In [1]:
```
pip install pygame
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting pygame
  Downloading pygame-2.5.2-cp39-cp39-win_amd64.whl (10.8 MB)
     ---------------------------------------- 10.8/10.8 MB 3.4 MB/s eta 0:00:
00
Installing collected packages: pygame
Successfully installed pygame-2.5.2
Note: you may need to restart the kernel to use updated packages.
```

In [101]:
```
# odigo prueba
import pygame
import sys
import random

# Constantes
ANCHO = 800
ALTO = 600
COLOR_NEGRO = (0, 0, 0)
COLOR_BLANCO = (255, 255, 255)
COLOR_ROJO = (255, 0, 0)
COLOR_AZUL = (0, 0, 255)

# Inicializar Pygame
pygame.init()

# Crear ventana
ventana = pygame.display.set_mode((ANCHO, ALTO))
pygame.display.set_caption("Menú")

# Fuente para el texto del menú
fuente = pygame.font.Font(None, 36)

def mostrar_menu():
    ventana.fill(COLOR_NEGRO)

    texto_titulo = fuente.render("Juego", True, COLOR_BLANCO)
    ventana.blit(texto_titulo, (ANCHO // 2 - texto_titulo.get_width() // 2, 100

    texto_circulo = fuente.render("1. Círculo", True, COLOR_BLANCO)
    ventana.blit(texto_circulo, (ANCHO // 2 - texto_circulo.get_width() // 2, 1

    texto_rectangulo = fuente.render("2. Rectángulo", True, COLOR_BLANCO)
    ventana.blit(texto_rectangulo, (ANCHO // 2 - texto_rectangulo.get_width() /

    texto_cuadrado = fuente.render("3. Cuadrado", True, COLOR_BLANCO)
    ventana.blit(texto_cuadrado, (ANCHO // 2 - texto_cuadrado.get_width() // 2

    texto_salir = fuente.render("4. Salir", True, COLOR_BLANCO)
    ventana.blit(texto_salir, (ANCHO // 2 - texto_salir.get_width() // 2, 350))

    pygame.display.flip()

def juego_principal(forma):
    # Código principal del juego
    game_over = False
    clock = pygame.time.Clock()

    jugador_ancho = 50
    jugador_size = 50
    jugador_posicion = [ANCHO / 2, ALTO - jugador_ancho * 2]  # Posición inici
    jugador_velocidad_y = 0
    gravedad = 0.5
    salto = -15  # Salto más rápido
    salto_disponible = True  # Indica si el jugador puede saltar en este momen

    # Lista de rectángulos que caen del cielo
```

```
rectangulos_cielo = []

# Lista de rectángulos en la parte inferior de la pantalla
rectangulos_inferior = []
rectangulo_velocidad = 5

def generar_rectangulo_cielo():
    # Generar un nuevo rectángulo en la parte superior de la pantalla
    rectangulo_largo = random.randint(50, 150)
    rectangulo_posicion = [ANCHO, random.randint(0, ALTO - 20)]  # Parte su
    rectangulos_cielo.append([rectangulo_posicion, rectangulo_largo])

def generar_rectangulo_inferior():
    # Generar un nuevo rectángulo en la parte inferior de la pantalla
    rectangulo_alto = random.randint(20, 100)
    rectangulo_posicion = [ANCHO, ALTO - 20 - rectangulo_alto]  # Parte inf
    rectangulos_inferior.append([rectangulo_posicion, rectangulo_alto])

def mover_rectangulos_inferior():
    # Mover todos los rectángulos de derecha a izquierda
    for rectangulo in rectangulos_inferior:
        rectangulo[0][0] -= rectangulo_velocidad
        if rectangulo[0][0] <= 0:
            rectangulos_inferior.remove(rectangulo)  # Eliminar si sale de

def mover_rectangulos_cielo():
    # Mover todos los rectángulos de derecha a izquierda
    for rectangulo in rectangulos_cielo:
        rectangulo[0][0] -= 5  # Mover hacia la izquierda
        if rectangulo[0][0] <= 0:
            rectangulos_cielo.remove(rectangulo)  # Eliminar si sale de la

def detectar_colision(jugador_posicion, rectangulos):
    jx = jugador_posicion[0]
    jy = jugador_posicion[1]
    for rectangulo in rectangulos:
        rx = rectangulo[0][0]
        ry = rectangulo[0][1]
        ancho = rectangulo[1]
        if (rx >= jx and rx < (jx + jugador_size)) or (jx >= rx and jx < (r
            if (ry >= jy and ry < (jy + jugador_size)) or (jy >= ry and jy
                return True
    return False

while not game_over:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            game_over = True
        if event.type == pygame.KEYDOWN:  # el evento en que se presione un
            if event.key == pygame.K_LEFT:
                jugador_posicion[0] -= jugador_ancho
            elif event.key == pygame.K_RIGHT:
                jugador_posicion[0] += jugador_ancho
            elif event.key == pygame.K_UP:  # Permitir salto en cualquier m
                jugador_velocidad_y = salto

    ventana.fill(COLOR_NEGRO)
```

```
        # Generar rectángulos aleatorios en el cielo
        if random.randint(0, 100) < 2:  # Probabilidad de generar un nuevo rect
            generar_rectangulo_cielo()

        # Generar rectángulos aleatorios en la parte inferior de la pantalla
        if random.randint(0, 100) < 2:  # Probabilidad de generar un nuevo rect
            generar_rectangulo_inferior()

        # Mover y dibujar rectángulos del cielo
        for rectangulo in rectangulos_cielo:
            pygame.draw.rect(ventana, COLOR_BLANCO, (rectangulo[0][0], rectangu

        if detectar_colision(jugador_posicion, rectangulos_cielo):
            game_over = True

        mover_rectangulos_cielo()

        # Mover y dibujar rectángulos inferiores
        for rectangulo in rectangulos_inferior:
            pygame.draw.rect(ventana, COLOR_AZUL, (rectangulo[0][0], rectangulo

        if detectar_colision(jugador_posicion, rectangulos_inferior):
            game_over = True

        mover_rectangulos_inferior()

        # Aplicar gravedad al jugador
        jugador_velocidad_y += gravedad
        jugador_posicion[1] += jugador_velocidad_y

        # Limitar la posición y velocidad del jugador para que no se salga de l
        if jugador_posicion[1] > ALTO - jugador_ancho * 2:
            jugador_posicion[1] = ALTO - jugador_ancho * 2
            jugador_velocidad_y = 0

        # Dibujar jugador según la forma seleccionada
        if forma == "circulo":
            pygame.draw.circle(ventana, COLOR_ROJO, (int(jugador_posicion[0] +
        elif forma == "rectangulo":
            pygame.draw.rect(ventana, COLOR_ROJO, (jugador_posicion[0], jugador
        elif forma == "cuadrado":
            pygame.draw.rect(ventana, COLOR_ROJO, (jugador_posicion[0], jugador

        # Actualizar la ventana
        clock.tick(30)
        pygame.display.update()

    # Salir del juego
    pygame.quit()
    sys.exit()

def main():
    while True:
        mostrar_menu()

        for event in pygame.event.get():
```

```
            if event.type == pygame.QUIT:
                pygame.quit()
                sys.exit()
            elif event.type == pygame.KEYDOWN:
                if event.key == pygame.K_1:
                    juego_principal("circulo")
                elif event.key == pygame.K_2:
                    juego_principal("rectangulo")
                elif event.key == pygame.K_3:
                    juego_principal("cuadrado")
                elif event.key == pygame.K_4:
                    pygame.quit()
                    sys.exit()

if __name__ == "__main__":
    main()
```

```
An exception has occurred, use %tb to see the full traceback.

SystemExit
```

In [ ]:

In [15]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: