

CODER POUR LA VR, L'XR ET L'AR I

THEME : GESTION DES EVENEMENTS CLAVIERS AVEC PYGAME

1) PRESENTATION DE PYGAME

Pygame est une bibliothèque open source écrite en Python, spécifiquement conçue pour le développement de jeux vidéo en 2D. Elle offre une gamme complète d'outils et de fonctionnalités permettant aux développeurs de créer des jeux de manière efficace. Cette bibliothèque offre plusieurs fonctionnalités parmi les quelles :

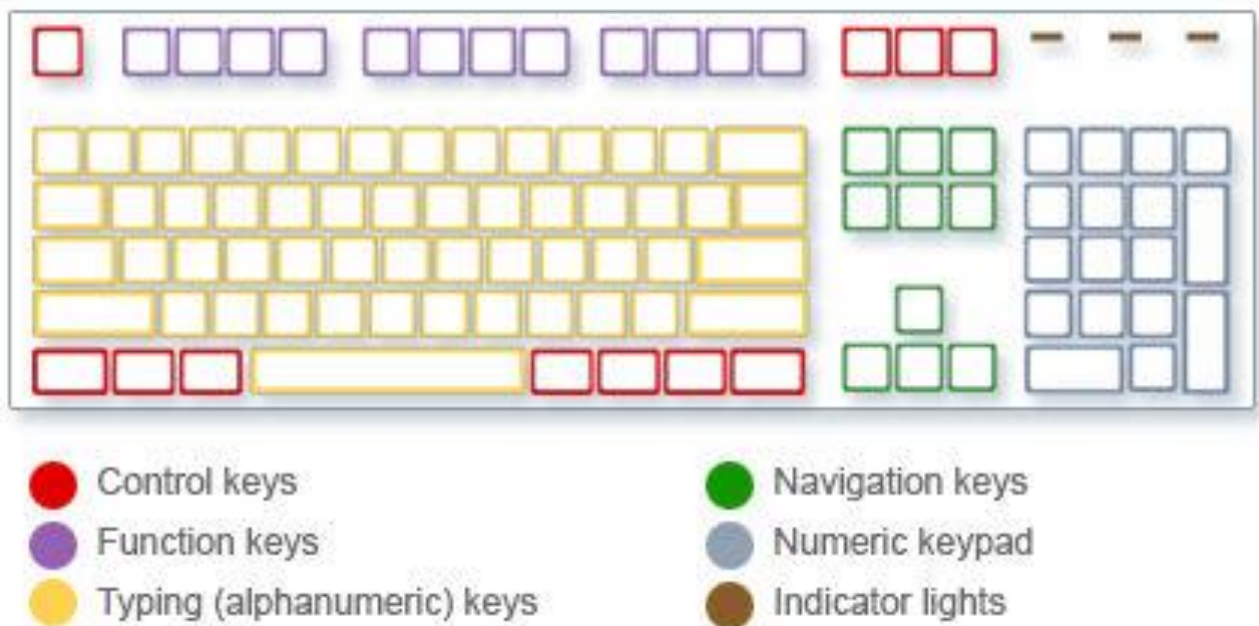
- **Le graphismes et l'affichage** : pygame propose des fonctionnalités pour la gestion des fenêtres, l'affichage d'images, la création de sprites, et la manipulation des surfaces pour permettre un rendu graphique en 2D.
- **Entrées Utilisateur** : Pygame gère les événements utilisateur tels que les mouvements de souris, les clics, les pressions de touches, etc., facilitant ainsi la gestion des interactions entre le joueur et le jeu.
- **Son** : La bibliothèque prend également en charge la gestion du son, permettant aux développeurs d'intégrer des effets sonores et de la musique dans leurs jeux.
- **Gestion du Temps** : Pygame fournit des outils pour gérer le temps dans le jeu, ce qui est essentiel pour créer des animations fluides et des mécanismes de jeu basés sur le temps.

2) Les différentes catégories de touches clavier

Les touches du clavier peuvent être réparties en plusieurs groupes selon la fonction :

- **Touches de saisie (alphanumériques).** Ces touches incluent les mêmes touches associées à des lettres, des chiffres, des marques de ponctuation et des symboles qu'un clavier de machine à écrire.
- **Touches Contrôle.** Ces clés sont utilisées seules ou avec d'autres touches pour effectuer certaines actions. Les touches de contrôle les plus fréquemment utilisées sont Ctrl, Alt, la touche de logo Windows et Échap.
- **Touches de fonction.** Elles permettent d'effectuer des tâches spécifiques. Il s'agit des touches F1 à F12. Le but de ces touches diffère d'un programme à l'autre.
- **Touches de navigation.** Ces touches permettent à l'utilisateur de se déplacer dans des documents ou des pages web, et de modifier du texte. Cela inclut les touches de direction, Début, Fin, Page précédente, Page suivante, Suppr et Inser.
- **Pavé numérique.** Le pavé numérique s'avère pratique pour saisir rapidement des nombres. Les touches de ce pavé sont regroupées au sein d'un bloc, comme une calculatrice ou une machine à calculer classique.

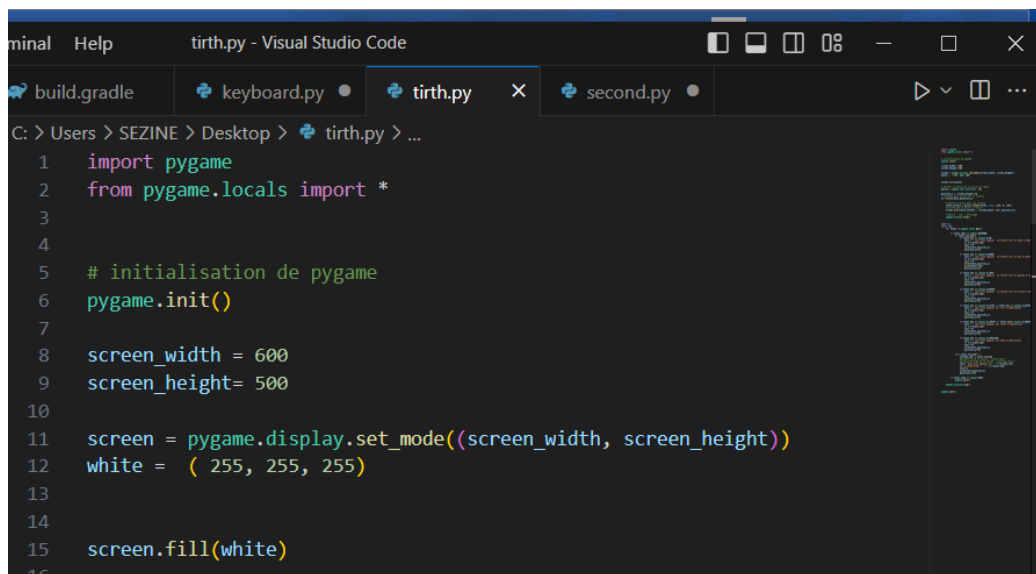
L'illustration suivante indique de quelle manière ces touches sont organisées sur un clavier standard. La disposition de votre clavier peut être différente.



3)Pygame et la gestion des évènements clavier

Pygame propose une gestion d'événements robuste, qui inclut la prise en charge des événements clavier. Cette fonctionnalité permet aux développeurs de créer des jeux interactifs où les actions du joueur sont capturées et traitées en temps réel. Pygame simplifie la gestion des événements clavier en fournissant des outils intuitifs pour détecter les pressions, les relâchements, les combinaisons de touches et l'état actuel du clavier. Cette flexibilité permet aux développeurs de créer des jeux interactifs et réactifs en fonction des actions du joueur sur le clavier.

Tout d'abord la bibliothèque a besoin d'être installée grâce à la commande `pip install pygame` dans le terminal ensuite, on doit Initialiser le module pygame Avant de commencer à gérer les événements clavier, Cela se fait généralement avec la ligne de code `pygame.init()` après avoir importé pygame.

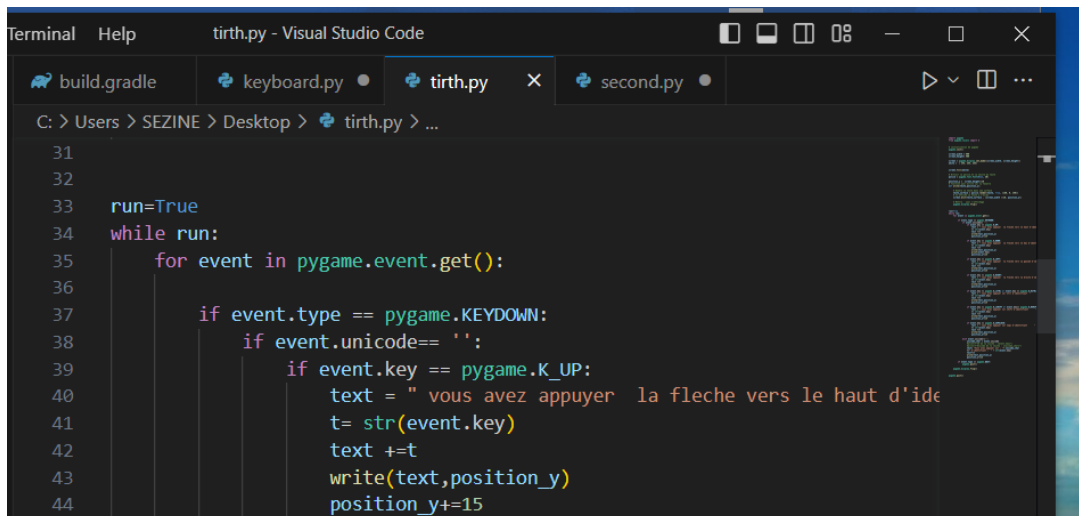
The image shows a screenshot of the Visual Studio Code editor interface. The title bar at the top reads "tirth.py - Visual Studio Code". Below the title bar, there are several tabs: "build.gradle", "keyboard.py", "tirth.py" (which is the active tab), and "second.py". The main editor area displays the following Python code:

```
C: > Users > SEZINE > Desktop > tirth.py > ...
1  import pygame
2  from pygame.locals import *
3
4
5  # initialisation de pygame
6  pygame.init()
7
8  screen_width = 600
9  screen_height= 500
10
11  screen = pygame.display.set_mode((screen_width, screen_height))
12  white = ( 255, 255, 255)
13
14
15  screen.fill(white)
16
```

On peut donc entrer dans le vif du sujet, la gestion des évènements clavier avec quelques etapes clés :

➤ **Capture des Événements Clavier :**

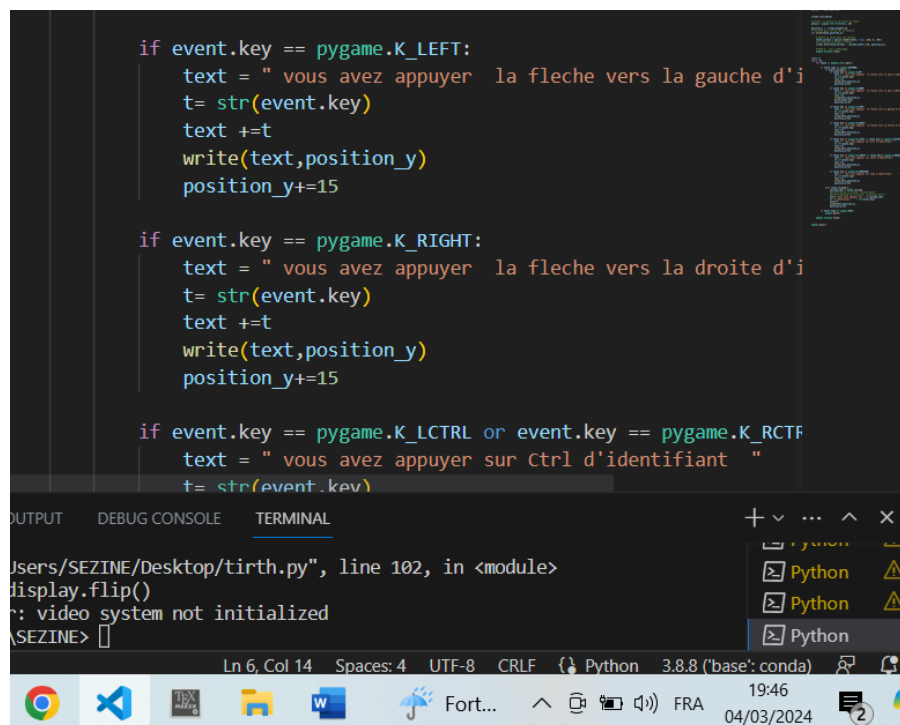
La boucle principale d'un jeu Pygame inclut généralement un bloc pour la gestion des événements. Pour capturer les événements clavier, on utilise la fonction `pygame.event.get ()` qui renvoie une liste d'événements survenus depuis la dernière fois que cette fonction a été appelée.



```
31
32
33 run=True
34 while run:
35     for event in pygame.event.get():
36
37         if event.type == pygame.KEYDOWN:
38             if event.unicode== '':
39                 if event.key == pygame.K_UP:
40                     text = " vous avez appuyer la fleche vers le haut d'id
41                     t= str(event.key)
42                     text +=t
43                     write(text,position_y)
44                     position_y+=15
```

➤ **Utilisation des Constantes Pygame pour les Touches :**

Pygame fournit un ensemble de constantes pour représenter les touches du clavier, telles que `pygame.K_LEFT`, `pygame.K_RIGHT`, `pygame.K_SPACE`, etc. Ces constantes facilitent la gestion des événements clavier sans avoir à mémoriser les codes de touches spécifiques.



```
if event.key == pygame.K_LEFT:
    text = " vous avez appuyer la fleche vers la gauche d'i
    t= str(event.key)
    text +=t
    write(text,position_y)
    position_y+=15

if event.key == pygame.K_RIGHT:
    text = " vous avez appuyer la fleche vers la droite d'i
    t= str(event.key)
    text +=t
    write(text,position_y)
    position_y+=15

if event.key == pygame.K_LCTRL or event.key == pygame.K_RCTF
    text = " vous avez appuyer sur Ctrl d'identifiant "
    t= str(event.kev)
```

Users/SEZINE/Desktop/tirth.py", line 102, in <module>
display.flip()
pygame.error: video system not initialized
SEZINE> █

Ln 6, Col 14 Spaces: 4 UTF-8 CRLF Python 3.8.8 ('base': conda)

- En Pygame, le terme "**Unicode**" se réfère généralement à la représentation des caractères textuels à l'aide du standard Unicode. Dans le contexte de Pygame, la gestion de l'Unicode est pertinente principalement lors de la saisie de texte via le clavier, car cela permet de traiter des caractères de

toutes les langues. La propriété clé en jeu est `event.unicode` qui est utilisée lors de la gestion des événements de type `KEYDOWN` dans Pygame. Lorsqu'une touche est enfoncée, l'événement `KEYDOWN` est déclenché, et `event.unicode` contient le caractère Unicode associé à la touche pressée.

Dans notre projet, nous avons subdiviser notre clavier en deux partis à savoir celle qui ont un **unicode différent de la chaîne de caractère vide** et celle qui ont un **unicode égale à la chaîne de caractère vide**. D'une part, comme de l'autre on récupère l'identifiant de la touches grâce à `event.key`

Voici une capture d'écran de notre travail sur la capture d'évènements clavier ainsi que leur identifiants grâce à pygame

