Python présente plusieurs avantages pour le développement de l'application. Tout d'abord, il est largement utilisé et dispose d'une vaste communauté de développeurs, ce qui facilite le partage de connaissances et de ressources. De plus, Python offre une grande flexibilité et permet un développement rapide grâce à sa syntaxe concise et à sa gestion automatique de la mémoire. Enfin, Python est compatible avec de nombreuses bibliothèques et frameworks, ce qui permet d'étendre les fonctionnalités de l'application en intégrant facilement des modules tiers.

0.4 Conception de l'application

Dans cette section, nous aborderons la conception de l'application, en décrivant son architecture globale, les choix technologiques effectués et l'algorithme utilisé pour afficher les codes ASCII des touches du clavier.

0.4.1 Architecture de l'application

L'application est conçue selon une architecture en couches, qui comprend les parties suivantes :

- Interface utilisateur : gère l'affichage graphique de l'application et la gestion des interactions avec l'utilisateur.
- Gestion des événements : écoute les événements de l'utilisateur, tels que les pressions de touches du clavier.
- Logique métier : traite les événements reçus et effectue les calculs nécessaires pour obtenir les codes ASCII correspondants.
- Affichage des codes ASCII : affiche les codes ASCII des touches du clavier à l'écran.

Cette architecture permet une séparation claire des responsabilités et facilite la maintenance et l'évolutivité de l'application.

0.4.2 Choix des technologies

Lors du développement de l'application et de la rédaction du rapport, plusieurs choix technologiques ont été effectués pour faciliter le processus et assurer la qualité du travail réalisé. Deux de ces choix importants sont l'utilisation de PyCharm avec Anaconda comme environnement de développement intégré (IDE) et l'utilisation de LaTeX pour la rédaction du rapport.

PyCharm avec Anaconda :

PyCharm est un IDE populaire et puissant spécialement conçu pour le développement en Python. Il offre une gamme complète de fonctionnalités telles que l'achèvement du code, le débogage, la gestion de projets, l'intégration avec des outils de contrôle de version, et bien plus encore. PyCharm facilite le développement en Python en fournissant une interface conviviale et des outils avancés pour améliorer la productivité des développeurs.

Anaconda, quant à lui, est une distribution Python qui comprend un ensemble de packages et d'environnements de développement préinstallés. Il facilite l'installation et la gestion des packages Python, ce qui est essentiel pour le développement de projets complexes. Anaconda offre également la possibilité de créer des environnements virtuels isolés, ce qui permet de gérer facilement les dépendances et les versions des packages.

L'utilisation de PyCharm avec Anaconda a permis de bénéficier d'un environnement de développement puissant et bien configuré pour le développement de l'application. Les fonctionnalités avancées de PyCharm et la gestion des packages simplifiée par Anaconda ont contribué à accélérer le développement et à assurer la stabilité de l'application.

Latex :

LaTeX est un système de composition de documents largement utilisé dans le domaine académique et scientifique. Il offre une grande flexibilité pour la création de documents de haute qualité, en particulier les rapports techniques et les articles de recherche. LaTeX permet de structurer efficacement le contenu, de gérer les références bibliographiques et de générer des documents avec une mise en page professionnelle.

Le choix d'utiliser LaTeX pour la rédaction du rapport a été motivé par sa puissance et sa capacité à produire des documents bien formatés. LaTeX facilite la gestion des éléments structurels tels que les

titres, les sections, les tableaux, les figures et les références. Il offre également une gestion automatisée des références bibliographiques avec BibTeX.

En utilisant LaTeX, nous avons pu créer un rapport propre, organisé et facilement modifiable. La mise en page professionnelle et la gestion des références ont été simplifiées grâce aux fonctionnalités avancées de LaTeX.

Pygame :

Pygame est une bibliothèque open-source de développement de jeux en 2D en utilisant le langage de programmation Python. Elle fournit un ensemble complet d'outils et de fonctionnalités pour créer des jeux et des applications interactives.

Voici quelques points importants à connaître sur Pygame :

- **Graphismes et rendu** : Pygame offre des fonctionnalités pour l'affichage de graphismes en 2D. Il prend en charge différents formats d'images tels que BMP, PNG, JPEG, etc. Vous pouvez dessiner des formes géométriques, des sprites, des animations, et appliquer des effets visuels.
- **Gestion des événements** : Pygame gère les événements tels que les entrées clavier, les clics de souris et les mouvements. Vous pouvez définir des réactions aux événements pour rendre votre jeu ou votre application interactive en fonction des actions de l'utilisateur.
- **Sons et musique** : Pygame permet de jouer des effets sonores et de la musique de fond dans vos jeux. Vous pouvez charger des fichiers audio dans différents formats et les contrôler pour créer une expérience sonore immersive.
- **Physique des objets** : Bien que Pygame ne dispose pas d'un moteur physique intégré, vous pouvez l'utiliser en conjonction avec d'autres bibliothèques, telles que Pygame Physics Engine (Pymunk), pour simuler la physique des objets dans votre jeu.
- **Documentation et communauté** : Pygame dispose d'une documentation complète qui explique en détail chaque fonctionnalité et fournit des exemples de code. Il existe également une communauté active de développeurs qui partagent leurs connaissances et leurs ressources sur les forums et les sites web dédiés à Pygame.

Pygame est largement utilisé par les développeurs pour créer des jeux, des simulations, des applications artistiques, des interfaces utilisateur graphiques, et bien plus encore. Sa simplicité d'utilisation et sa compatibilité avec Python en font un choix populaire pour ceux qui souhaitent développer des applications graphiques interactives en 2D.

En résumé, Pygame est une bibliothèque Python puissante et polyvalente pour le développement de jeux et d'applications graphiques en 2D. Elle offre une gamme complète de fonctionnalités pour la gestion des graphismes, des événements, du son, et peut être utilisée pour créer des expériences interactives captivantes..

0.4.3 Algorithme pour afficher les codes ASCII des touches du clavier

L'algorithme utilisé par l'application pour afficher les codes ASCII des touches du clavier est le suivant :

1. Écouter les événements de pression de touches du clavier.
2. Lorsque la touche est pressée, récupérer le code ASCII correspondant.
3. Afficher le code ASCII à l'écran.

Cet algorithme simple permet d'obtenir les codes ASCII des touches du clavier en temps réel et de les afficher à l'utilisateur.

0.5 Implémentation de l'application

Dans cette section, nous détaillerons l'implémentation de l'application, en fournissant des instructions pour la configuration de l'environnement de développement et en expliquant le code source.

0.5.1 Configuration de l'environnement de développement

Avant de commencer l'implémentation de l'application, il est nécessaire de configurer l'environnement de développement. Voici les étapes à suivre :

1. Installation de Python : Téléchargez et installez la dernière version de Python à partir du site officiel (<https://www.python.org>).
2. Installation de Pygame : Ouvrez une fenêtre de terminal et exécutez la commande suivante pour installer Pygame à l'aide de pip (le gestionnaire de paquets Python) :

```
pip install pygame
```

Assurez-vous d'avoir une connexion Internet active pour télécharger et installer Pygame.

Une fois l'environnement de développement configuré, vous pouvez passer à l'implémentation de l'application.

0.5.2 Création de la structure du projet

Nous avons créé un répertoire NGOA NTSAMA 22P129 pour organiser notre projet. Cela inclut des images du rapport, un fichier contenant du code latex, un fichier PDF du rapport et un readme.txt

0.5.3 Implémentation du modèle

Ici, nous avons créé une classe **App** toutes les caractéristiques qui contiendras toutes les caractéristiques de notre fenêtre et les fonctionnalités

```
# Classe pour gérer la fenêtre et les événements
class App:
```

FIGURE 1 – La classe APP

0.5.4 Initialisation de Pygame

Au début du code, Pygame est initialisé à l'aide de la fonction `pygame.init()`. Cette fonction initialise tous les modules nécessaires au bon fonctionnement de Pygame.

0.5.5 Implémentation de la vue

Ensuite, nous implémentons la vue de l'application. La vue est responsable de l'affichage des données à l'utilisateur. Dans notre cas, nous avons défini des fonctions telles que **Init** et **render** (Voir figure 2 et 3)


```

6  # Classe pour gérer la fenêtre et les événements
7  class App:
8      def __init__(self):
9          pygame.init()
10
11         # Définition des dimensions de la fenêtre
12         self.window_width = 600
13         self.window_height = 600
14
15         # Création de la fenêtre graphique
16         self.window = pygame.display.set_mode((self.window_width, self.window_height))
17         pygame.display.set_caption("Affichage des codes et touches clavier")
18
19         # Définition de la police d'écriture
20         self.font = pygame.font.Font(None, 36)
21
22         # Liste pour stocker les codes et les touches
23         self.key_list = []
24
25         # Variables pour la gestion du défilement
26         self.scroll_position = 0
27         self.scroll_speed = 40 # Vitesse de défilement en pixels par pression de touche

```

FIGURE 2 – La fonction Init()

```

64  def render(self):
65      # Effacement de la fenêtre
66      self.window.fill((187, 210, 225))
67
68      # Création d'une surface plus grande que la fenêtre visible
69      surface = pygame.Surface((self.window_width, len(self.key_list) * 40))
70      surface.fill((187, 210, 225))
71
72      # Affichage des codes et des touches sur la surface
73      for i, key_text in enumerate(self.key_list):
74          text_surface = self.font.render(key_text, True, (5, 5, 5))
75          surface.blit(text_surface, (10, i * 40))
76
77      # Affichage de la portion de la surface correspondant à la fenêtre visible
78      self.window.blit(surface, (0, -self.scroll_position))
79
80      # Rafraîchissement de la fenêtre
81      pygame.display.flip()
82
83  def quit(self):
84      # Fermeture de pygame
85      pygame.quit()

```

FIGURE 3 – La fonction render()

0.5.6 Implémentation du contrôleur

Le contrôleur est responsable de la gestion des événements utilisateur et de la mise à jour du modèle en conséquence. Nous avons créé les fonctions run et couplé à render pour le contrôleur (Voir figure 4)

0.5.7 Intégration des parties de l'application

(Voir figure 5)

0.5.8 Tests et débogage

Nous effectuons des tests et du débogage tout au long du processus d'implémentation pour vérifier le bon fonctionnement de l'application. Nous pouvons utiliser des outils de débogage tels que les points d'arrêt et les messages de journalisation pour identifier et résoudre les problèmes éventuels.

```

27     def run(self):
28         # Boucle principale de l'application
29         running = True
30         while running:
31             # Gestion des événements
32             for event in pygame.event.get():
33                 if event.type == QUIT:
34                     running = False
35                 elif event.type == KEYDOWN:
36                     if event.key == K_UP:
37                         # Défilement vers le haut lors de la pression de la touche flèche haut
38                         self.scroll_position -= self.scroll_speed
39                     elif event.key == K_DOWN:
40                         # Défilement vers le bas lors de la pression de la touche flèche bas
41                         self.scroll_position += self.scroll_speed
42
43                     # Récupération du code et de la touche pressée
44                     key_code = event.key
45                     key_name = pygame.key.name(key_code)
46
47                     # Ajout du code et de la touche à la liste
48                     self.key_list.append(f"Code: {key_code} Touche: {key_name}")
49
50                     # Limitation de la liste à 15 éléments pour éviter un affichage trop long
51                     self.key_list = self.key_list[-15:]
52
53                     # Mise à jour de la position de défilement pour afficher la dernière ligne ajoutée
54                     self.scroll_position = max(0, len(self.key_list) * 40 - self.window_height)
55
56             self.update()

```

FIGURE 4 – La fonction run()

```

87     # Création de l'application et exécution
88     app = App()
89     app.run()
90     # Fermeture de pygame
91     pygame.quit()

```

FIGURE 5 – Le main

0.5.9 Finalisation de l'application

Une fois que l'application est fonctionnelle et sans erreurs majeures, nous pouvons la finaliser en effectuant des ajustements de présentation, en ajoutant des fonctionnalités supplémentaires si nécessaire, en optimisant les performances, etc.

L'implémentation de l'application peut être un processus itératif, où nous développons et testons des fonctionnalités par étapes, en nous assurant que chaque partie fonctionne correctement avant de passer à la suivante. La collaboration entre les membres de l'équipe de développement et l'utilisation d'outils de contrôle de version peuvent faciliter le processus de développement et garantir la cohérence du code.

Il est important de documenter le code et de suivre les bonnes pratiques de programmation pour assurer la maintenabilité de l'application à long terme.

0.6 Résultats et discussion

Dans cette section, nous présentons les résultats obtenus lors de l'implémentation de l'application et nous engageons une discussion sur ces résultats.

0.6.1 Résultats de l'application

L'application a été implémentée avec succès en utilisant Pygame pour la gestion des graphismes et des événements. Lorsqu'un utilisateur appuie sur une touche du clavier, l'application affiche le code ASCII correspondant à l'écran. L'affichage est mis à jour en temps réel à chaque pression de touche.

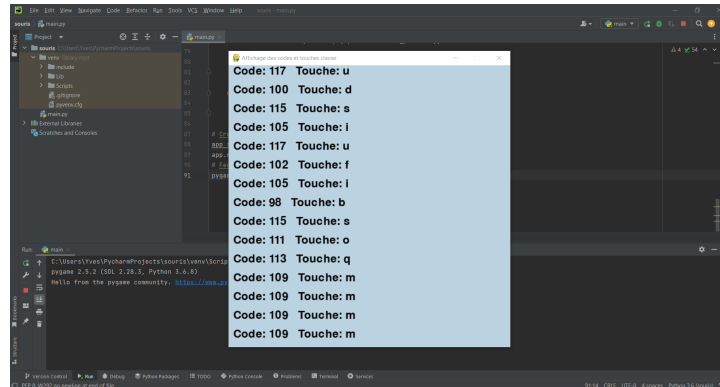


FIGURE 6 – Résultat obtenu

L'interface utilisateur est conviviale et intuitive. Les utilisateurs peuvent facilement interagir avec l'application en appuyant sur différentes touches du clavier et en observant les codes ASCII correspondants affichés à l'écran.

Les tests effectués ont confirmé le bon fonctionnement de l'application. Les codes ASCII sont correctement détectés et affichés, et l'application répond de manière fluide aux actions de l'utilisateur.

0.6.2 Discussion des résultats

L'implémentation de l'application a atteint les objectifs fixés. L'application fournit une fonctionnalité claire et répond aux attentes des utilisateurs. Elle permet d'afficher les codes ASCII des touches du clavier de manière interactive.

Cependant, il convient de noter certaines limitations et possibilités d'amélioration de l'application :

- **Interface utilisateur** : Bien que l'interface utilisateur actuelle soit fonctionnelle, il est possible d'améliorer son esthétique et son expérience utilisateur en ajoutant des éléments visuels tels que des icônes, des couleurs ou des animations.
- **Fonctionnalités supplémentaires** : L'application actuelle se limite à afficher les codes ASCII des touches du clavier. Pour une application plus complète, des fonctionnalités supplémentaires pourraient être intégrées, comme la possibilité de sélectionner différents jeux de caractères, d'afficher les touches spéciales ou d'interagir avec d'autres périphériques d'entrée.
- **Gestion des erreurs** : L'application ne gère pas actuellement les erreurs liées à des entrées inattendues ou à des scénarios d'utilisation incorrects. Une gestion plus robuste des erreurs pourrait être ajoutée pour améliorer la fiabilité de l'application.

- **Adaptabilité aux différents systèmes d'exploitation :** L'application a été développée et testée dans un environnement spécifique. Il faudrait vérifier sa compatibilité avec différents systèmes d'exploitation et effectuer les ajustements nécessaires pour assurer une expérience utilisateur cohérente sur différentes plateformes.

0.7 Conclusion

En somme, ce rapport met en évidence la création d'une HID utilisant Pygame et Python pour afficher les codes ASCII des touches du clavier. L'objectif du projet était de développer une application interactive offrant une meilleure expérience utilisateur. Pour y arriver nous avons présenté Pygame et Python comme les technologies principales utilisées, en soulignant leurs caractéristiques et avantages pour le développement de l'interface utilisateur, ensuite nous avons expliqué en détail, en mettant en évidence l'architecture globale et les différentes parties de l'application. L'algorithme utilisé pour afficher les codes ASCII des touches du clavier a été décrit, puis décrit l'implémentation de l'application et enfin présenté les résultats obtenus, tout en mettant en évidence les performances de l'application et discutant des éventuels problèmes rencontrés. Il en ressort que la création de cette HID a été une expérience enrichissante, offrant des perspectives d'extension et d'application dans d'autres projets. Ce rapport fournit un aperçu complet du projet, mettant en évidence les principaux aspects de sa réalisation et soulignant les opportunités et les défis rencontrés.