PYGAME **ESTIMULOS**

SSVEP · Adquisición de señales EEG

Funciones · Clases · Métodos

Pygame agrega funcionalidad sobre la excelente biblioteca SDL; es una biblioteca C multiplataforma para controlar multimedia, comparable a DirectX. Pygame es altamente portátil y se ejecuta en casi todas las plataformas y sistemas operativos. Pygame es gratis. Lanzado bajo la licencia LGPL, puede crear código abierto, freeware, shareware y juegos comerciales con él. Vea la licencia para más detalles.

INICIACIÓN

PYGAME.INIT()
Inicializar todos los módulos de pygame importados. Siempre puede inicializar módulos individuales manualmente, pero pygame.init() inicializa todos los módulos pygame importados, es una forma conveniente de comenzar todo. Las init() funciones para módulos individuales generarán excepciones cuando fallen.

PYGAME.GET_INIT( )
Devuelve True si pygame está actualmente inicializado

PYGAME.QUIT( )
Desinicializar todos los módulos de Pygame. Cuando el intérprete de Python se apaga, este método se llama independientemente, por lo que su programa no debería necesitarlo, excepto cuando quiere terminar sus recursos de pygame y continuar. Es seguro llamar a esta función más de una vez ya que las llamadas repetidas no tienen efecto.
Nota Llamar a pygame.quit() no saldrá del programa. Se debe finalizar de la misma manera que finalizará un programa Python normal.

EVENTO

PYGAME.EVENT()
Módulo de pygame para interactuar los eventos como colas. Pygame maneja todos sus mensajes de eventos a través de una cola de eventos. Las rutinas en este módulo lo ayudan a administrar esa cola de eventos. La cola de entrada depende en gran medida del módulo pygame.display().
La cola de eventos tiene un límite superior en la cantidad de eventos que puede contener (128 para SDL 1.2 estándar). Cuando la cola se llena, los nuevos eventos se descartan silenciosamente. Para evitar eventos perdidos, especialmente eventos de entrada que señalan un comando para salir, su programa debe verificar periódicamente los eventos y procesarlos. Para acelerar el procesamiento de la cola, use el pygame.event.set_blocked() control de los eventos permitidos en la cola para limitar los eventos.
Para obtener el estado de varios dispositivos de entrada, puede renunciar a la cola de eventos y acceder a los dispositivos de entrada directamente con sus módulos apropiados: pygame.mouse para trabajar con el mouse , pygame.key para trabajar con el teclado y pygame.joystick para interactuar con joysticks, gamepads y trackballs. Si utiliza este método, recuerde que pygame requiere alguna forma de comunicación con el administrador de ventanas del sistema y otras partes de la plataforma. Para mantener Pygame sincronizado

con el sistema, deberá llamar a pygame.event.pump()los controladores de eventos de Pygame de proceso interno para mantener todo actualizado.

PYGAME.EVENT.PUMP()
Esto garantiza que su programa pueda interactuar internamente con el resto del sistema operativo. Manejo de eventos individuales.

PYGAME.EVENT.GET()
Esto obtendrá todos los mensajes y los eliminará de la cola. Si se proporciona un tipo o secuencia de tipos, solo esos mensajes se eliminarán de la cola.

PYGAME.EVENT.POLL()
Devuelve un solo evento de la cola. Si la cola de eventos está vacía pygame.NOEVENT, se devolverá un evento de tipo inmediatamente. El evento devuelto se elimina de la cola.

PYGAME.EVENT.WAIT()
Devuelve un solo evento de la cola. Si la cola está vacía, esta función esperará hasta que se cree una. El evento se elimina de la cola una vez que se ha devuelto. Mientras el programa está esperando, dormirá en un estado inactivo. Esto es importante para los programas que desean compartir el sistema con otras aplicaciones.

PYGAME.EVENT.PEEK()
Devuelve True si hay algún evento del tipo dado esperando en la cola. Si se pasa una secuencia de tipos de eventos, esto regresará Truesi alguno de esos eventos está en la cola.

PYGAME.EVENT.CLEAR()
Elimina todos los eventos de la cola.

DISPLAY

Este módulo ofrece control sobre la pantalla de pygame. Pygame tiene una superficie de visualización única que está contenida en una ventana o se ejecuta a pantalla completa. El origen de la pantalla, donde x = 0 e y = 0, es la esquina superior izquierda de la pantalla. Ambos ejes aumentan positivamente hacia la esquina inferior derecha de la pantalla.

PYGAME.DISPLAY.INIT()
Inicializa el módulo de visualización de pygame. El módulo de visualización no puede hacer nada hasta que se inicialice. Esto generalmente se maneja automáticamente cuando llama al nivel superior pygame.init().

PYGAME.DISPLAY.QUIT( )
Esto cerrará todo el módulo de pantalla. Esto significa que cualquier pantalla activa se cerrará. Esto también se manejará automáticamente cuando el programa salga.

PYGAME.DISPLAY.GET_INIT()
Devuelve True si el pygame está actualmente inicializado.

PYGAME.DISPLAY.SET_MODE()
Esta función creará una superficie de visualización. Los argumentos pasados son solicitudes de un tipo de visualización. La pantalla creada real será la mejor coincidencia posible admitida por el sistema.

PYGAME.DISPLAY.GET_SURFACE()
Devuelve una referencia a la superficie de visualización configurada actualmente. Si no se ha configurado el modo de visualización, esto devolverá “Ninguno”.

PYGAME.DISPLAY.FLIP()
Actualice la pantalla completa Surface a la pantalla.

PYGAME.DISPLAY.UPDATE()
Esta función es como una versión optimizada de pygame.display.flip()para pantallas de software. Solo permite actualizar una parte de la pantalla, en lugar de toda el área. Si no se pasa ningún argumento, actualiza toda la superficie como pygame.display.flip().

PYGAME.DISPLAY.SET_ICON( )
Cambiar la imagen del sistema para la ventana de visualización. Establece el icono de tiempo de ejecución que el sistema usará para representar la ventana de visualización. Todas las ventanas tienen un logotipo de pygame simple para el icono de la ventana.
Puede pasar cualquier superficie, pero la mayoría de los sistemas quieren una imagen más pequeña alrededor de 32x32. La imagen puede tener una transparencia de colorkey que se pasará al sistema.
Algunos sistemas no permiten que el icono de la ventana cambie después de que se haya mostrado. Se puede llamar pygame.display.set_mode()a esta función antes para crear el icono antes de configurar el modo de visualización.

PYGAME.DISPLAY.SET_CAPTION( )
Establecer el título de la ventana actual
Si la pantalla tiene un título de ventana, esta función cambiará el nombre en la ventana. Algunos sistemas admiten un título alternativo más corto que se utilizará para pantallas minimizadas.

PYGAME.DISPLAY.GET_NUM_DISPLAYS( )
Devuelve el número de pantallas de pygame
Devuelve el número de pantallas disponibles. Esto siempre es 1 si pygame.get_sdl_version() obtener el número de versión de SDL devuelve un número de versión principal por debajo de 2.

PYGAME.DISPLAY.GET_WINDOW_SIZE()
Devuelve el tamaño de la ventana o pantalla
Devuelve el tamaño de la ventana inicializada con pygame.set_mode(). Esto puede diferir del tamaño de la superficie de la pantalla si SCALED se usa.
Nuevo en pygame 2.0.

IMAGE

El módulo de imagen contiene funciones para cargar y guardar imágenes, así como para transferir Superficies a formatos utilizables por otros paquetes.
Tenga en cuenta que no hay clase de imagen; una imagen se carga como un objeto Surface. La clase Surface permite la manipulación (dibujar líneas, establecer píxeles, capturar regiones, etc.).

PYGAME.IMAGE.LOAD()

Cargue una imagen de una fuente de archivo. Puede pasar un nombre de archivo o un objeto similar a un archivo Python.

PYGAME.IMAGE.SAVE()
Guardar una imagen en el disco. Pygame 1.8: guardar archivos PNG y JPEG.

KEY

Este módulo contiene funciones para tratar con el teclado.

El pygame.event es un módulo de pygame para interactuar con eventos y colas obtiene pygame.KEYDOWN y pygame.KEYUP eventos cuando se presionan y sueltan los botones del teclado. Ambos eventos tienen key y mod atributos.

key: una ID entera que representa cada tecla del teclado
mod: una máscara de bits de todas las teclas modificadoras que estaban presionadas cuando ocurrió el evento

PYGAME.KEY.GET_PRESSED ()
Obtener el estado de todos los botones del teclado

```
for e in pygame.event.get():
    if e.type == pygame.QUIT:break
    if e.type == pygame.KEYDOWN:
        if e.key == pygame.K_p: state = PAUSE
        if e.key == pygame.K_s: state = RUNNING
        if e.key == pygame.K_ESCAPE:
            pygame.quit()
```

SURFACE

Una superficie de pygame se usa para representar cualquier imagen. La superficie tiene una resolución fija y formato de píxel. Las superficies con píxeles de 8 bits utilizan una paleta de colores para asignar a color de 24 bits.

PYGAME. SURFACE. BLIT ()
Dibujar una imagen sobre otra

PYGAME. SURFACE. BLITS ()
Dibujar muchas imágenes en otra

PYGAME. SURFACE. CONVERT ()
Crea una nueva copia de Surface con el formato de píxel cambiado.

PYGAME. SURFACE. COPY ()
Hace una copia duplicada de una superficie. La nueva superficie tendrá los mismos formatos de píxeles, paletas de colores, configuraciones de transparencia y clase que la original.

PYGAME. SURFACE. FILL ()
Rellene la superficie con un color sólido. Si no se da un argumento correcto, se rellenará toda la superficie.

PYGAME. SURFACE. SCROLL ()
Desplaza la imagen de la superficie en su lugar. desplazamiento (dx = 0, dy = 0) -> Ninguno

Mueva la imagen dx píxeles hacia la derecha y dy píxeles hacia abajo. dx y dy pueden ser negativos para los desplazamientos hacia la izquierda y hacia arriba, respectivamente. Las áreas de la superficie que no se sobrescriben conservan sus valores de píxeles originales. El desplazamiento está contenido en el área de recorte de superficie. Es seguro tener valores dx y dy que excedan el tamaño de la superficie.

FONT

El módulo de fuente permite representar fuentes TrueType en un nuevo objeto Surface. Puede cargar fuentes desde el sistema utilizando la pygame.font.SysFont() función Hay algunas otras funciones para ayudar a buscar las fuentes del sistema.

PYGAME. FONT.SYSFONT()
Devuelve un nuevo objeto Font que se carga desde las fuentes del sistema. La fuente coincidirá con los indicadores solicitados en negrita y cursiva. Si no se encuentra una fuente de sistema adecuada, se volverá a cargar la fuente de pygame predeterminada. El nombre de la fuente puede ser una lista separada por comas de nombres de fuentes para buscar.

PYGAME.FONT.GET_FONTS ()
Devuelve una lista de todas las fuentes disponibles en el sistema. Los nombres de las fuentes se establecerán en minúsculas con todos los espacios y signos de puntuación eliminados. Esto funciona en la mayoría de los sistemas, pero algunos devolverán una lista vacía si no pueden encontrar las fuentes.

PYGAME.FONT.MATCH_FONT ()
Devuelve la ruta completa a un archivo de fuente en el sistema. Si negrita o cursiva se establecen en verdadero, esto intentará encontrar la familia correcta de fuentes.

PYGAME. FONT.FONT.SIZE ()
Devuelve las dimensiones necesarias para representar el texto. Esto se puede usar para ayudar a determinar el posicionamiento necesario para el texto antes de que se procese. También se puede usar para envolver palabras y otros efectos de diseño.

PYGAME. FONT.FONT.RENDER ()
Rellene la superficie con un color sólido. Si no se da un argumento correcto, se rellenará toda la superficie.

EJEMPLOS

```
import pygame

pygame.init()
screen = pygame.display.set_mode((400, 300))
done = False

while not done:
    for event in pygame.event.get():
        if event.type == pygame.quit():
            done = True

    pygame.display.flip()
```

import pygame - necesario para acceder al marco de PyGame.
pygame.init() - Inicializa todos los módulos necesarios para PyGame.
pygame.display.set_mode((width, height)) - Esto abrirá una ventana del tamaño deseado. El valor de retorno es

```
import pygame

pygame.init()
screen = pygame.display.set_mode((400, 300))
done = False
is_blue = True
x = 30
y = 30

clock = pygame.time.Clock()

while not done:
        for event in pygame.event.get():
                if event.type == pygame.QUIT:
                        done = True
                if event.type == pygame.KEYDOWN
and event.key == pygame.K_SPACE:
                        is_blue = not is_blue

        pressed = pygame.key.get_pressed()
        if pressed[pygame.K_UP]: y -= 3
        if pressed[pygame.K_DOWN]: y += 3
        if pressed[pygame.K_LEFT]: x -= 3
        if pressed[pygame.K_RIGHT]: x += 3

        screen.fill((0, 0, 0))
        if is_blue: color = (0, 128, 255)
        else: color = (255, 100, 0)
        pygame.draw.rect(screen, color, pygame.R
ect(x, y, 60, 60))

        pygame.display.flip()
        clock.tick(60)
```

un objeto Surface, que es el objeto sobre el que realizará operaciones gráficas.
pygame.event.get() - Esto vacía la cola del evento. Si no llama a esto, los mensajes de Windows comenzarán a acumularse y su juego dejará de responder en opinión del sistema operativo.
pygame.quit() - Este es el tipo de evento que se activa cuando cerrar la ventana.
pygame.display.flip() - Las actualizaciones en la pantalla sean visibles.

REFERENCIAS

[1] https://www.pygame.org/docs/