



ELEMENTOS GRAFICOS: ESCENAS Y SPRITES

Entramos ahora en el terreno de los artistas. Para ello abriremos la cajita de herramientas de Python y buscaremos algunas *bibliotecas* que nos permitirán importar algunas funciones que necesitaremos.

La principal biblioteca con la que trabajaremos se llama **Spyral** y fue diseñada para la XO, aunque permite hacer juegos para casi cualquier computadora. Puedes consultar la documentación de Spyral en la dirección: <http://platipy.org/>

Spyral es lo que se conoce como un “motor de juegos”. Un motor de juegos principalmente se encarga de llevar la cuenta de los *elementos gráficos* que necesitan ser dibujados en la *pantalla*.

Para llevar a cabo su tarea, Spyral define algunos *objetos* que se pueden armar para construir cualquier juego en 2D. Estos *objetos* son estructuras de datos que pueden tener varios *atributos* y *funciones* asociadas.

Por ejemplo empecemos definiendo la pantalla. Para poder utilizar la pantalla primero es necesario crear una “escena”. Por ejemplo, un menú principal podría ser una escena, una pantalla de opciones podría ser otra, y debe haber al menos una escena que contenga el juego como tal.

Ejemplo para definir una escena:

```
import spyral

class Pong(spyral.Scene):
    def __init__(self):
        spyral.Scene.__init__(self)
```

El código anterior solamente define la escena Pong, pero no la hemos ejecutado todavía. Para ello es necesario llamar a la clase.

Ahora que la clase Pong ya está definida, la podemos llamar así: **mi_escena = Pong()**

Cuando *instanciamos* a una clase de esta forma, varias cosas suceden:

- Primero Python crea un nuevo objeto en memoria. Llamaremos a este objeto la variable *self*.
- Luego Python ejecuta la función especial `__init__`, dándole como primer parámetro la variable *self*.
- Si al momento de llamar a la clase lo hacemos con parámetros, es decir `Pong(valor1, valor2)`, entonces estos parámetros serán entregados a `__init__` de la forma `__init__(self, valor1, valor2)`.
- Python entonces *devolverá* el objeto nuevo *self* el cual será asignado a la variable *mi_escena*.
- La función especial `__init__` se llama el *constructor* de la clase.

Podemos tomar como ejemplo la clase `spyral.Image` que viene incluida con Spyral. Con esta clase podemos crear todas las imágenes que queramos de la forma:

```
mi_imagen = spyral.Image("nombre_de_archivo_de_imagen.png")
```

Luego se puede asignar este valor a, por ejemplo, la escena o cualquier otro objeto que sepa usar imágenes.

Si la imagen no corresponde al tamaño de la escena, se puede escalar con la función **scale** del objeto `Image`.

```
mi_imagen.scale(mi_escena.size)
mi_escena.background = mi_imagen
```

Con esto tendremos un lindo fondo de escena. La clase `Image` soporta archivos `.png`, `.jpg`, `gif` entre otros. También se puede hacer una imagen vacía, dándole un tamaño con el parámetro `size`, así:

```
mi_imagen = spyral.Image(size=(100,100))
```

Atributos principales de <code>spyral.Scene</code>		
Atributo	Tipo	Descripción
<code>background</code>	<code>Image</code>	Imagen de fondo
<code>width *</code>	<code>int</code>	El ancho de la pantalla
<code>height *</code>	<code>int</code>	La altura de la pantalla
<code>size *</code>	<code>Vec2D</code>	El tamaño de la pantalla, es decir (<code>width</code> , <code>height</code>)



El director de la escena

Una vez que hemos definido nuestra escena necesitamos avisarle a Spyral que la escena está lista para ser reproducida. Para ello usaremos el objeto `spyral.Director` de la siguiente manera:

```
spyral.director.push(mi_escena)
```

La última escena siempre es la activa, y se puede regresar a la anterior usando `spyral.director.pop()`

Estoy confundido... ¿para qué sirven las clases?



Es normal confundirse un poco en este punto. Esto es porque estamos aprendiendo las bases de la “programación orientada a objetos”. La idea es que a medida que un programa se hace más y más complejo, por ORDEN y CONVENIENCIA, es útil agrupar las funciones y las variables que se relacionan entre sí dentro de clases, las cuales representan un tipo de objeto.

Veamos otro ejemplo para que nos quede más claro. Ya definimos cómo lucirá la pantalla creando la clase Pong, que es una escena. Ahora aprenderemos a colocar elementos gráficos en ella. El nombre técnico de los elementos gráficos que se pueden mover por la pantalla es *Sprite*:

Atributos principales de <code>spyral.Sprite</code>		
Atributo	Tipo	Descripción
pos	Vec2D	La posición en la que el objeto es dibujado, es decir (x,y)
x	int	La posición horizontal
y	int	La posición vertical
anchor	str	El “ancla”, es decir el punto referencia dentro de la imagen
image	Image	La imagen determina qué se dibujará para este objeto
visible	bool	Determina si el objeto se dibuja o no
width *	int	El ancho actual de la imagen
height *	int	La altura actual de la imagen
size *	Vec2D	El tamaño actual de la imagen
scale	Vec2D (acepta float)	La escala de la imagen
scale_x	float	La escala horizontal de la imagen
scale_y	float	La escala vertical de la imagen
flip_x	bool	Determina si la imagen está volteada verticalmente
flip_y	bool	Determina si la imagen está volteada horizontalmente
angle	float	Ángulo en que la imagen se muestra (en radianes)

Entonces vamos a definir nuestro propio Sprite el cual va a representar una pelota:

```
class Pelota (spyral.Sprite):
    def __init__(self, scene):
        spyral.Sprite.__init__(self, scene)

        self.image = spyral.Image(size=(20, 20))
        self.image.draw_circle((255, 255, 255), (10, 10), 10)
        self.anchor = 'center'
```



Entonces, para añadir nuestra pelota a la escena tenemos que hacer así:

```
mi_pelota = Pelota ( mi_escena )
```

