PyGame

- PyGame is the library used to create GUI effect in games
- It supports the picture, sound, shapes
- PyGame provides cross platform with system's multimedia hardware like, keyboard, joystick, sound etc

---

- Difference between turtle libary and PyGame
```
  - The Turtle package is a tool to help to draw Graphical output
  - On the other hand, PyGame is game development package that gives you all
the tools to create a game with Python.
```

Installing PyGame

For windows

`pip install pygame`

For mac (Provided you have python3 installed)

`pip3 install pygame`

---

if the commands don't work its porbably because you have not added python/python3 to the enviornment variable. Please check the steps to resolve this probem attached in the folder

Bare minimum code

Below code creates a simple blank pygame window

In [ ]:

```python
import pygame
import time


pygame.init()

screen = pygame.display.set_mode([500, 500])   # creates a screen with the said size

run = True
while run:
    pass

pygame.quit()
```

```
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
```

Command explaintion

- The `pygame.display.set_mode([500, 500])` is used to create a screen of a specified size in pygame. Creating a screen by default creates a surface. Surface is the drawable portion on the screen.

PROBLEM : The launched window does not work properly!!!

You can also see the not responding sign on top of the window.

To understand why this is happeneing we must understand how an app actually works. Any app can be divided into 2 parts the

1. The main app window
2. The event handler

- The main window is where we get the graphical outputs
- The event handler is a task which is supossed to run continously in the background checking for all the events that are happening in the main window.

Events

- Events are bacially any unser interaction that can be caught on the screen.
- This can include keyboard key presses, mouse interaction etc.

In the above code we created a main window when we created the display. Calling the init() funcion also creates an event handler for us which is runnig in the background.

Then what is the issue???????

- The event handler that is running in the background is capturing events(user interactions) and storing them into a
  list(queue).
- We are supposed to go through this list frequently enough and tell the app what has to be done when a particular event has occured.
- Even if no action has to be performed after a particular event, reading the event list(queue) has to be done.

- In the above we have a game loop but the game loop is not reading the event list. Which causes the app to become not responding.

In [ ]:

```
import pygame
import time


pygame.init()

screen = pygame.display.set_mode([500, 500])   # creates a screen with the said size

run = True
while run:
    for event in pygame.event.get():
        pass

pygame.quit()
```

PROBLEM : We wont be able to close the above window!!

- To be able to close the window we need to check if the close button was clicked and break the game loop(while loop).

In [3]:

```
#TASK :closing output window without error
```

In [4]:

```
import pygame
import time


pygame.init()

screen = pygame.display.set_mode([500, 500])   # creates a screen with the said size

run = True
while run:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            run = False

pygame.quit()
```

Command explaintion

`pygame.event.get()` returns us a list of event which the event handler has caught. We then need to go through the list and check manually if a particular event has occured.

---

list of all events that can be captured by the event handler.

Event	Attributes

Event	Attributes
QUIT	none
ACTIVEEVENT	gain, state
KEYDOWN	key, mod, unicode, scancode
KEYUP	key, mod
MOUSEMOTION	pos, rel, buttons
MOUSEBUTTONUP	pos, button
MOUSEBUTTONDOWN	pos, button
JOYAXISMOTION	joy (deprecated), instance_id, axis, value
JOYBALLMOTION	joy (deprecated), instance_id, ball, rel
JOYHATMOTION	joy (deprecated), instance_id, hat, value
JOYBUTTONUP	joy (deprecated), instance_id, button
JOYBUTTONDOWN	joy (deprecated), instance_id, button
VIDEORESIZE	size, w, h
VIDEOEXPOSE	none
USEREVENT	code

Screen / Surface

`screen = pygame.display.set_mode([500, 500])`

The above line of code creates a screen. The screen that we create is also a surfcae in pygame that can be used to draw things on to.

Surface objects has its own set of methods.

Command explaination:

- The most frequenty used one is the `fill(color)` method which fill the screen with a specified color.

In [2]:

```python
import pygame
import time


pygame.init()

screen = pygame.display.set_mode([500, 500])   # creates a screen with the said size

run = True
while run:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            run = False
    screen.fill((255,255,255))

pygame.quit()
```

PROBLEM: We just filled the screen with white but then why is the screen not white

- The issue is that the screen has been painted white but has not yet been updated.
- To update screen we need to flip the disply. Fliping is the operation by which the screen is updated.

Command explaination:

`pygame.display.flip()`

In [3]:

```
import pygame
import time


pygame.init()

screen = pygame.display.set_mode([500, 500])   # creates a screen with the said size

run = True
while run:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            run = False

    screen.fill((255,255,255))

    pygame.display.flip()


pygame.quit()
```

Drawing shapes

The pygame.draw module has a bunch of function that can used to draw shapes.

Drawaing circles

`pygame.draw.circle(Surface, color , center location , radius)`

Arguments	
Surface	The screen or any drawable surface
color	Tuple representing the RGB vlaues
center loaction	Tuple representing the x and y coordinate of the center point of the rectangle
radius	Scalar vlaues representing the radius of the circle

Drawing Rectangles

`pygame.draw.rect(Surface, color , rect)`

Arguments	
Surface	The screen or any drawable surface
color	Tuple representing the RGB vlaues
center loaction	Tuple representing the x and y coordinate of the center point of the rectangle
rect	Tuple representing the x and y coordinate of top left hand corner of the rectangle along with the width and height of the rectangle(left, top, width, height)

Drawing Circle

In [4]:

```python
import pygame
import time


pygame.init()

screen = pygame.display.set_mode([500, 500])   # creates a screen with the said size

run = True
while run:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            run = False

    screen.fill((255,255,255))

    pygame.draw.circle(screen, (0, 0, 255), (250, 250), 75)

    pygame.display.flip()


pygame.quit()
```

Drawing Rectangle

In [5]:

```python
import pygame
import time


pygame.init()

screen = pygame.display.set_mode([500, 500])   # creates a screen with the said size

run = True
while run:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            run = False

    screen.fill((255,255,255))

    pygame.draw.rect(screen, (0, 0, 255), (100,100,300,300))
    pygame.display.flip()


pygame.quit()
```

Setting the FPS

FPS stands for frames per second. This parameter defines how fast the screen is refreshing. As of now we don't have any control over the speed at which the while loop is running. We can limit the speed by creating a clock and setting the frame rate using the tick method.

We won't see andy deiffernce though as we are not animating anything as of yet. But once we start with the animation we will be able to see the difference.

`clock = pygame.time.Clock()`
The above line of code creates a clock object

`clock.tick(30)`
the above line of code sets the frame rate. A frame rate of 30 means the while loop/ game loop will run 30 times every second.
The `clock.tick(30)` has to be inside the game loop for which the frame rate has to controller.

In [ ]:
```
import pygame
import time


pygame.init()

screen = pygame.display.set_mode([500, 500])   # creates a screen with the said size

clock = pygame.time.Clock()

run = True
while run:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            run = False

    screen.fill((255,255,255))

    pygame.draw.rect(screen, (0, 0, 255), (100,100,300,300))

    pygame.display.flip()
    clock.tick(30)


pygame.quit()
```

REVISION :

- Understandng : pygame , difference between pygame
- Installation of pygame

- Understaning : bare code for game window
- Event handling : quit event
- screen colour filling
- Updating screen color
- drawing shapes

HOMEWORK

1. Draw 2 different shapes together on the screen with different size, color and position

---

```
In [ ]:
```