

Pygame : les transformations géométriques

Présentation

Le module **transform** de Pygame permet de créer de **nouvelles surfaces Pygame** en appliquant à une surface source une ou plusieurs transformations, isométriques ou non.

La fonction flip

```
pygame.transform.flip(Surface, xbool, ybool)
```

La fonction **flip** permet de faire une **symétrie axiale, verticale** ou **horizontale** ou les **deux**.

Elle prend en **paramètre** :

- la **surface** que l'on va transformer,
- un **booléen** indiquant que l'on procède à une **symétrie selon l'axe vertical** en indiquant **True**,
- un booléen indiquant que l'on procède à une **symétrie selon l'axe horizontal** en indiquant **True**.

Si les deux booléens sont à True, alors on procède aux deux symétries.

Exemple :

```
IMG_FRAISE = pygame.image.load("fraise.png")
IMG_FRAISE = pygame.transform.flip(IMG_FRAISE, True, True)
```

La fonction rotate

```
rotate(Surface, angle)
```

La fonction **rotate** permet de faire faire une rotation à la surface. L'angle passé en paramètre est exprimé selon le sens antihoraire, en degrés, et non en radians comme c'est parfois le cas pour d'autres fonctions Pygame (le tracé d'arcs d'ellipse notamment).

Le programme complet

```
import pygame
pygame.init()
HAUTEUR = 400
LARGEUR = 400
COULEUR_FOND = (255, 255, 255)
ECRAN = pygame.display.set_mode((LARGEUR, HAUTEUR))
ARRET = False

# image Fraise
x_fraise = 50
y_fraise = 50
largeur_fraise = 40
hauteur_fraise = 40
IMG_FRAISE = pygame.image.load("fraise.png")

# image Ballon
x_ballon = 10
y_ballon = 10
largeur_ballon = 20
hauteur_ballon = 20
IMG_BALLOON = pygame.image.load("ballon.png")
ballonSpeed = [1, 0]
#IMG_BALLOON.move(ballonSpeed)
ballonRect = IMG_BALLOON.get_rect()
while not ARRET:
    pygame.time.delay(100)
    ECRAN.fill(COULEUR_FOND)
```

```
for event in pygame.event.get():
    if event.type == pygame.QUIT:
        sys.exit()
    elif event.type == pygame.KEYDOWN:
        if event.key == pygame.K_RIGHT:
            x_fraise = x_fraise + 4
        elif event.key == pygame.K_LEFT:
            x_fraise = x_fraise - 4

    ECRAN.blit(IMG_FRAISE, (x_fraise, y_fraise))

    ECRAN.blit(IMG_BALLOON, ballonRect)
    ballonRect = ballonRect.move(ballonSpeed)
    if ballonRect.right > LARGEUR:
        ballonSpeed[0] = - ballonSpeed[0]

pygame.display.update()
```

Les activités ...

[Je reviens à la liste des activités.](#)

<https://repl.it/@charlestecher/Gerer-une-image> <html>

</html>

Les activités ...

[Je reviens à la liste des activités.](#)

From:

/ - **Les cours du BTS SIO**

Permanent link:

</doku.php/isn/pygame03?rev=1573724646>

Last update: **2019/11/14 10:44**

