

Funciones predefinidas de EasyCanvas

Función de creación de la ventana gráfica

- `easycanvas_configure(size=(600,400), coordinates=(0,0,1000,1000), title='EasyCanvas', background = 'white')`.
 - `size=(width,height)`: Tamaño físico en píxeles de la ventana gráfica.
 - `coordinates=(xinf, yinf, xsup, ysup)`: Tamaño lógico de la ventana gráfica.
 - `title`. Título de la ventana gráfica.
 - `background`: color de fondo de la ventana gráfica.

Funciones de dibujo

- `create_point(x, y, color)`. Dibuja el punto y devuelve un índice para poder borrarlo. Si no se indica color, se toma 'black' por defecto.
- `create_line(x1, y1, x2, y2, color)`. Dibuja la línea y devuelve un índice para poder borrarla. Si no se indica color, se toma 'black' por defecto.
- `create_circle(x, y, radio, color)`. Dibuja el círculo y devuelve un índice para poder borrarlo. Si no se indica color, se toma 'black' por defecto.
- `create_filled_circle(x, y, radio, colorExterior, colorRelleno)`. Ídem al anterior pero relleno. Si sólo se indica un color, se tomará como color exterior y de relleno.
- `create_rectangle(x1, y1, x2, y2, color)`. Dibuja el rectángulo y devuelve un índice para poder borrarlo. Si no se indica color, se toma 'black' por defecto.
- `create_filled_rectangle(x1, y1, x2, y2, colorExterior, colorRelleno)`. Ídem al anterior pero relleno. Si sólo se indica un color, se tomará como color exterior y de relleno.
- `create_text(x, y, cadena, tamaño, ancla, colortexto)`. Escribe el texto cadena en la ventana gráfica. El parámetro ancla puede tomar los siguientes valores: 'CENTER', 'N', 'S', 'E', 'W', 'NE', 'SE', 'NW' y 'SW'. Si no se pone tamaño, toma el valor 10, del mismo modo ancla toma el valor 'CENTER' y colortexto 'black'.

Funciones de borrado

- `erase(id)`. Borra un objeto dado su id. Todas las funciones de dibujo devuelven un id (un entero).
- `erase()`. Borra todos los objetos de la ventana gráfica.

Funciones entrada(leer teclado) / salida (guardar imagen)

- `readkey(blocking=True)`. Lee una tecla sin hacer echo por pantalla. Si 'blocking' es True y no hay ninguna tecla pulsada se espera a que se pulse (si 'blocking' fuera False devolvería directamente None).
- `save_EPS(nombreFichero)`. Guarda el dibujo actual de la ventana gráfica en el fichero 'nombreFichero' con el formato PostScript Encapsulado (EPS).

Funciones avanzadas

- `erase(etiqueta)`. Borra todos los objetos con dicha etiqueta. Las funciones de dibujo (`create_point`, `create_line`, `create_circle`, `create_filled_circle`, `create_rectangle`, `create_filled_rectangle` y `create_text`) tienen un parámetro adicional para añadir una etiqueta (tag) a los objetos. Por ejemplo:

```
create_point(10,10,'red',tags='etiqueta1')
```

```
create_point(20,30,'blue',tags='etiqueta1')
```

```
erase('etiqueta1') # borra los dos puntos
```

- `move(indice,xinc,yinc)`. Desplaza una distancia relativa el objeto con dicho índice.
- `move(etiqueta,xinc,yinc)`. Desplaza una distancia relativa (a cada objeto) todos los objetos con dicha etiqueta.
- `mouse_state()`. Devuelve la tupla (botón, posx, posy) donde botón puede ser 0 (si no hay ningún botón pulsado) o un entero del 1 al 3 que identifica qué botón se encuentra actualmente pulsado (1:izq. 2:central 3:der.). Los otros dos elementos de la tupla (posx y posy) son las coordenadas del ratón en la ventana gráfica. Si el ratón se encuentra fuera de la ventana gráfica esta función devuelve (None, None, None). IMPORTANTE: debes tener cuidado al utilizar esta función, pues cada vez que se pulsa un botón, éste se encuentra pulsado durante varios milisegundos, suficiente para que sucesivas llamadas a esta función devuelvan los mismos valores. Por esto, necesitas asegurarte de que el botón se ha soltado antes de volver a mirar si se ha pulsado de nuevo.

Ejemplo

```
from easycanvas import EasyCanvas
```

```
class Demo1(EasyCanvas):
    def main(self):
        self.easycanvas_configure(title = 'Demo 1 - Funciones predefinidas',
                                   background = 'white',
                                   size = (600,600),
                                   coordinates = (0,0, 1000, 1000))

        l=[]
        #Dibuja matriz de puntos
        for x in range(50,450,21):
            for y in range(550,950,21):
                l.append(self.create_point(x,y,['black','red'][(x+y)%2]))

        #Dibuja dos círculos (uno relleno y otro no)
        l.append(self.create_filled_circle(750,750,250,'black','blue'))
        l.append(self.create_circle(250,250,250,'red'))

        #Dibuja dos rectángulos (uno relleno y otro no)
        l.append(self.create_rectangle(300,150,400,350,'red'))
        l.append(self.create_filled_rectangle(100,150,200,350,'black','red'))

        #Dibuja dos líneas en cruz
        l.append(self.create_line(500,250,1000,250,'green'))
        l.append(self.create_line(750,0,750,500,'black'))

        #escribe texto
        l.append(self.create_text(500,500,"Press any key to delete all",12))

        #Borra todos los objetos al pulsar Return
        self.readkey(True)
        for indice in l:
            self.erase(indice)

        self.create_text(500,500,"Press any key to exit",12)
        self.readkey(True)
```

```
Demo1().run()
```