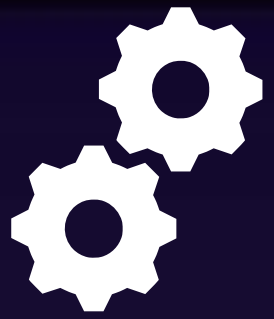

PyGame

PyGame

Características

- Biblioteca para el desarrollo de videojuegos 2d
- Facilita funcionalidades gráficas
- Maneja formas geométricas
- Contiene un sistema de colisiones
- Manejo de sonidos



Ventana

- Área grafica en la que se muestra el contenido del juego
- Es una superficie
- Puede contener imagenes, sprites y texto
- El usuario puede interactuar con la ventana y con su contenido

Creación de Ventana

```python
#Para inicializar pygame
pygame.init()

# Creamos la ventana
PANTALLA = pygame.display.set_mode((500,400)) # en pixeles

# Establecemos el titulo de la ventana
pygame.display.set_caption("Mi primer juego.")
```

- Creamos la ventana principal
- Creamos un bucle para mantener la ventana abierta
- Leemos las acciones del usuario y las manejamos

```python
while True:
    for event in pygame.event.get():
        if event.type == QUIT:
            pygame.quit()
            sys.exit()
    pygame.display.update()
```

Eventos

- Acciones o sucesos que ocurren durante la ejecución
- Representan interacciones del usuario
- Deben ser capturados y manejados

Eventos

- Fundamentales para interactuar con el jugador
- Detectamos las acciones efectuadas
- Las manejamos con el módulo de pygame

Eventos

```python
while True:
    #Obtengo la lista de eventos que estan ocurriendo y la recorro
    for event in pygame.event.get():
        #Verifico el typo del evento para saber si coincide con lo que busco
        if event.type == QUIT:
            #Ejecuto las acciones correspondientes con el evento
            pygame.quit()
            sys.exit()
```

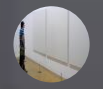
```python
# Obtengo la lista de teclas presionadas
teclas_presionadas = pygame.key.get_pressed()
# Verifico si es la tecla que necesito
if teclas_presionadas[pygame.K_RIGHT]:
    # Ejecuto la accion correspondiente
    mover_derecha()
```

```python
for event in pygame.event.get():
    # Escucho y verifico si el evento es que una tecla fue presionada
    # Una vez y no mantenida
    if event.type == pygame.KEYDOWN:
        # Ejecuto la accion correspondiente
        if event.key == pygame.K_TAB:
            funcion_compleja()
```

Con la método pygame.event.get() obtenemos la lista de eventos ocurriendo y podemos verificar las acciones que realizo el usuario con el tipo (.type) de evento,

Superficies

- La primer superficie es la ventana



- Se pueden dibujar elementos gráficos

- Lienzo en el que se van a imprimir distintos elementos

- Tamaño definido en pixeles

- Cada pixel representa un color
- Puede representar una imagen

Superficies

La Ventana es la primer superficie creada

```
# Definimos el tamaño de nuestra pantalla en pixeles
tamaño_en_pixeles = (1000,500)
# Creamos una ventana del tamaño dado
PANTALLA = pygame.display.set_mode(tamaño_en_pixeles)
# Nos devuelve la superficie que se va a imprimir en la ventana
```

Podemos crear superficies vacías

```
# Definimos un tamaño en pixeles
tamaño_superficie = (200, 50)
# Creamos una superficie vacia de ese tamaño
superficie = pygame.Surface((200,50))
```

Podemos crear una superficie a partir de una imagen

```
fondo = pygame.image.load("fondo.jpg")
```

Formatos soportados: bmp | png | jpg| gif.

Superficies

Métodos útiles

```
superficie.blit(otra_superficie, posicion)
```

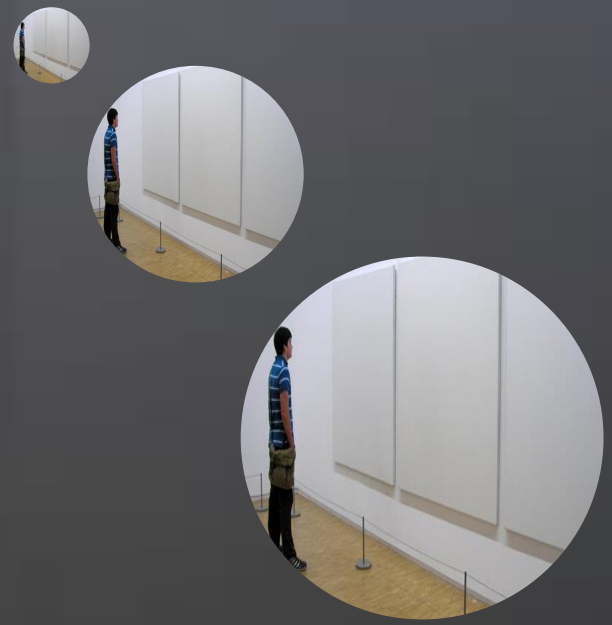
Imprimimos una superficie sobre otra superficie en la posición dada

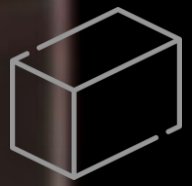
```
superficie.fill("Red")
```

Pintamos todos los pixeles del color ingresado

```
superficie.get_rect()
```

Obtenemos un rectángulo del tamaño de la superficie

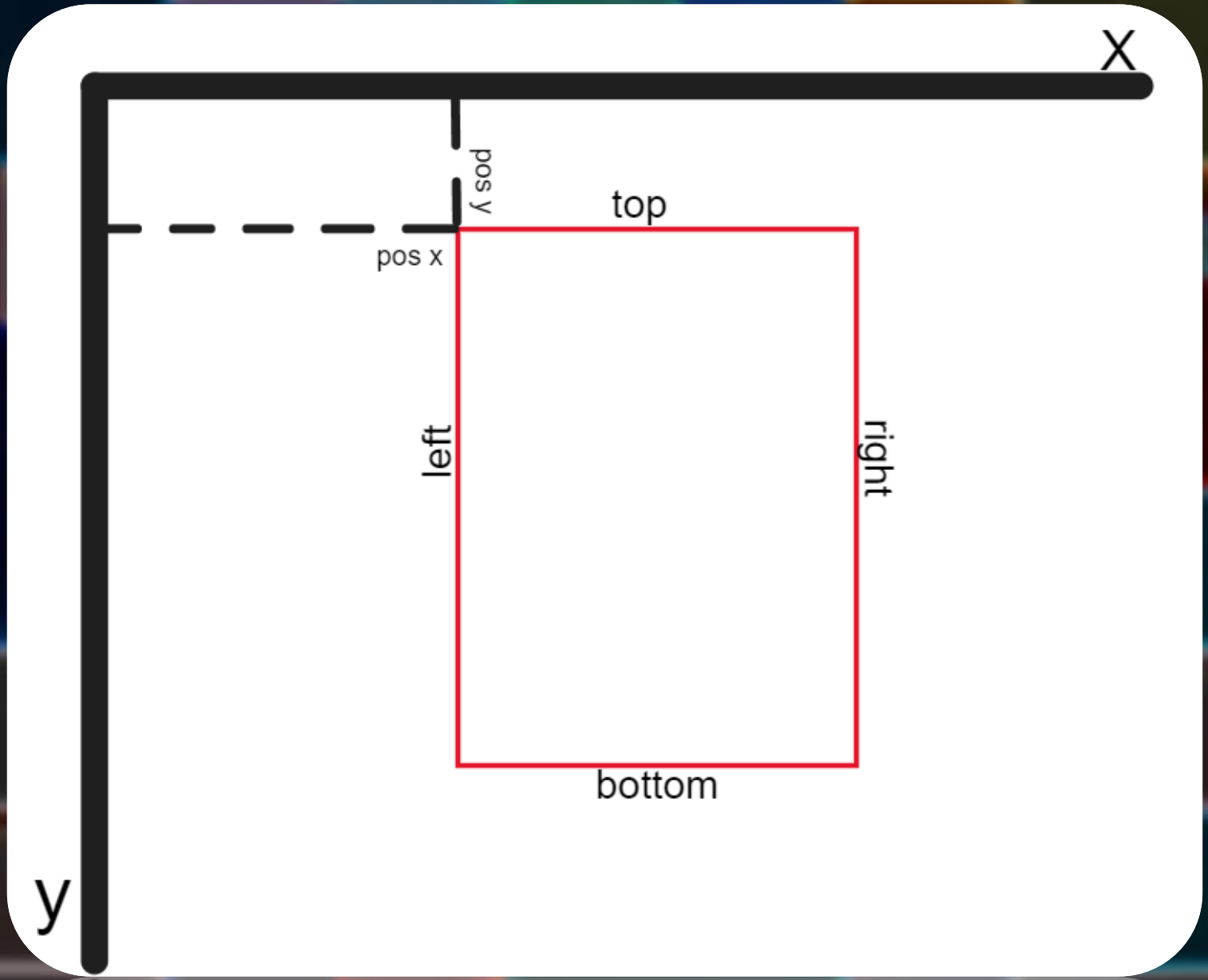


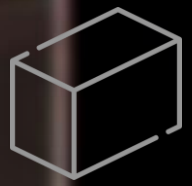


Rectángulos

- Es uno de los objetos principales en un juego.
- Representa abstractamente un área rectangular
- Sirve para definir posición (x e y) y tamaño (width y height)
- Podemos obtener las posiciones sus lados con propiedades
- Podemos Verificar colisión con otros rectángulos

Rectángulos





Rectángulos: Creación

A partir de parámetros

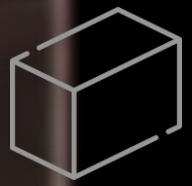
```
# Definimos (x e y) en el que se va a encontrar el rectangulo
x = 40
y = 30

# Definimos su tamaño (ancho y alto)
width = 400
height = 300

# Creamos el rectangulo con proporciones correspondientes
rectangulo = pygame.Rect(x, y, width, height)
```

A partir de una superficie

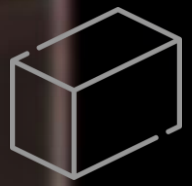
```
superficie.get_rect()
```



Rectángulos

Propiedades

```
# Son de lectura y escritura
# Retorna la posicion en x del rectangulo
rectangulo.x
# Retorna la posicion en y del rectangulo
rectangulo.y
# Retorna la posicion en y del lado superior del rectangulo
rectangulo.top
# Retorna la posicion en y del lado inferior del rectangulo
rectangulo.bottom
# Retorna la posicion en x del lado derecho del rectangulo
rectangulo.right
# Retorna la posicion en y del lado izquierdo del rectangulo
rectangulo.left
# Retorna el punto en (x, y) del centro del rectangulo
rectangulo.center
```



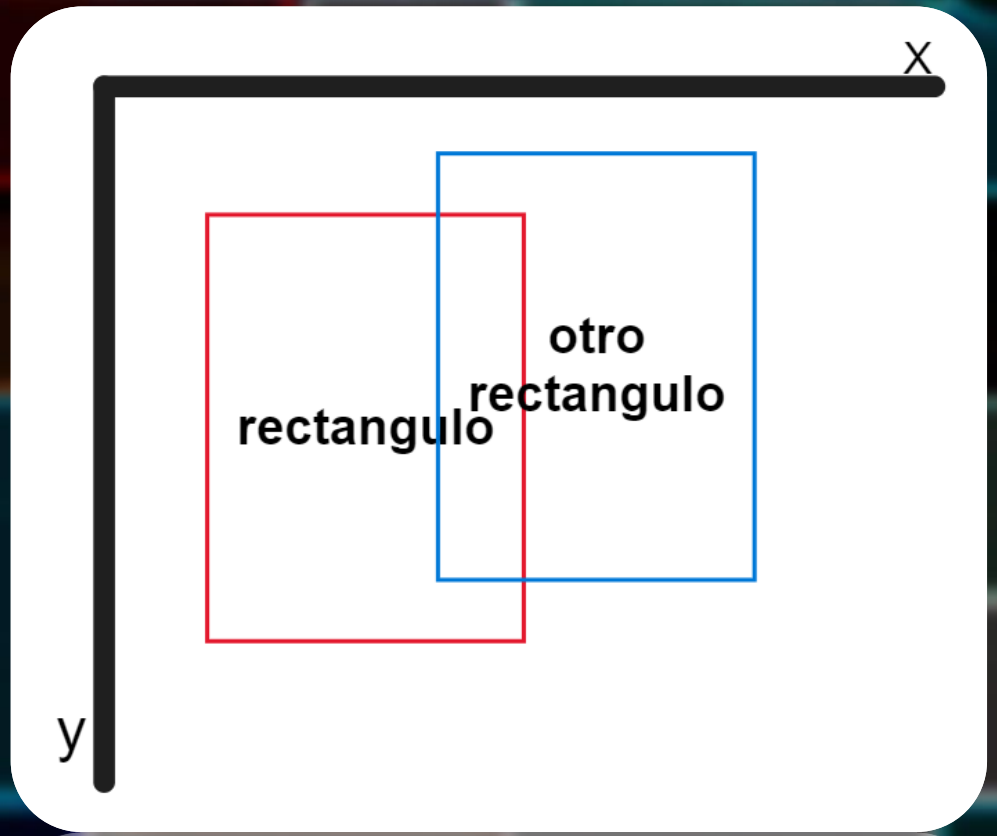
Rectángulos

Métodos

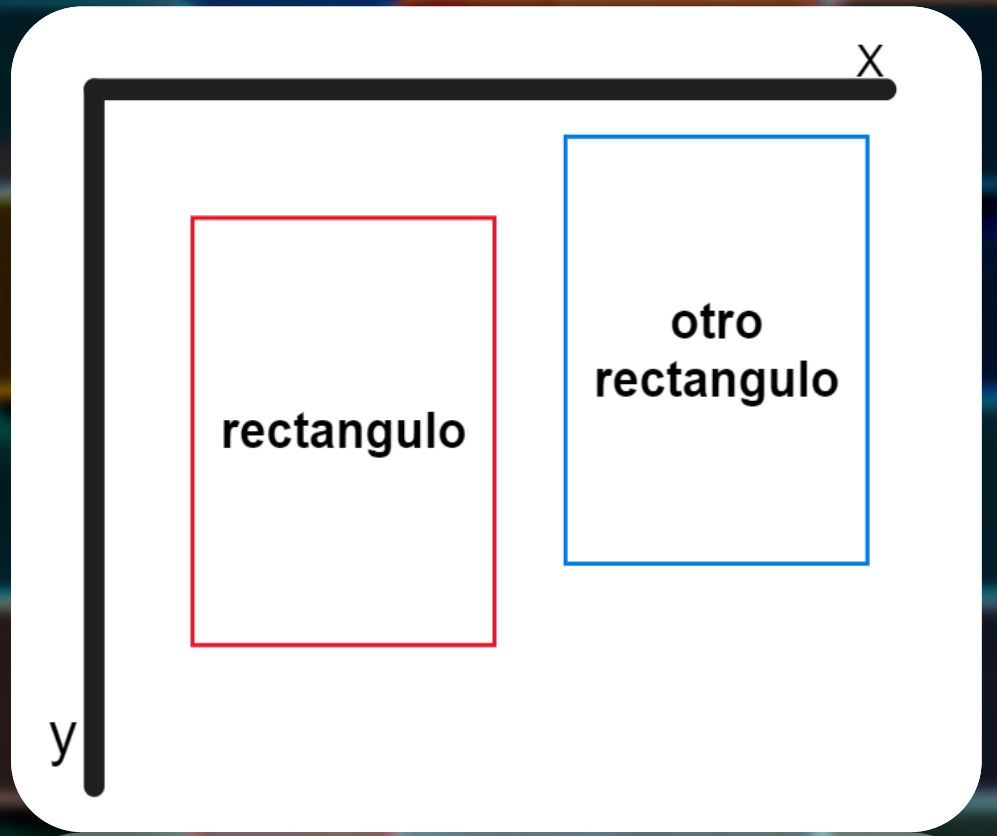
```
# Retorna true si los rectangulos estan colisionando
rectangulo.colliderect(otro_rectangulo)
# Retorna true si el punto en (x, y) esta adentro del rectangulo
rectangulo.collidepoint(x, y)
```

Rectángulos: Colisiones

```
# Retorna True
rectangulo.colliderect(otro_rectangulo)
```




```
# Retorna False
rectangulo.colliderect(otro_rectangulo)
```



Tiempo

- La mayor parte del ciclo de vida del programa se encuentra en el bucle principal del juego
- Es posible manipular la cantidad de iteraciones por segundo del bucle principal del juego

Tiempo: FPS

```python
FPS = 60
# Creamos un reloj
RELOJ = pygame.time.Clock()
while True:
    for event in pygame.event.get():
        if event.type == QUIT:
            pygame.quit()
            sys.exit()
    # Limitamos los FPS
    RELOJ.tick(FPS)
```

Es fundamental controlar los cuadros por segundo para mejorar la jugabilidad

Tiempo: FPS

FPS no limitados

FPS limitados

Tiempo: FPS

```
while True:
    for event in pygame.event.get():
        if event.type == QUIT:
            pygame.quit()
            sys.exit()
    # Limitamos los FPS
    personaje.mover_derecha()
    RELOJ.tick(FPS)
    pygame.display.flip()
```


```
while True:
    for event in pygame.event.get():
        if event.type == QUIT:
            pygame.quit()
            sys.exit()
    # Limitamos los FPS
    personaje.mover_derecha()
    RELOJ.tick(FPS)
    pygame.display.flip()
```



```
while True:
    for event in pygame.event.get():
        if event.type == QUIT:
            pygame.quit()
            sys.exit()
    # Limitamos los FPS
    personaje.mover_derecha()
    RELOJ.tick(FPS)
    pygame.display.flip()
```