



Municipalidad de
Tres de Febrero



Programá
tu futuro



PYTHON INTERMEDIO

¡Les damos la bienvenida!





Municipalidad de
Tres de Febrero



Programá
tu futuro



PYGAMES

CLASE 7



¿Qué es PYGAME?

es una biblioteca de Python utilizada para el desarrollo de videojuegos en dos dimensiones. Proporciona una interfaz simple y fácil de usar para crear juegos 2D con gráficos, sonido, entrada de usuario y mucho más.



Características

- **Sencillo de aprender:** Pygame está diseñado para ser accesible incluso para programadores principiantes. Utiliza sintaxis de Python, lo que lo hace familiar para aquellos que ya conocen el lenguaje.
- **Versátil:** Pygame se puede utilizar para crear una amplia variedad de juegos 2D, desde simples juegos arcade hasta juegos de plataformas e incluso juegos de rol básicos.
- **Multiplataforma:** Los juegos creados con Pygame se pueden ejecutar en Windows, macOS y Linux sin modificaciones en el código.
- **Gratuito y de código abierto:** Pygame es una biblioteca gratuita y de código abierto, lo que significa que puedes utilizarla y modificarla libremente para tus proyectos.

Que podemos hacer ?

- Crear sprites (imágenes animadas) y fondos para tu juego.
- Reproducir música y efectos de sonido.
- Detectar la entrada del usuario a través del teclado, mouse o joystick.
- Gestionar colisiones entre objetos en el juego.
- Implementar la lógica del juego, como reglas de puntuación y ganar/perder.



En resumen

Pygame es una herramienta poderosa para crear juegos 2D divertidos y atractivos. Si estás interesado en el desarrollo de videojuegos, Pygame es un excelente punto de partida.

Documentación: <https://www.pygame.org/docs/>

CONECTA 4 - PARTE 1



Programá
tu futuro



Municipalidad de
Tres de Febrero

Empecemos...

Vamos a crear un entorno virtual con el nombre de **Conecta4**

```
py -m venv Conecta4
```

Ahora vamos a encender el entorno

```
source Conecta4/Scripts/activate
```

Una vez activado instalamos los paquetes

```
pip install numpy pygame
```


Definamos lo importante

Vamos a crear nuestro archivo .py y vamos a importar los paquetes que usaremos

```
import numpy as np  
import pygame  
import sys
```

Ahora guardaremos
en **constantes** el
valor de las filas y
columnas

```
FILA = 6  
COLUMNA = 7
```

También podemos guardar los colores
que usaremos

```
AZUL = (0,0,255)  
NEGRO = (0,0,0)
```

Creaendo Tablero

Vamos a definir una funcion para crear nuestro tablero

```
def crear_tablero():  
    board = np.zeros((FILAS,COLUMNAS))  
    return board
```

El metodo **zeros** de numpy, nos permite crear un array bi dimensional pasandole una tupla para definir el alto y ancho

Creaendo Tablero

Vamos a definir una funcion para dibujar nuestro tablero

```
def dibujar_tablero(board):  
    for c in range(COLUMNA):  
        for f in range(FILA):  
            pygame.draw.rect(pantalla, AZUL, (c*TAMANIOCAJA, f*TAMANIOCAJA+TAMANIOCAJA,  
TAMANIOCAJA, TAMANIOCAJA))  
            pygame.draw.circle(pantalla, NEGRO, (int(c*TAMANIOCAJA+TAMANIOCAJA/2),  
int(f*TAMANIOCAJA+TAMANIOCAJA+TAMANIOCAJA/2)), RADIO)
```

Como ven necesitamos definir nuevos datos, vamos a eso

Creaendo Tablero

Vamos a definir los datos que nos estan faltando

```
TAMANIOCAJA = 100  
ancho = COLUMNA * TAMANIOCAJA  
alto = (FILA+1) * TAMANIOCAJA  
  
tamanio = (ancho, alto)  
RADIO = int(TAMANIOCAJA/2 - 5)
```

Ahora vamos a definir la pantalla

Creaendo Tablero

Vamos a definir los datos que nos estan faltando

```
pantalla = pygame.display.set_mode(tamano)
```

bien, ahora **antes de nuestra variable pantalla**, vamos a definir la variable tablero y una variable game:_over (ya veremos para que)

```
tablero = crear_tablero()  
game_over = False
```


Creaendo Tablero

Ahora **entre la definician de la variable tablero y la creacion de la variable pantalla** vamos a indicarle a pygame que inicie con el metodo init

```
pygame.init()
```

Bien ahora, luego de defiinr la **variable pantalla**, vamos a ejecutar la funcion para dibujar el tablero

```
dibujar_tablero(tablero)
```

Creaendo Tablero

De seguro lo intentaron ejecutar y parpadeo , bien definamos el bucle principiapl

```
while not game_over:
    #con esto evitamos que la pantalla se
    nos cierre
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            sys.exit()
```

Este bucle va ultimo de todo en nuestro codigo actual ... pero por que queda en negro ?

Creando Tablero

Como ultimo paso **luego de ejecutar la funcion dibujar_tablero(tablero)**
vamos a updatear nuestra pantalla

```
pygame.display.update()
```

Con esto vamos a refrescar la pantalla para ir viendo los cambios

¡MUCHAS GRACIAS!



Programá
tu futuro



Municipalidad de
Tres de Febrero