12 – El módulo Pygame

Diego Andrés Alvarez Marín
Profesor Asociado
Universidad Nacional de Colombia
Sede Manizales

Pygame es un librería de Python diseñada para escribir juegos **2D**, desarrollada por Pete Shinners. El desarrollo de Pygame empezó en el año 2000. Está basada en una librería escrita en lenguaje C llamada SDL https://www.libsdl.org/. Pygame permite acceso a teclado, mouse, joystick, sonido, funciones para el manejo de imágenes, animaciones, etc.

Se puede descargar de: http://www.pygame.org/

- Pygame does not work well with the interactive shell because it relies on a game loop (we will describe game loops later). Because of this, you can only write Pygame programs and cannot send instructions to Pygame one at a time through the interactive shell.
- Pygame programs also do not use the input() function. There is no text input and output. Instead, the program displays output in a window by drawing graphics and text to the window.
- Pygame program's input comes from the keyboard and the mouse through things called events.

Módulos incluidos en Pygame

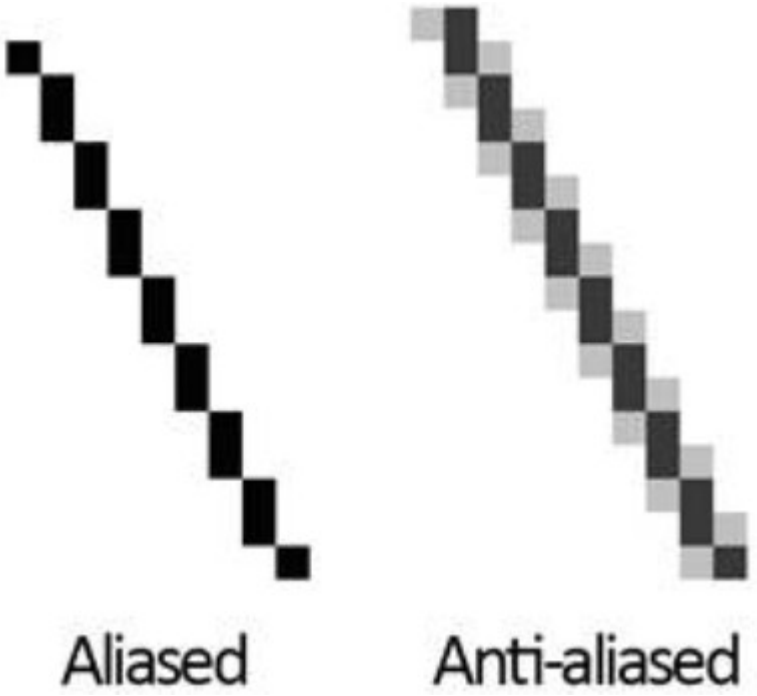
• pygame.cdrom	How to access and control the CD audio devices.
• pygame.cursors	Loading and compiling cursor images.
• pygame.display	Configure the display surface.
• pygame.draw	Draw simple shapes like lines, circles and ellipses
• pygame.event	Manage the incoming events
• pygame.font	Loading and rendering Truetype fonts.
• pygame.freetype	Module for loading and rendering font faces.
• pygame.gfxdraw	Anti-aliasing draw functions.
• pygame.image	Loading, saving, and transferring of surfaces.
• pygame.joystick	Manage the joystick devices.
• pygame.key	Manage the keyboard device.
• pygame.locals	Pygame constants.
• pygame.mixer	Load and play sounds
• pygame.mouse	Manage the mouse device and display.
• pygame.movie	Video playback from MPEG movies.
• pygame.mixer.music	Play streaming music tracks.
• pygame.scrap	Native clipboard access.
• pygame.sndarray	Manipulate sound sample data.
• pygame.sprite	Higher level objects to represent game images.
• pygame.surfarray	Manipulate image pixel data.
• pygame.time	Manage timing and framerate.
• pygame.transform	Resize and move images.

Módulos vs tipos de datos

En Pygame:

- el nombre de los módulos comienza en minúscula (pygame.font, pygame.mouse, etc);
- dentro de los módulos se definen nuevos tipos de datos u objetos. Los nombres de los tipos de datos comienzan con una letra mayúscula (pygame.Surface, pygame.Rect, pygame.Font).

Línea con y sin aliasing



El aliasing crea líneas/texto etc difuminado de modo que se ve mucho más agradable a la vista y menos pixelada. Si se utiliza el modo con antialiasing, el programa correrá un poco más lento, pero las gráficas se verán mejor.

Los objetos pygame.Rect

- Se utilizan para representar el tamaño y la posición de una región rectangular.
- Se crean con el método pygame.Rect(). Los parámetros son las coordenadas X,Y del borde superior izquierdo, seguido del ancho y del alto (todo en pixeles):

```
pygame.Rect(left, top, width, height)
```

- Los objetos Rect tienen varios atributos, que se listan a continuación:

Lista de atributos asociados a un objeto de tipo Rect llamado myRect

pygame.Rect Attribute	Description
`myRect.left`	Integer value of the X-coordinate of the left side of the rectangle.
`myRect.right`	Integer value of the X-coordinate of the right side of the rectangle.
`myRect.top`	Integer value of the Y-coordinate of the top side of the rectangle.
`myRect.bottom`	Integer value of the Y-coordinate of the bottom side of the rectangle.
`myRect.centerx`	Integer value of the X-coordinate of the center of the rectangle.
`myRect.centery`	Integer value of the Y-coordinate of the center of the rectangle.
`myRect.width`	Integer value of the width of the rectangle.
`myRect.height`	Integer value of the height of the rectangle.
`myRect.size`	A tuple of two integers: (width, height)
`myRect.topleft`	A tuple of two integers: (left, top)
`myRect.topright`	A tuple of two integers: (right, top)
`myRect.bottomleft`	A tuple of two integers: (left, bottom)
`myRect.bottomright`	A tuple of two integers: (right, bottom)
`myRect.midleft`	A tuple of two integers: (left, centery)
`myRect.midright`	A tuple of two integers: (right, centery)
`myRect.midtop`	A tuple of two integers: (centerx, top)
`myRect.midbottom`	A tuple of two integers: (centerx, bottom)

Los objetos pygame.Rect

Cuando se modifica uno de los atributos, los otros se ajustan automáticamente.

Por ejemplo, si se hace pygame.Rect(30,40,20,20), entonces la coordenada del borde inferior derecho automáticamente se ajustará a (50, 60).

O si se cambia la coordenada de la izquierda con myRect.left = 100, entonces automáticamente se cambiarán el resto de coordenadas.

pygame.Rect

class **pygame.Rect**

pygame object for storing rectangular coordinates

```
Rect(left, top, width, height) -> Rect
Rect((left, top), (width, height)) -> Rect
Rect(object) -> Rect
```

pygame.Rect.copy	—	copy the rectangle
pygame.Rect.move	—	moves the rectangle
pygame.Rect.move_ip	—	moves the rectangle, in place
pygame.Rect.inflate	—	grow or shrink the rectangle size
pygame.Rect.inflate_ip	—	grow or shrink the rectangle size, in place
pygame.Rect.clamp	—	moves the rectangle inside another
pygame.Rect.clamp_ip	—	moves the rectangle inside another, in place
pygame.Rect.clip	—	crops a rectangle inside another
pygame.Rect.union	—	joins two rectangles into one
pygame.Rect.union_ip	—	joins two rectangles into one, in place
pygame.Rect.unionall	—	the union of many rectangles
pygame.Rect.unionall_ip	—	the union of many rectangles, in place
pygame.Rect.fit	—	resize and move a rectangle with aspect ratio
pygame.Rect.normalize	—	correct negative sizes
pygame.Rect.contains	—	test if one rectangle is inside another
pygame.Rect.collidepoint	—	test if a point is inside a rectangle
pygame.Rect.colliderect	—	test if two rectangles overlap
pygame.Rect.collidelist	—	test if one rectangle in a list intersects
pygame.Rect.collidelistall	—	test if all rectangles in a list intersect
pygame.Rect.collidedict	—	test if one rectangle in a dictionary intersects
pygame.Rect.collidedictall	—	test if all rectangles in a dictionary intersect

Colisiones

Se detectan con el método pygame.Rect.colliderect(); esta verifica si dos objetos en la pantalla se están tocando mutuamente. Ejemplos: si un jugador toca un enemigo puede perder la vida. El programa necesita saber cuando el jugador toca una moneda para automáticamente recogerla. La detección de colisiones puede ayudar a determinar si el personaje del juego está de pie en tierra firme o si no hay nada más que aire vacío debajo de ellos.

Examples of colliding rectangles (left) and rectangles that don't collide (right).

The game loop

- http://en.wikipedia.org/wiki/Game_programming
-

```
while( user doesn't exit )
  check for user input
  run AI
  move enemies
  resolve collisions
  draw graphics
  play sounds
end while
```

Los objetos pygame.Surface

- Son áreas rectangulares que representan un conjunto de pixeles (los objetos Rect solo representan la ubicación y tamaño de un espacio rectangular).
- El método blit() se toma la imagen que está en un objeto Surface y le hace “copiar y pegar” a otro objeto Surface. En verdad lo que hace es cambiar (o copiar) el color de los pixeles.
- El objeto Surface que retorna la función pygame.display.set_mode() es especial, porque cualquier cosa dibujada sobre ella se muestra en la pantalla cuando se invoca el comando pygame.display.update().

Las imágenes (o sprites)

- Algunos tipos de imagen soportados son .bmp, .png, .jpg y .gif.
- Se cargan de forma similar a:

```
player = pygame.image.load('player.bmp').convert()
```

- pygame.image.load() retorna un objeto Surface.
- convert() recibe un objeto Surface y retorna otro objeto Surface con un formato interno en la representación de pixeles similar al utilizado en la pantalla (display); esto se hace con el objeto de incrementar la velocidad de la función blit(), ya que no tiene que convertir de un tipo de pixel a otro.

Las imágenes (o sprites)

- En el módulo pygame.image ...
- En el módulo pygame.transform hay muchos métodos para transformar imágenes: rotarla, escalarla, etc.
- Muchas transformaciones sobre la misma imagen la degradan. Por esta razón, siempre que se realicen transformaciones sobre una imagen, estas se deben hacer sobre la imagen original.

¿Cómo se mueve una imagen?

Para mover una imagen en la pantalla, simplemente se hace un blit() de la imagen en la nueva posición, teniendo en cuenta en borrar los pixeles que ocupaban la posición anterior. La forma más fácil de hacer esto es mantener una copia separada del fondo de la pantalla (background).

```
screen = pygame.display.set_mode((640, 480))
player = pygame.image.load('player.bmp').convert()
background = pygame.image.load('background.bmp').convert()
screen.blit(background, (0, 0))
objects = []
for x in range(10):                    #create 10 objects
   o = GameObject(player, x*40, x)
   objects.append(o)
while 1:
   for event in pygame.event.get():
       if event.type in (QUIT, KEYDOWN):
           sys.exit()
   for o in objects:
       screen.blit(background, o.pos, o.pos)
   for o in objects:
       o.move()
       screen.blit(o.image, o.pos)
   pygame.display.update()
   pygame.time.delay(100)
```

pygame.event.Event

- Son objetos generados por pygame cada vez que el usuario presiona una tecla, hace click o mueve el mouse.
- La función pygame.event.get() retorna una lista de estos objetos Event.
- Se puede determinar el tipo de evento al comparar su propiedad type con las constantes QUIT, KEYDOWN, MOUSEBUTTONUP, etc. (definidas en pygame.locals)

```
while True:
    for event in pygame.event.get():
        if event.type in (QUIT, KEYDOWN):
            pygame.quit()
            exit()
    mueva_y_dibuje_todos_los_objetos_del_juego()
```

Table 18-1: Events and when they are generated.

Event Type	Description
QUIT	Generated when the user closes the window.
KEYDOWN	Generated when the user presses down a key. Has a `key` attribute that tells which key was pressed. Also has a `mod` attribute that tells if the Shift, Ctrl, Alt, or other keys were held down when this key was pressed.
KEYUP	Generated when the user releases a key. Has a `key` and `mod` attribute that are similar to those for KEYDOWN.
MOUSEMOTION	Generated whenever the mouse moves over the window. Has a `pos` attribute that returns tuple (x, y) for the coordinates of where the mouse is in the window. The `rel` attribute also returns a (x, y) tuple, but it gives coordinates relative since the last MOUSEMOTION event. For example, if the mouse moves left by four pixels from (200, 200) to (196, 200), then `rel` will be the tuple value (-4, 0). The `buttons` attribute returns a tuple of three integers. The first integer in the tuple is for the left mouse button, the second integer for the middle mouse button (if there's a middle mouse button), and the third integer is for the right mouse button. These integers will be 0 if they are not being pressed down when the mouse moved and 1 if they are pressed down.
MOUSEBUTTONDOWN	Generated when a mouse button is pressed down in the window. This event has a `pos` attribute which is an (x, y) tuple for the coordinates of where the mouse was when the button was pressed. There is also a `button` attribute which is an integer from 1 to 5 that tells which mouse button was pressed, explained in Table 18-2.

MOUSEBUTTONUP	Generated when the mouse button is released. This has the same attributes as MOUSEBUTTONDOWN.

Table 18-2: The button attribute values and mouse button.

Value of button	**Mouse Button**
1	Left button
2	Middle button
3	Right button
4	Scroll wheel moved up
5	Scroll wheel moved down

Mouse y teclado

Más información sobre el teclado:

https://www.pygame.org/docs/ref/key.html

Más información sobre el mouse:

https://www.pygame.org/docs/ref/mouse.html

pygame.font y pygame.freetype

El módulo pygame.font tiene el tipo de dato fuente que se utiliza para escribir el texto en Pygame.

pygame.font.SysFont(nombre_fuente, tamaño)

A veces se pone nombre_fuente=None para utilizar la fuente por defecto en el sistema. El módulo utiliza fuentes TypeTrue (*.ttf).

El módulo pygame.freetype es un módulo más avanzado que pygame.font, para el manejo de fuentes: https://www.pygame.org/docs/ref/freetype.html Uselo si requiere imprimir los diferentes Unicodes

pygame.time.Clock

- pygame.time.Clock - The Clock object in the pygame.time module is helpful for keeping our games from running as fast as possible. The Clock object has a tick() method, which we pass how many frames per second (FPS) we want the game to run at. The higher the FPS, the faster the game runs.

Sound

- The sound file formats that Pygame supports are MID, WAV, and MP3.
- En el módulo pygame.mixer están los métodos para trabajar con sonidos.

Algunos colores

Table 17-1: Colors and their RGB values.

Color	RGB Values
Black	(0, 0, 0)
Blue	(0, 0, 255)
Gray	(128, 128, 128)
Green	(0, 128, 0)
Lime	(0, 255, 0)
Purple	(128, 0, 128)
Red	(255, 0, 0)
Teal	(0, 128, 128)
White	(255, 255, 255)
Yellow	(255, 255, 0)