PyGame: Desarrollo de juegos en Python

Jesús Espino García

6 de Septiembre de 2008

1. Introducción
2. Elementos
3. Estructura básica
4. Incorporando objetos
5. Sprites
6. Para terminar

1 Introducción

2 Elementos

3 Estructura básica

4 Incorporando objetos

5 Sprites

6 Para terminar

¿Qué es?

- Es una biblioteca SDL de python.
- Simplifica el uso de SDL.
- Esta orientada al desarrollo de juegos.

Ventajas e inconvenientes

Ventajas:

- Es específico para crear juegos.
- Da capacidades multimedia de forma sencilla.
- Es Python!!
- Es rápido de programar.
- Es LGPL.

Desventajas:

- Es específico para crear juegos.
- Es Python!!
- No es especialmente rápido.

1 Introducción

2 Elementos

3 Estructura básica

4 Incorporando objetos

5 Sprites

6 Para terminar

Elementos

En el diseño de juegos con pygame, tendremos 4 elementos básicos.

- La ventana principal (Display)
- Las superficies (Sufaces)
- Los eventos (Events)
- Los rectángulos (Rects)

La ventana principal

Esto nos va a permitir el control sobre la ventana principal con operaciones como...

- `Info()`: Obtiene información sobre la configuración gráfica.
- `set_mode()`: Establece las dimensiones de la ventana.
- `set_caption()`: Establece el título de la ventana.
- `set_icon()`: Establece el icono de la ventana.
- `flip()`: Actualiza el contenido de la ventana.
- `update()`: Actualiza el contenido de la ventana solo en algunas partes.

Además tiene una superficie que engloba todo el contenido de la ventana.

Superficies

Son elementos que almacenan las imágenes que vamos a mostrar.

- `convert()`: Convierte nuestra superficie al formato final que usaremos, lo que hace que trábajar con el sea más rápido.
- `blit()`: Copia una superficie a una posición.
- `copy()`: Copia una superficie.
- `fill()`: Rellena una superficie con un color especificado.
- `get_rect()`: Obtiene un objeto Rect asociado a la superficie.
- `subsurface()`: Crea una superficie con datos compartidos.

Eventos

Gestiona la interacción del usuario con el juego.

Rectángulos

Es un objeto que nos permite trabajar cómodamente con rectángulos que representan las posiciones de nuestras superficies y nuestros objetos de juego. Tiene varios métodos muy útiles.

- `move()`: Mueve la posición en referencia a la actual.
- `colliderect()`: Comprueba si hay una colisión con otro rectángulo.
- `union()`: Devuelve un Rect que engloba a dos elementos que se le pasen como parametro.
- `unionall()`: Devuelve un Rect que engloba a todos los elementos que se le pasen como parámetro.

Además tiene algunos atributos que nos sirven para modificar la posición de nuestra superficie o objeto de juego.

- `centerx,centery`: El centro respecto de x o de y.
- `left,right,top,bottom`: El borde izquierdo, derecho, superior o inferior.

1 Introducción

2 Elementos

3 Estructura básica

4 Incorporando objetos

5 Sprites

6 Para terminar

Estructura básica

- Importar módulos
- Inicializamos PyGame
- Inicialización del escenario de juego (Display y Surfaces)
- Crear el bucle de eventos (Events)
  - Procesar evento capturado (Events)
  - Detectamos y procesamos situaciones del juego.
  - Redibujar lo que sea necesario (Display y Surfaces)

Ejemplo: Hola mundo

1 Introducción

2 Elementos

3 Estructura básica

4 Incorporando objetos

5 Sprites

6 Para terminar

Como utilizar textos en pygame

Para trabajar con textos pygame nos da la clase Font.

- `pygame.font.Font()`: nos crea un objeto font con la fuente y el tamaño pasados por parametro.
- `render()`: Nos devuelve una superficie con el texto y el color que le pasamos como parametro. Tambien se puede establecer si se usa o no antialiasing.
- `set_bold()`: Pone el texto en negrita.
- `set_italic()`: Pone el texto en itálica.
- `set_underline()`: Pone el texto subrayado

Como utilizar imágenes en pygame

Para trabajar con imágenes pygame nos da algunas funciones.

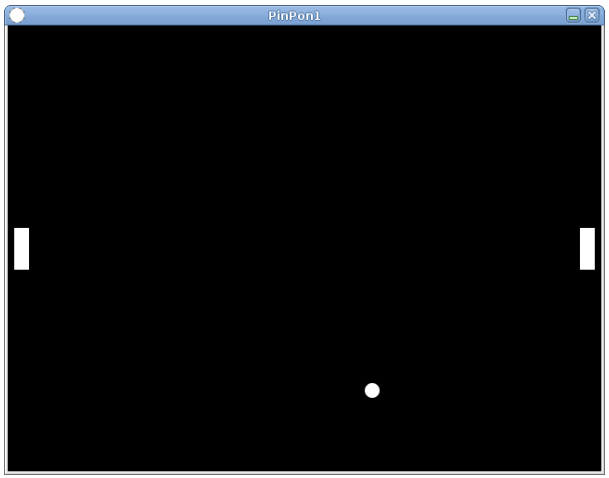
- `pygame.image.load()`: Nos devuelve una superficie con la imagen que le hemos pasado.
- `pygame.image.save()`: Guarda una un fichero en formato TGA o BMP la superficie que se le pasa como parámetro.
- `pygame.image.tostring()`: Pasa una superficie a una cadena de texto (para poder pasar imágenes entre programas).
- `pygame.image.fromstring()`: Pasa una cadena a una superficie.

Como utilizar sonido en pygame

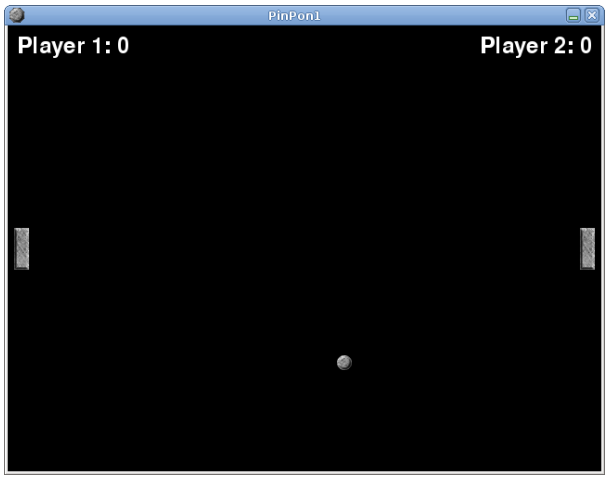
Para trabajar con sonidos pygame nos da la clase Sound

- `pygame.mixer.Sound()`: Nos devuelve un objeto de tipo sound haciendo referencia al fichero que se le pasa como parámetro.
- `play()`: Este método reproduce el sonido, se le puede especificar el numero de veces y el tiempo máximo para reproducir el sonido.
- `stop()`: Para todos las reproducciones de este sonido.
- `set_volume()`: Establece el volumen de reproducción del sonido, un valor entre 0.0 y 1.0.
- `get_length()`: Devuelve el numero de segundos de datos que tiene el fichero.

Ejemplo: Pin Pon



Ejemplo: Pin Pon con sonido y puntuación



1 Introducción

2 Elementos

3 Estructura básica

4 Incorporando objetos

5 Sprites

6 Para terminar

¿Qué son?

- Serán nuestros objetos de juego.
- Fáciles de manejar.
- Objeto compuesto de una imagen y una posición.
- Se gestiona a si mismo.
- Conjunto de clases.

Clase Group

- Contenedor de sprites.
- Permite realizar operaciones conjuntas.
- Permite detectar colisiones entre grupos.

Uniendo ambas clases I

Los objetos Sprite tienen métodos para trabajar con los grupos.

- `__init__`: En el constructor se le puede especificar a que grupos pertenece en el momento de su creacion.
- `add()`: Añade el sprite a un grupo.
- `remove()`: Elimina el sprite de un grupo.
- `kill()`: Elimina el sprite de todos los grupos.
- `alive()`: Nos devuelve true si el sprite pertenece todavía a algún grupo.

Uniendo ambas clases II

Los objetos Group tienen métodos para trabajar con los sprites.

- `__init__`: En el constructor se puede especificar que sprites pertenecen en el momento de su creación.
- `add()`: Añade un sprite al grupo.
- `remove()`: Elimina un sprite del grupo.
- `empty()`: Vacía el grupo.
- `copy()`: Devuelve una copia del grupo con todos sus miembros.
- `has()`: Comprueba si pertenecen al grupo los sprites especificados.
- `sprites()`: Devuelve una lista de los sprites del grupo.
- `update()`: Ejecuta el método `update()` en cada sprite del grupo (por defecto un método vacío).
- `len()`: Devuelve el numero de sprites del grupo.

Uniendo ambas clases III

Cave destacar varias razones para usar las clases Sprite y Group conjuntamente.

- Comodidad: Se pueden hacer operaciones sobre los grupos en vez de sobre cada sprite.
- Eficiencia: Tiene poco coste añadir y quitar un sprite de un grupo.
- Orden: Nuestro código y nuestras ideas estarán más ordenadas.

Tipos de grupos I

Hay varios tipos de grupos que se comportan de manera diferente.

- `Group`
- `GroupSigle`
- `RenderPlain`
- `RenderClear`:
- `RenderUpdates`:

Tipos de grupos II: Group

- Este es el grupo estándar.
- Todos los demás grupos derivan de él añadiendo funcionalidad.

Tipos de grupos III: GroupSigle

- Hereda de Group.
- Solo almacena un srpite.
- Cuando se inserta uno olvida el anterior.

Tipos de grupos IV: RenderPlain

- Hereda de Grupo.
- Añade el método draw() que dibuja todos los sprites que contiene.
- Los sprites necesitan tener los atributos image y rect para saber que y donde dibujar.

Tipos de grupos V: RenderClear

- Hereda de RenderPlain,
- Añade el método clear() que limpia la posición anterior de los sprites.
- Utiliza una imagen de fondo para rellenar las posiciones antiguas.

Tipos de grupos VI: RenderUpdates

- Hereda del grupo RenderClear.
- Modifica el método draw() para que devuelva una lista de rects con las posiciones que han cambiado.

Tipos de grupos VII

Es necesario saber que tipo de Render Group utilizar.
Para juegos con fondos muy dinámicos es mejor usar RenderPlain, porque no se obtendrá las ventajas que nos proporciona RenderUpdates pero si su consumo de recursos, que es menor. En cambio, para juegos con fondo estático sera mejor usar RenderUpdates que nos facilitara el trabajo y probablemente lo hará mejor que nosotros.
Se pueden mezclar en un mismo juego los grupos que se quieran.

Detección de colisiones

Una de las cosas importantes en los juegos es la detección de las colisiones, pygame nos ofrece dos funciones.

- `spritecollide(sprite,group,dokill)->list`: Nos devuelve una lista de los sprites que se solapan con el sprite indicado. dokill es un parámetro booleano que si es cierto envía una señal kill a todos los sprites de la lista.
- `groupcollide(group1,group2,dokill1,dokill2)->dictionary`: Nos devuelve un diccionario con los sprites del grupo uno que colisionan con los sprites del grupo dos, si dokill1 es verdadero enviara la señal kill a todos los sprites del diccionario que pertenezcan al grupo 1, si dokill2 es verdadero hará lo mismo con los del grupo 2.

Estas funciones son básicas y probablemente para juegos complejos sea necesario redefinirlas.

El bucle de eventos

El bucle de eventos nos permite gestionar los eventos de nuestro juego. Pulsaciones de tecla, clicks del ratón, movimientos del joystick, incluso definir nuestros propios eventos. Las funciones que se utilizan para control de eventos son:

- `get()`: Obtiene la lista de eventos que están en espera.
- `wait()`: Espera a la aparición de algún evento.
- `clear()`: Elimina todos los elementos que están en espera.
- `post()`: Mete un elemento en espera.
- `poll()`: Saca un elemento de la espera.
- `set_allowed()`: Permite que elementos se pongan en espera.
- `set_blocked()`: Bloquea que elementos se pongan en espera.

Eventos existentes

Existen algunos elementos predefinidos:

- QUIT
- ACTIVEEVENT
- KEYDOWN
- KEYUP
- MOUSEMOTION
- MOUSEBUTTONDOWN
- MOUSEBUTTONUP
- JOYAXISMOTION
- JOYBALLMOTION
- JOYHATMOTION
- JOYBUTTONDOWN
- JOYBUTTONUP
- VIDEORESIZE
- VIDEOEXPOSE
- USEREVENT

Ejemplo de bucle de eventos

```
for event in pygame.event.wait():
    if event == QUIT:
        return
```

Una aplicación con sprites, ejemplo del space invaders.

Otra aplicación con sprites, ejemplo del air shutter.

1 Introducción

2 Elementos

3 Estructura básica

4 Incorporando objetos

5 Sprites

6 Para terminar

Referencias

- http://www.pygame.org
- irc://irc.freenode.org/pygame
- Lista de correo: pygame-users@seul.org
- Beginning Game Development with Python and Pygame - Apress

Dudas

. . .

Fin