Le site de référence sur la bibiothèque graphique pygame.

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

Démarrer avec pygame

Importation des modules pygame.On utilise:

import pygame
from pygame.locals import *.

Ensuite on initialise les modules contenus dans la bibliothèque pygame:

pygame.init()

En fin de programme , on peut faire appel à la fontion **quit** pour tout “nettoyer” même si ceci n’est pas nécessaire (python s’en occupe automatiquement):

pygame.quit()

Initialisation d'une fenêtre pour l'affichage:

Une fenêtre pour l'affichage
On définit la largeur et la hauteur de la fenêtre.

Largeur_fenetre=700
hauteur_fenetre=400
On définit la fenêtre avec ces dimensions:

screen=pygame.displayset_mode([Largeur_fenetre,hauteur_fenetre])
La fenêtre est munie d'un système de coordonnées :

L'origine du repère est dans le coin haut-gauche.
x est compris entre 0 et 700.
y est compris entre 0 et 400.

Les surfaces

La surface est un objet de base du module pygame.

Voici un extrait du site [*]:
"The most important part of pygame is the surface. Just think of a surface as a blank piece of paper. You can do a lot of things with a surface you can draw lines on it,fill parts of it with color, copy images to and from it, and set or read individual pixel colors on it. A surface can be any size (within reason)and you can have as many of them as you like (again, within reason). One surface is special the one you create with pygame.display.set_mode().
This 'display surface' represents the screen; whatever you do to it will appear on the user's screen."
Pour créer une surface:

rectangle=pygame.Surface((400,400))

On peut lui donner une couleur de fond:

rectangle.fill(GREEN)

Ensuite , on indique à pygame où afficher ce rectangle:ici aux coordonnées (150;150):

screen.blit(rectangle,(150,250))

Pour que l'affichage ait lieu , il est nécessaire d'utiliser la fonction update:

pygame.display.update()

Dessiner dans une surface

Plusieurs tracés sont possibles à l'intérieur d'une surface.

Tracer une ligne droite:
pygame.draw.line(Surface,couleur,début,fin,width=1)

- **Surface** est la surface à l'intérieur de laquelle on souhaite tracer la ligne.
- **couleur** est la couleur du trait : un triplet RVB (255,0,0) par exemple pour un trait rouge.
- **debut** et **fin** désignent les coordonnées du début et de la fin du trait.
- **width** est l'épaisseur du trait.

Dessiner dans une surface

On peut de même tracer des polygones , des cercles, etc..
pygame.draw.polygon(Surface,couleur,[liste des sommets du polygone],width)
width est l'épaisseur du trait.
Quand width=0 le polygone est plein.

Voir le site [*] pour un descriptif détaillé de ces méthodes de dessin.

un premier programme avec pygame

```
# -*- coding : utf 8 -*-
import pygame
from pygame.locals import *
#On importe également deux modules nécessaires à la gestion du système
import os
import sys
Hauteur , Largeur = 800 ,800

pygame.init()
#On définit la fenêtre 800*800 dans laquelle tout va s'afficher

screen=pygame.display.set_mode((Hauteur,Largeur))
```

un premier programme avec pygame

#Une boucle infinie pour maintenir l'affichage
while True:

e=event.wait()
if e.type==KEYDOWN : break

Dans cette boucle while on peut écrire un traitement

Sans oublier de rafraîchir la fenêtre.
pygame.display.update()
pygame.quit()

Gestion d'évènements

Gestion de la sortie d'une boucle infinie:

```
Continuer=1
while Continuer:
    for event in pygame.event.get():
        if event.type==pygame.KEYDOWN:
            Continuer=0
```

rectangles

Surface.get_rect()
get the rectangular area of the Surface

“Returns a new rectangle covering the entire surface.
This rectangle will always start at 0, 0 with a width and height the same size as the image.”
Exemple:

On crée une surface
rectangle=pygame.Surface((100,100))
puis on récupère un rectangle ayant les mêmes dimensionsque Surface
nouveau_rectangle=rectangle.get_rect()

déplacement de rectangles

On peut déplacer le rectangle en utilisant la fonction **move(x,y** (x,y) sont les coordonnées du déplacement.

nouveau_rectangle = nouveau_rectangle.move(0,10)
fenetre.blit(rectangle, nouveau_rectangle)
pygame.display.flip()

Application: Ecrire un programme qui affiche un rectangle qui se déplace verticalement.

Exerices d'initiation

1.
   1. Ecrire un programme *base_01.py* qui affiche une fenêtre 500×500.
   2. Tracer à l'intérieur de la fenêtre un carré de côté 100.
   3. Tracer les deux diagonales du carré.
2.
   1. Ecrire un programme qui affiche un carré $C1$ de côté 100.
   2. Tracer à l'intérieur du carré un autre carré dont les sommets sont les milieux des côtés de $C1$.
   3. Tracer de la même manière 10 carrés imbriqués.
3.
   1. Afficher une fenêtre de dimensions 800*800.
   2. Afficher une grille 10*10 de la fenêtre à l'aide de lignes horizontales et verticales.