



Municipalidad de  
Tres de Febrero



Programá  
tu futuro



# PYTHON INTERMEDIO

¡Les damos la bienvenida!





Municipalidad de  
Tres de Febrero



Programá  
tu futuro



# PYGAMES

CLASE 8



# CONECTA 4 - PARTE 2



Programá  
tu futuro



Municipalidad de  
Tres de Febrero



## Continuemos..

Agreguemos a la funcion de dibujar el tablero , el codigo para dibujar las fichas

```
for c in range(COLUMNA):
    for f in range(FILA):
        if board[f][c] == 1:
            pygame.draw.circle(pantalla, AMARILLO,
                                (int(c*TAMANIOCAJA+TAMANIOCAJA/2),
                                 (alto+TAMANIOCAJA)-
                                  int(f*TAMANIOCAJA+TAMANIOCAJA/2)), RADIO)
        elif board[f][c] == 2:
            pygame.draw.circle(pantalla, ROJO, (int(c*TAMANIOCAJA+TAMANIOCAJA/2),
                                (alto+TAMANIOCAJA)-int(f*TAMANIOCAJA+TAMANIOCAJA/2)), RADIO)

pygame.display.update()
```

Como aca le agregamos el update, el que teniamos fuera lo podemos comentar

# Funcion para ganar

Vamos a agregar una funcion que nos sirva para corroborar si se ha ganado

```
def es_ganador(board, piece):
    #revisando las posiciones horizontales
    for c in range(COLUMNA-3):
        for r in range(FILA):
            if board[r][c] == piece and board[r][c+1] == piece and board[r][c+2] == piece and board[r][c+3] == piece:
                return True

    # verificando las posiciones verticales
    for c in range(COLUMNA):
        for r in range(FILA-3):
            if board[r][c] == piece and board[r+1][c] == piece and board[r+2][c] == piece and board[r+3][c] == piece:
                return True

    # verificando diagonales positivas
    for c in range(COLUMNA-3):
        for r in range(FILA-3):
            if board[r][c] == piece and board[r+1][c+1] == piece and board[r+2][c+2] == piece and board[r+3][c+3] == piece:
                return True

    # verificando diagonales negativas
    for c in range(COLUMNA-3):
        for r in range(3, FILA):
            if board[r][c] == piece and board[r-1][c+1] == piece and board[r-2][c+2] == piece and board[r-3][c+3] == piece:
                return True
```

## El turno de ...

Agregaremos una variable para definir a los usuarios

```
tablero = crear_tablero()  
game_over = False  
Turno = 0
```

Con tinuemos haciendo una modificacion al bucle principal

# Habeus fichas

En nuestro bucle principal, vamos a agregar este fragmento para dibujar y poder movernos con las fichas

```
if event.type == pygame.MOUSEMOTION:
    pygame.draw.rect(pantalla, NEGRO, (0,0, ancho,TAMANIOCAJA))
    posX = event.pos[0]
    if turno == 0:
        pygame.draw.circle(pantalla, AMARILLO, (posx, int(TAMANIOCAJA/2)), RADIO)
    else:
        pygame.draw.circle(pantalla, ROJO, (posx, int(TAMANIOCAJA/2)), RADIO)
pygame.display.update()
```

Ejecutemos, aver como vamos

## Verificando si esta vacio

Vamos a crear una funcion para verificar que la posicion este vacia

```
def lugar_valida(board, col):  
    return board[FILA-1][col] == 0
```

Ahora una para la siguiente fila

```
def obtener_siguiente_fila_disponible(board, col):  
    for r in range(FILA):  
        if board[r][col] == 0:  
            return r
```



## Dejala ser...

Ahora por ultimo creamos una funcion para que suelte la ficha

```
def soltar_pieza(board, row, col, piece):  
    board[row][col] = piece
```

Bien.. ahora que ya tenemos esto.. antes de avanzar, creamos uan funcion para actualizar el tablero que cremaos con numpy

```
def mostrar_tablero(board):  
    print(np.flipud(board))
```

# Volviendo al Bucle principal

Bien, vamos a solicitarle al jugador uno que mueva

```
if event.type == pygame.MOUSEBUTTONDOWN:
    pygame.draw.rect(pantalla, NEGRO, (0,0, ancho, TAMANIOCAJA))

    if turno == 0:
        posx = event.pos[0]
        col = int(math.floor(posx/TAMANIOCAJA))

        if lugar_valida(tablero, col):
            row = obtener_siguiente_fila_disponible(tablero, col)
            soltar_pieza(tablero, row, col, 1)

        if es_ganador(tablero, 1):
            label = MY_FUENTE.render("Jugador 1 Gana!!!", 1, VERDE)
            pantalla.blit(label, (40,10))
            game_over = True
```

## Agregando los detalles nuevos

Lo primero será importar **math**

```
import math
```

Seguido agregamos el color verde

```
VERDE = (0, 255, 0)
```

Luego seteamos la fuente

```
pygame.font.init()  
MY_FUENTE = pygame.font.SysFont("monospace", 75)
```

# A pasar el turno

Vamos a pedirle al jugador 2 que juegue

```
else:
```

```
    posx = event.pos[0]
```

```
    col = int(math.floor(posx/TAMANIOCAJA))
```

```
    if lugar_valida(tablero, col):
```

```
        row = obtener_siguiente_fila_disponible(tablero, col)
```

```
        soltar_pieza(tablero, row, col, 2)
```

```
    if es_ganador(tablero, 2):
```

```
        label = MY_FUENTE.render("Jugador 2 Gana!!!", 1, VERDE)
```

```
        pantalla.blit(label, (40,10))
```

```
        game_over = True
```

# Terminando...

Por ultimo Vamos a llamar a la funcion para dibujar el tablero, y asi refresque los movimientos que hagamos,y cambiemos el turno

**dibujar\_tablero(tablero)**

```
turn += 1
```

```
turn = turn % 2
```

```
if game_over:
```

```
pygame.time.wait(3000)
```





# ¡MUCHAS GRACIAS!



Programá  
tu futuro



Municipalidad de  
Tres de Febrero