

UN JUEGO PASO A PASO

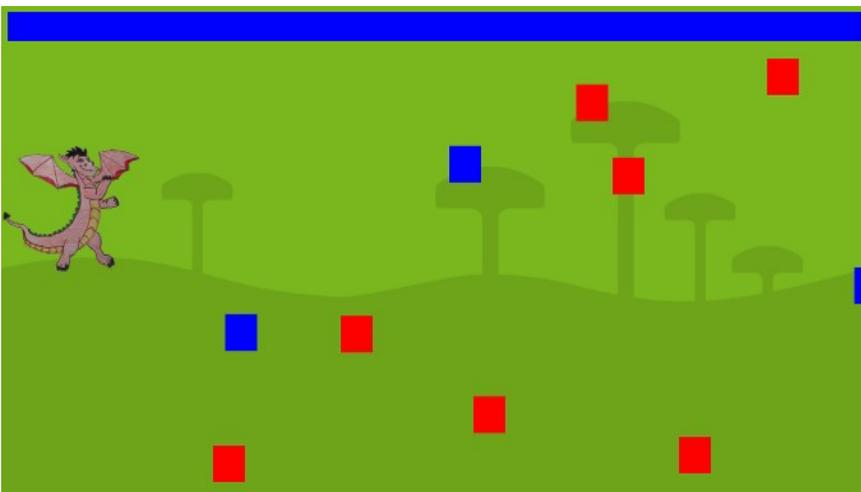
Para realizar un juego lo primero que debes hacer es imaginarlo. Debes identificar cada elemento que participa en tu juego. Por ejemplo: ¿hay un personaje principal? ¿enemigos? ¿objetos? ¿premios? ¿vidas extra? ¿indicadores en pantalla? ¿el fondo de tu juego se desplaza?

Un diseño sencillo

Hemos pensado en un juego sencillo:

La idea del juego es que hay un **dragón rojo** que avanza hacia la derecha (en realidad lo que se mueve es el **fondo**).

El dragón se llama *Comodo* y fue diseñado por Cristian Camilo Aguirre.



Comodo puede volar al presionar la barra espaciadora. En su camino encontrará **obstáculos**, que al tocarlos le producen daño y también podrá encontrar **premios** que le restituirán la energía. Ellos están representados por cuadrados rojos y azules respectivamente.

La energía vital de Comodo está representada por la barra azul que se muestra en la parte superior de la pantalla. A

medida que va bajando la energía, ella va disminuyendo de tamaño. Si la barra llega a cero, el juego termina.

A continuación detallaremos cada elemento.

Comodo

<i>Acción</i>	<i>Cuándo realiza la acción</i>
Salta	Cuando se presiona la barra.
Cae	Cuando no está saltando o tocando el piso.

Obstáculos

<i>Acción</i>	<i>Cuándo realiza la acción</i>
Se desplaza hacia la izquierda.	Siempre
Daña a Comodo y desaparece.	Cuando colisiona con Comodo.

Premios

<i>Acción</i>	<i>Cuándo realiza la acción</i>
Se desplaza hacia la izquierda.	Siempre
Sana a Comodo y desaparece.	Cuando colisiona con Comodo.

Barra

<i>Acción</i>	<i>Cuándo realiza la acción</i>
Se agranda	Cuando aumenta la vida de Comodo
Se reduce	Cuando disminuye la vida de Comodo

Fondo

<i>Acción</i>	<i>Cuándo realiza la acción</i>
Se desplaza hacia la izquierda.	Siempre

Manos a la obra

Como puedes ver, incluso un juego sencillo como el descrito anteriormente tiene varios elementos que necesitan ser detallados para poder realizarse. Siempre es conveniente tener el diseño claro y definido antes de empezar a escribir el código.

Basicamente cualquier cosa que en tu diseño se mueva o esté animada es considerada un elemento de juego y requiere su propia clase tipo `spyral.Sprite`.

Entonces escribimos una clase `Sprite` asociada a **cada elemento identificado del juego**.

A continuación encontrarás un ejemplo del prototipo de clase `Comodo`:

```
class Comodo(spyral.Sprite):
    def __init__(self, scene):
        spyral.Sprite.__init__(self, scene)
        self.image = spyral.Image(filename="images/comodo.png")

    def salta(self):
        pass

    def cae(self, lado):
        pass
```

Las clases agrupan todas las acciones posibles que cada elemento del juego puede realizar. Cada acción posible está definida como una función (método **def**). Mientras no hemos *implementado* cada acción, conviene de todas maneras crear la función correspondiente. La sentencia **pass** no realiza ninguna acción, tan solo permite que se pueda crear una función vacía.

Lo que hemos hecho hasta aquí es únicamente declarar la clase, pero no la hemos utilizado aún. Es decir, no se ha ejecutado todavía ninguna línea de código, sólo hemos informado a Python que existe

la clase **Comodo**.

Para hacer que de hecho Comodo aparezca en pantalla, es necesario crear una variable a partir de la clase Comodo, de la siguiente forma:

```
comodo1 = Comodo(escena)
```

Este código lo que hace es *instanciar* una variable de tipo *Comodo* la cual tendrá las características de la clase que hemos definido anteriormente. Cuando uno *instancia* una clase, lo primero que se ejecuta es la función especial **__init__** que es conocida como el *constructor de la clase*.

Nota!

*Para poder instanciar (crear) un objeto de la clase Comodo, es necesario pasarle una variable de tipo `spiral.Scene`. Para los juegos hechos en el "taller del artesano", la escena principal se suele llamar **self.game**.*

Una vez que hemos creado la variable, es posible llamar a las funciones definidas de la clase *Comodo*:

```
comodo1.salta()  
comodo1.cae()
```

También es posible llamar a las funciones definidas por la clase *Sprite* (porque *Comodo* es una clase derivada de la clase *spiral.Sprite*)

Por lo tanto las siguientes operaciones son posibles:

```
comodo1.scale = 0.5  
comodo1.pos = (500, 500)  
comodo1.animate(animacion)  
comodo1.kill()
```

Las primeras dos son asignaciones de *variables* de la clase, que se llaman **atributos**. Las últimas dos son *funciones* de la clase, se llaman **métodos**. Para que la función de animación funcione correctamente es necesario pasarle una variable de tipo *spiral.Animation*. Puedes consultar la documentación de todos los objetos de *Spyral* en la página web <http://platipy.org/>.

Es utilizando estos métodos que podrás programar las acciones que cada uno de los elementos será capaz de realizar. Por ejemplo la función saltar terminó siendo así:

```
def salto(self):  
    self.stop_all_animations()  
    self.estado = "saltando"  
    animacion = spiral.Animation("y", spiral.easing.CubicOut(self.y, self.y-  
100), duration=1)  
    self.animate(animacion)
```

Nota!

*Los métodos de las clases son funciones especiales cuyo primer parámetro se llama **self**. Cada vez que uno invoca a una de estas funciones, automáticamente Python define este parámetro para que haga referencia a la instancia de la clase que está llamando al método. En el ejemplo anterior, **self = comodo1**.*