

# Documentation Pygame

ISN – Décembre 2022

## 1 Code minimal

```
1 # Importation de pygame
2 import pygame
3
4 pygame.init()
5
6 # Initialisation de la fenetre
7 largeur = 600
8 hauteur = 400
9 windowSurface = pygame.display.set_mode((largeur, hauteur), 0,32)
10
11 # Initialisation des parametres
12
13
14 # Boucle de jeu
15 clock = pygame.time.Clock()
16 running = True
17 while running:
18     # Limitation du nombre de tours de boucle par seconde.
19     clock.tick(10)
20     # Boucle des evenements
21     for event in pygame.event.get():
22         if event.type == pygame.QUIT:
23             running = False
24
25     # Elements a tracer
26
27     pygame.display.update()
28
29
30 pygame.quit()
```

## 2 Dessiner sur la fenêtre : Draw

Dans toute la suite, on supposera que vous avez appelé la fenêtre `windowSurface`.

Les couleurs se définissent avec leur code RGB :

```
1 BLACK = (0, 0, 0)
2 WHITE = (255, 255, 255)
3 RED = (255, 0, 0)
4 GREEN = (0, 255, 0)
5 BLUE = (0, 0, 255)
```

Tracer de objets géométriques

- **Un segment :** (60, 60) sont les coordonnées du points de départ, (120,60) le point d'arrivé et 4 est l'épaisseur du trait.

```
1 pygame.draw.line(windowSurface, color, (60, 60), (120, 60), 4)
2
```

- **Un cercle :** (300, 50) sont les coordonnées du centre, 50 le rayon et 0 l'épaisseur du trait (0 signifie que le cercle est entièrement colorié).

```
1 pygame.draw.circle(windowSurface, color, (300, 50), 20, 0)
2
```

- **Une ellipse** : 300 et 250 sont les coordonnées du centre, 40 le rayon horizontal, 80 le rayon vertical et 1 l'épaisseur du trait.

```
1 pygame.draw.ellipse(windowSurface, color, (300, 250, 40,80), 1)
2
```

- **Un rectangle** : 20 et 30 sont les coordonnées du coin en haut à gauche du rectangle, 40 est la largeur et 50 est la hauteur.

```
1 pygame.draw.rect(windowSurface, color, (20, 30, 40, 50))
2
```

- **Un polygone** : ((146, 0), (291, 106), (236, 277), (56, 277), (0, 106)) sont les coordonnées des sommets du polygone.

```
1 pygame.draw.polygon(windowSurface, color,
2     ((146, 0), (291, 106), (236, 277), (56, 277), (0, 106))
3     )
4
```

Il ne faut pas oublier la ligne suivante après avoir tracé tout ce que vous vouliez, sinon rien ne s'affichera.

```
1 pygame.display.update()
```

D'autres fonctions de dessins existent. Voici un exemple de tout ce qui peut être fait en dessin avec pygame.

```
1 # Import a library of functions called 'pygame'
2 import pygame
3 from math import pi
4
5 # Initialize the game engine
6 pygame.init()
7
8 # Define the colors we will use in RGB format
9 BLACK = ( 0, 0, 0)
10 WHITE = (255, 255, 255)
11 BLUE = ( 0, 0, 255)
12 GREEN = ( 0, 255, 0)
13 RED = (255, 0, 0)
14
15 # Set the height and width of the screen
16 size = [400, 300]
17 screen = pygame.display.set_mode(size)
18
19 pygame.display.set_caption("Example code for the draw module")
20
21 #Loop until the user clicks the close button.
22 done = False
23 clock = pygame.time.Clock()
24
25 while not done:
26
27     # This limits the while loop to a max of 10 times per second.
28     # Leave this out and we will use all CPU we can.
29     clock.tick(10)
30
31     for event in pygame.event.get(): # User did something
32         if event.type == pygame.QUIT: # If user clicked close
33             done=True # Flag that we are done so we exit this loop
34
35     # All drawing code happens after the for loop and but
36     # inside the main while done==False loop.
37
38     # Clear the screen and set the screen background
```

```

39 screen.fill(WHITE)
40
41 # Draw on the screen a GREEN line from (0,0) to (50,75)
42 # 5 pixels wide.
43 pygame.draw.line(screen, GREEN, [0, 0], [50,30], 5)
44
45 # Draw on the screen a GREEN line from (0,0) to (50,75)
46 # 5 pixels wide.
47 pygame.draw.lines(screen, BLACK, False, [[0, 80], [50, 90], [200, 80], [220, 30]], 5)
48
49 # Draw on the screen a GREEN line from (0,0) to (50,75)
50 # 5 pixels wide.
51 pygame.draw.aaline(screen, GREEN, [0, 50],[50, 80], True)
52
53 # Draw a rectangle outline
54 pygame.draw.rect(screen, BLACK, [75, 10, 50, 20], 2)
55
56 # Draw a solid rectangle
57 pygame.draw.rect(screen, BLACK, [150, 10, 50, 20])
58
59 # Draw an ellipse outline, using a rectangle as the outside boundaries
60 pygame.draw.ellipse(screen, RED, [225, 10, 50, 20], 2)
61
62 # Draw an solid ellipse, using a rectangle as the outside boundaries
63 pygame.draw.ellipse(screen, RED, [300, 10, 50, 20])
64
65 # This draws a triangle using the polygon command
66 pygame.draw.polygon(screen, BLACK, [[100, 100], [0, 200], [200, 200]], 5)
67
68 # Draw an arc as part of an ellipse.
69 # Use radians to determine what angle to draw.
70 pygame.draw.arc(screen, BLACK,[210, 75, 150, 125], 0, pi/2, 2)
71 pygame.draw.arc(screen, GREEN,[210, 75, 150, 125], pi/2, pi, 2)
72 pygame.draw.arc(screen, BLUE, [210, 75, 150, 125], pi,3*pi/2, 2)
73 pygame.draw.arc(screen, RED, [210, 75, 150, 125], 3*pi/2, 2*pi, 2)
74
75 # Draw a circle
76 pygame.draw.circle(screen, BLUE, [60, 250], 40)
77
78 # Go ahead and update the screen with what we've drawn.
79 # This MUST happen after all the other drawing commands.
80 pygame.display.update()
81
82 # Be IDLE friendly
83 pygame.quit()

```

**Ajouter une image :** Pygame permet d'ajouter des images ayant les formats suivant : JPG, PNG, GIF (non-animated),BMP. On supposera dans la suite qu'elles sont rangées dans le même dossier que notre programme.

```

1 #Charger l'image
2 img = pygame.image.load('image.jpg')
3 # L'afficher sur la surface
4 windowSurface.blit(img, (0,0))

```

Les coordonnées (0,0) sont les coordonnées de l'angle en haut à droit de l'image sur la surface.

### 3 Interaction avec les périphériques : Events

L'interaction avec l'utilisateur se fait dans la boucle des évènements.

```

1 for event in pygame.event.get():
2     if event.type == pygame.QUIT:

```

```

3     running = False
4     elif event.type == pygame.KEYUP:
5         # choses a faire quand une touche du clavier est relachee
6         if event.key == pygame.K_UP:
7             # choses a faire quand c'est la touche fleche du haut
8         elif event.key == pygame.K_DOWN:
9             # choses a faire quand c'est la touche fleche du bas
10        elif event.type == pygame.KEYDOWN:
11            # choses a faire quand une touche du clavier est pressee
12            elif event.key == pygame.K_LEFT:
13                # choses a faire quand c'est la touche fleche de gauche
14            elif event.key == pygame.K_RIGHT:
15                # choses a faire quand c'est la touche fleche de droite
16        elif event.type == pygame.MOUSEBUTTONUP:
17            # choses a faire quand le bouton de la souris est relache
18        elif event.type == pygame.MOUSEBUTTONDOWN:
19            # choses a faire quand le bouton de la souris est pressee

```

De manière générale, le nom des touches de clavier sont faite sur le même modèle : K\_### où on remplace les # par le nom de la touche.

- Touche flèche du haut : K\_UP
- Touche E : K\_E
- Touche entrée : K\_ESCAPE

Quelques méthodes pratiques pour manipuler la souris

- `pygame.mouse.get_pos()` : connaître la position de la souris sur la fenêtre.
- `pygame.mouse.set_pos(x,y)` : déplacer la souris à un endroit.