



Práctica 1: Crear una ventana principal de un juego.	Práctica 2: COLORES RGB
<pre> 1 import pygame 2 3 # Iniciar Pygame 4 pygame.init() 5 6 # Crear ventana 7 pantalla = pygame.display.set_mode((800, 600)) 8 9 # Título de la ventana 10 pygame.display.set_caption("Mi primer juego") 11 12 # Variable para mantener el juego abierto 13 corriendo = True 14 15 # Loop principal del juego 16 while corriendo: 17 18 # Revisar eventos 19 for evento in pygame.event.get(): 20 21 # Si el usuario cierra la ventana 22 if evento.type == pygame.QUIT: 23 corriendo = False 24 25 # Cerrar Pygame 26 pygame.quit() </pre>	<pre> import pygame pygame.init() pantalla = pygame.display.set_mode((800,600)) pygame.display.set_caption("Colores RGB") AZUL = (0, 0, 255) corriendo = True while corriendo: # Pintar fondo pantalla.fill(AZUL) for evento in pygame.event.get(): if evento.type == pygame.QUIT: corriendo = False pygame.display.update() pygame.quit() </pre>
<p>Cambiar tamaño de ventana: 1000 x 700 o por 1200 x 800</p> <ul style="list-style-type: none"> ✓ pygame.init(): ✓ set_mode(): ✓ set_caption(): ✓ loop principal : ✓ evento QUIT: 	<p>Crear: un fondo gamer o un fondo Minecraft y un color favorito</p> <ul style="list-style-type: none"> ✓ fill() ✓ RGB: ✓ pygame.display.update() <p>ROJO = (255,0,0) VERDE = (0,255,0) NEGRO = (0,0,0) BLANCO = (255,255,255)</p>
<p style="text-align: center;">Sello de realizado en clase</p>	<p style="text-align: center;">Sello de culminado en clase</p>
Empty space for stamp	Empty space for stamp



Práctica 3: DIBUJAR FIGURAS	Práctica 4. DETECTAR TECLADO: leer teclas presionadas
<pre data-bbox="79 243 724 1128">import pygame pygame.init() pantalla = pygame.display.set_mode((800,600)) BLANCO = (255,255,255) ROJO = (255,0,0) AZUL = (0,0,255) VERDE = (0,255,0) corriendo = True while corriendo: pantalla.fill(BLANCO) # Rectángulo pygame.draw.rect(pantalla, ROJO, (100,100,200,100)) # Círculo pygame.draw.circle(pantalla, AZUL, (500,200), 50) # Línea pygame.draw.line(pantalla, VERDE, (0,0), (800,600), 5) for evento in pygame.event.get(): if evento.type == pygame.QUIT: corriendo = False pygame.display.update() pygame.quit()</pre> <ul style="list-style-type: none"> <input checked="" type="checkbox"/> rectángulo: <code>pygame.draw.rect(pantalla, color, (x,y,ancho,alto))</code> <input checked="" type="checkbox"/> círculos: <input checked="" type="checkbox"/> líneas: <input checked="" type="checkbox"/> coordenadas x,y: <p>Construir con figuras: una casa, semáforo, o robot o personaje pixelado</p>	<pre data-bbox="1096 243 1711 1128">import pygame pygame.init() pantalla = pygame.display.set_mode((800,600)) corriendo = True while corriendo: for evento in pygame.event.get(): if evento.type == pygame.QUIT: corriendo = False if evento.type == pygame.KEYDOWN: if evento.key == pygame.K_RIGHT: print("Derecha") if evento.key == pygame.K_LEFT: print("Izquierda") if evento.key == pygame.K_SPACE: print("Salto") pygame.quit()</pre> <ul style="list-style-type: none"> <input checked="" type="checkbox"/> eventos <input checked="" type="checkbox"/> teclado <input checked="" type="checkbox"/> <code>pygame.KEYDOWN</code> <p>Configurar:</p> <ul style="list-style-type: none"> • A = atacar, W = saltar, ESC = salir
Sello de realizado en clase	Sello de culminado en clase



Práctica 5:	Mover un personaje	Práctica 6.	Imágenes: Usar sprites reales.
<pre> 1 import pygame 2 3 pygame.init() 4 5 pantalla = pygame.display.set_mode((800,600)) 6 x = 100 7 y = 100 8 velocidad = 5 9 10 corriendo = True 11 12 while corriendo: 13 14 pantalla.fill((255,255,255)) 15 teclas = pygame.key.get_pressed() 16 if teclas[pygame.K_RIGHT]: 17 x += velocidad 18 if teclas[pygame.K_LEFT]: 19 x -= velocidad 20 if teclas[pygame.K_UP]: 21 y -= velocidad 22 if teclas[pygame.K_DOWN]: 23 y += velocidad 24 pygame.draw.rect(pantalla,(0,0,255),(x,y,50,50)) 25 26 for evento in pygame.event.get(): 27 if evento.type == pygame.QUIT: 28 corriendo = False 29 pygame.display.update() 30 31 pygame.quit()</pre>	<p>coordenadas</p> <p>velocidad</p> <p>key.get_pressed()</p> <p>Crear:</p> <ul style="list-style-type: none"> • mini laberinto • mover personaje por pantalla 	<pre> import pygame pygame.init() pantalla = pygame.display.set_mode((800,600)) personaje = pygame.image.load("personaje.png") x = 100 y = 100 corriendo = True while corriendo: pantalla.fill((255,255,255)) pantalla.blit(personaje,(x,y)) for evento in pygame.event.get(): if evento.type == pygame.QUIT: corriendo = False pygame.display.update() pygame.quit()</pre>	<p>✓ image.load()</p> <p>✓ blit()</p> <p>Actividad Usar: personaje de proyecto</p>
<p>Sello de realizado en clase</p>		<p>Sello de culminado en clase</p>	
Empty space for stamp		Empty space for stamp	



Práctica 7: Colisiones: Detectar choques.

```
1 import pygame
2
3 pygame.init()
4
5 pantalla = pygame.display.set_mode((800,600))
6
7 jugador = pygame.Rect(100,100,50,50)
8 moneda = pygame.Rect(500,300,40,40)
9
10 corriendo = True
11
12 while corriendo:
13
14     pantalla.fill((255,255,255))
15
16     teclas = pygame.key.get_pressed()
17
18     if teclas[pygame.K_RIGHT]:
19         jugador.x += 5
20
21     if teclas[pygame.K_LEFT]:
22         jugador.x -= 5
23
24     if teclas[pygame.K_UP]:
25         jugador.y -= 5
26
27     if teclas[pygame.K_DOWN]:
28         jugador.y += 5
29
30     if jugador.colliderect(moneda):
31         print("Moneda recolectada")
32
33     pygame.draw.rect(pantalla,(0,0,255),jugador)
34     pygame.draw.rect(pantalla,(255,215,0),moneda)
35
36     for evento in pygame.event.get():
37         if evento.type == pygame.QUIT:
38             corriendo = False
39
40     pygame.display.update()
41
42 pygame.quit()
```

✓ Rect ✓ colliderect()

Recolectar:

- monedas
- comida
- vidas

Sello de realizado en clase



Práctica 8: Sonidos: Agregar efectos al videojuego.

```
1 import pygame
2
3 # Iniciar Pygame
4 pygame.init()
5
6 # Crear ventana
7 pantalla = pygame.display.set_mode((800,600))
8
9 # Título de la ventana
10 pygame.display.set_caption("Sonido de salto")
11
12 # Cargar sonido
13 sonido = pygame.mixer.Sound("salto.wav")
14
15 # Variable del juego
16 corriendo = True
17
18 # Loop principal
19 while corriendo:
20
21     # Fondo blanco
22     pantalla.fill((255,255,255))
23
24     # Revisar eventos
25     for evento in pygame.event.get():
26
27         # Cerrar ventana
28         if evento.type == pygame.QUIT:
29             corriendo = False
30
31         # Detectar teclas
32         if evento.type == pygame.KEYDOWN:
33
34             # Barra espaciadora
35             if evento.key == pygame.K_SPACE:
36                 sonido.play()
37                 print("¡Salto!")
38
39     pygame.display.update()
40
41 # Salir de pygame
42 pygame.quit()
```

mixer.Sound()

Agregar sonidos:

- disparo
- salto
- moneda ganar

Sello de realizado en clase

Sello de culminado en clase



Práctica 9:

MINI JUEGO FINAL: Recolector de monedas.

Debe incluir

- movimiento
- teclado
- personaje
- moneda
- colisión
- sonido
- puntaje

Evidencia

Captura + código comentado + mejoras personales.

Sello de realizado en clase

Sello de culminado en clase