Documentation Pygame

1 Code minimal

```
# Importation de pygame
import pygame

pygame.init()

# Initialisation de la fenetre
largeur = 600
hauteur = 400
windowSurface = pygame.display.set_mode((largeur, hauteur), 0, 32)

# Initialisation des parametres


# Boucle de jeu
clock = pygame.time.Clock()
running = True
while running:
    # Limitation du nombre de tours de boucle par seconde.
    clock.tick(10)
    # Boucle des evenements
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

    # Elements a tracer

    pygame.display.update()


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 :

```
BLACK = (0, 0, 0)
WHITE = (255, 255, 255)
RED = (255, 0, 0)
GREEN = (0, 255, 0)
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.

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

- 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).

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

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

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

- 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.

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

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

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

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

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

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

```
# Import a library of functions called 'pygame'
import pygame
from math import pi

# Initialize the game engine
pygame.init()

# Define the colors we will use in RGB format
BLACK = (  0,   0,   0)
WHITE = (255, 255, 255)
BLUE =  (  0,   0, 255)
GREEN = (  0, 255,   0)
RED =   (255,   0,   0)


# Set the height and width of the screen
size = [400, 300]
screen = pygame.display.set_mode(size)

pygame.display.set_caption("Examplezcodez for z the zdrawzmodule")

#Loop until the user clicks the close button.
done = False
clock = pygame.time.Clock()

while not done:

    # This limits the while loop to a max of 10 times per second.
    # Leave this out and we will use all CPU we can.
    clock.tick(10)

    for event in pygame.event.get(): # User did something
        if event.type == pygame.QUIT: # If user clicked close
            done=True # Flag that we are done so we exit this loop

    # All drawing code happens after the for loop and but
    # inside the main while done==False loop.

    # Clear the screen and set the screen background
```

```
    screen.fill(WHITE)

    # Draw on the screen a GREEN line from (0,0) to (50.75)
    # 5 pixels wide.
    pygame.draw.line(screen, GREEN, [0, 0], [50,30], 5)

    # Draw on the screen a GREEN line from (0,0) to (50.75)
    # 5 pixels wide.
    pygame.draw.lines(screen, BLACK, False, [[0, 80], [50, 90], [200, 80], [220,
        30]], 5)

    # Draw on the screen a GREEN line from (0,0) to (50.75)
    # 5 pixels wide.
    pygame.draw.aaline(screen, GREEN, [0, 50],[50, 80], True)

    # Draw a rectangle outline
    pygame.draw.rect(screen, BLACK, [75, 10, 50, 20], 2)

    # Draw a solid rectangle
    pygame.draw.rect(screen, BLACK, [150, 10, 50, 20])

    # Draw an ellipse outline, using a rectangle as the outside boundaries
    pygame.draw.ellipse(screen, RED, [225, 10, 50, 20], 2)

    # Draw an solid ellipse, using a rectangle as the outside boundaries
    pygame.draw.ellipse(screen, RED, [300, 10, 50, 20])

    # This draws a triangle using the polygon command
    pygame.draw.polygon(screen, BLACK, [[100, 100], [0, 200], [200, 200]], 5)

    # Draw an arc as part of an ellipse.
    # Use radians to determine what angle to draw.
    pygame.draw.arc(screen, BLACK,[210, 75, 150, 125], 0, pi/2, 2)
    pygame.draw.arc(screen, GREEN,[210, 75, 150, 125], pi/2, pi, 2)
    pygame.draw.arc(screen, BLUE, [210, 75, 150, 125], pi,3→pi/2, 2)
    pygame.draw.arc(screen, RED,  [210, 75, 150, 125], 3→pi/2, 2→pi, 2)

    # Draw a circle
    pygame.draw.circle(screen, BLUE, [60, 250], 40)

    # Go ahead and update the screen with what we've drawn.
    # This MUST happen after all the other drawing commands.
    pygame.display.update()

# Be IDLE friendly
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.

```
#Charger l'image
img = pygame.image.load('image.jpg')
# L'afficher sur la surface
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.

```
for event in pygame.event.get():
    if event.type == pygame.QUIT:
        running = False
    elif event.type == pygame.KEYUP:
        # choses a faire quand une touche du clavier est relachee
        if event.key == pygame.K_UP:
            # choses a faire quand c'est la touche fleche du haut
        elif event.key == pygame.K_DOWN:
            # choses a faire quand c'est la touche fleche du bas
    elif event.type == pygame.KEYDOWN:
        # choses a faire quand une touche du clavier est pressee
        elif event.key == pygame.K_LEFT:
            # choses a faire quand c'est la touche fleche de gauche
        elif event.key == pygame.K_RIGHT:
            # choses a faire quand c'est la touche fleche de droite
    elif event.type == pygame.MOUSEBUTTONUP:
        # choses a faire quand le bouton de la souris est relache
    elif event.type == pygame.MOUSEBUTTONDOWN:
        # 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.