

# Clase GameWorld

---

```
GameWorld( width: int, height: int, title: str = 'MiniGameEngine',  
           bgcolor: str = 'gray', bgpic: str = None )
```

Constructor de la clase GameWorld que inicializa una instancia del mundo de juego.

Args:

width (int)	: Ancho de la ventana del juego.
height (int)	: Altura de la ventana del juego.
title (str, optional)	: Título de la ventana del juego (opcional, por defecto es "MiniGameEngine").
bgcolor (str, optional)	: Color de fondo de la ventana del juego (opcional, por defecto es "gray").
bgpic (str, optional)	: Ruta de la imagen de fondo de la ventana del juego (opcional, por defecto es None).

```
exitGame()
```

Finaliza el loop principal del juego

```
gameLoop(fps: int)
```

Inicia el loop principal del juego.

Args:

fps (int)	: Fotogramas por segundo del juego.
-----------	-------------------------------------

```
getWorldHeight() -> int
```

Obtiene la altura del mundo de juego.

Returns:

int	: Altura del mundo de juego.
-----	------------------------------

```
getWorldWidth() -> int
```

Obtiene el ancho del mundo de juego.

Returns:

int	: Ancho del mundo de juego.
-----	-----------------------------

```
isPressed(key_name: str) -> bool
```

Verifica si una tecla específica está presionada.

Args:

key_name (str)	: Nombre de la tecla a verificar.
----------------	-----------------------------------

Returns:

bool	: True si la tecla está presionada, False en caso contrario.
------	--

```
onUpdate(dt: float)
```

Llamada por cada ciclo dentro del loop (fps veces por segundo)

Args:

dt (float)	: Tiempo en segundos desde la última llamada
------------	--

# Clase GameObject

---

```
GameObject(x: int, y: int, imagePath: str, tipo: str = 'undef')
```

Constructor de la clase GameObject que inicializa un objeto en el mundo de juego.

Args:

x (int)	: Coordenada x inicial del objeto.
y (int)	: Coordenada y inicial del objeto.
imagePath (str)	: Ruta de la imagen del objeto.
tipo (str, optional)	: Tipo del objeto (opcional, por defecto es "undef").

```
destroy()
```

Elimina el objeto del mundo de juego.

```
getHeight() -> int
```

Obtiene la altura del objeto.

Returns:

int	: Altura del objeto.
-----	----------------------

```
getTipo() -> str
```

Obtiene el tipo del objeto.

Returns:

str	: Tipo del objeto.
-----	--------------------

```
getWidth() -> int
```

Obtiene el ancho del objeto.

Returns:

int	: Ancho del objeto.
-----	---------------------

```
getWorldHeight() -> int
```

Obtiene la altura del mundo de juego.

Returns:

int	: Altura del mundo de juego.
-----	------------------------------

```
getWorldWidth() -> int
```

Obtiene el ancho del mundo de juego.

Returns:

int	: Ancho del mundo de juego.
-----	-----------------------------

### **getX() -> int**

Obtiene la coordenada x actual del objeto.

Returns:  
int : Coordenada x del objeto.

### **getY() -> int**

Obtiene la coordenada y actual del objeto.

Returns:  
int : Coordenada y del objeto.

### **isPressed(key\_name: str) -> bool**

Verifica si una tecla específica está presionada.

Args:  
key\_name (str) : Nombre de la tecla a verificar.

Returns:  
bool : True si la tecla está presionada, False en caso contrario.

### **onCollision(dt: float, gobj)**

Llamado cuando el objeto colisiona con otro objeto.

Args:  
dt (float) : Tiempo en segundos desde la última llamada.  
gobj (GameObject) : Objeto con el que colisiona.

### **onUpdate(dt: float)**

Llamado en cada actualización del juego para el objeto.

Args:  
dt (float) : Tiempo en segundos desde la última llamada.

### **setPosition(x: int, y: int)**

Establece la posición del objeto en el mundo de juego.

Args:  
x (int) : Nueva coordenada x del objeto.  
y (int) : Nueva coordenada y del objeto.

### **setShape(imagePath: str)**

Cambia la forma del objeto reemplazando su imagen.

Args:  
imagePath (str) : Ruta de la nueva imagen del objeto.

## Clase TextObject

---

```
__init__(self, x: int, y: int, text: str, font: str = 'Arial', size: int = 10, bold: bool = False, italic: bool = False, color: str = 'black')
```

Constructor de la clase TextObject que agrega un Texto al mundo del juego

Args:

x (int)	: Coordenada x del texto
y (int)	: Coordenada y del texto
text (str)	: Texto para este objeto
font (str, optional)	: Font a utilizar para el texto (opcional, por defecto es "Arial").
size (int, optional)	: Tamano a utilizar para el texto (opcional, por defecto es 10).
bold (bool, optional)	: Especifica que el texto estara en bold (opcional, por defecto es False).
italic (bool, optional)	: Especifica que el texto estara en italic (opcional, por defecto es False).
color (str, optional)	: Color a utilizar para el texto (opcional, por defecto es "black").

```
destroy(self)
```

Elimina este texto del mundo del juego

```
setText(self, x: int = None, y: int = None, text: str = None, font: str = None, size: int = None, bold: bool = None, italic: bool = None, color: str = None)
```

Modifica el texto desplegado y sus atributos. Si no se especifican atributos se conservan los existentes

Args:

x (int, optional)	: Coordenada x del texto.
y (int, optional)	: Coordenada y del texto
text (str, optional)	: Texto para este objeto
font (str, optional)	: Font a utilizar para el texto
size (int, optional)	: Tamano a utilizar para el texto
bold (bool, optional)	: Especifica que el texto estara en bold
italic (bool, optional)	: Especifica que el texto estara en italic
color (str, optional)	: Color a utilizar para el texto