Getting Started with Pygame

Last Updated : 30 Jun, 2021

Pygame is a free-to-use and open-source set of Python Modules. And as the name suggests, it can be used to build games. You can code the games and then use specific commands to change it into an executable file that you can share with your friends to show them the work you have been doing. It includes computer graphics and sound libraries designed to be used with the Python programming language. PyGame 2.0.1 is the latest version at the time of writing this article.

Setting Up Pygame:

By default, Python doesn't come with PyGame as an in-built Library. So we have to install it using the command prompt. Open a command prompt and type the following command:

```
pip install pygame
```

If you already have PyGame installed, use the following command to check the version:

```
pip show pygame
```

If your Pygame is not updated to the latest version, use the following command:

```
pip install pygame --upgrade
```

If this command shows a ModuleNotFoundError, then it is clear that pygame is not installed.

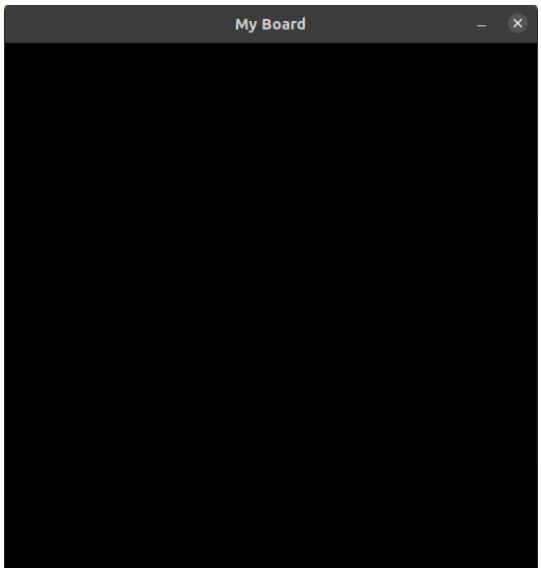
Simple PyGame Example:

Function used:

- **pygame.init():** This command is used to initiate the pygame module.
- **pygame.display.set_mode((500,500)):** This command is used to make a window of desired size, (width, height). The return value is a Surface Object which is the object where we perform different graphical operations.
- **pygame.display.set_caption(title = “”):** This command is used to set the title of the window/ board.
- **pygame.event.get():** This is used to empty the event queue. If we do not call this, the window messages will start to pile up and, the game will become unresponsive in the opinion of the operating system.
- **pygame.QUIT:** This is used to terminate the event when we click on the close button at the corner of the window.

```
import pygame

pygame.init()

# CREATING CANVAS
canvas = pygame.display.set_mode((500, 500))

# TITLE OF CANVAS
pygame.display.set_caption("My Board")
exit = False

while not exit:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            exit = True
    pygame.display.update()
```

Output:



Add Image to the Pygame WIndow:

Blitting is the process of rendering the game object onto the surface. When we create the game object, we should render it. If we don't render the game object, it will show the black window. In pygame there is a way to do this blitting process, i,e, **blit().**

It is one of the slowest operations in any game, so we need to be careful while using it as we shouldn't blit much onto the screen in every frame.

Syntax: *blit(src, dest)*

src : *It is the source of the image which we want to display on the screen*

dest : *It is the coordinates where we want our image to be displayed.*

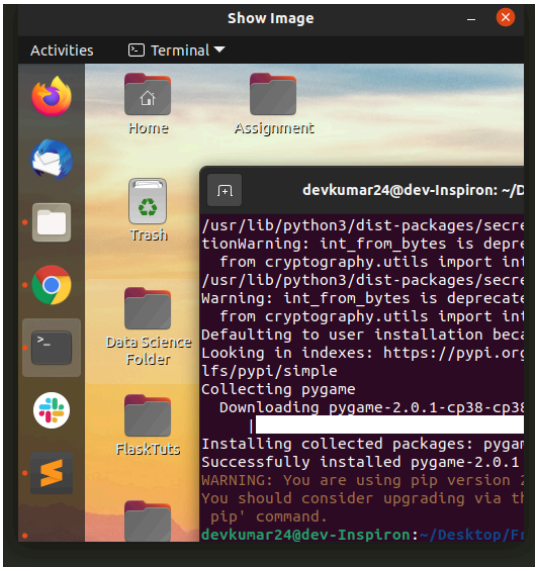
Code:

Python3

```
import pygame

pygame.init()

color = (255,255,255)
position = (0,0)
```

```
# CREATING CANVAS
canvas = pygame.display.set_mode((500,500))

# TITLE OF CANVAS
pygame.display.set_caption("Show Image")

image = pygame.image.load("Screenshot.png")
exit = False

while not exit:
    canvas.fill(color)
    canvas.blit(image, dest = position)

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            exit = True

    pygame.display.update()
```

Output:



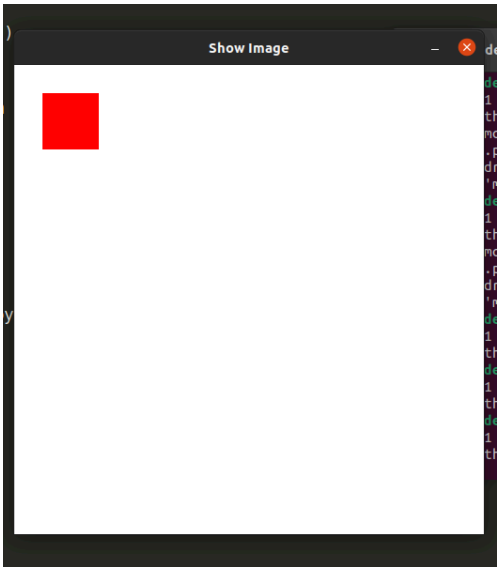
Rectangle Box in Pygame Window:

In Pygame we use **rect()** method to draw rectangle boxes on the window. Pygame uses **Rect** objects to store and manipulate rectangular areas. It can be formed by the combination of left, top, width, and height values.

Python3

```
import pygame

pygame.init()

color = (255,255,255)
rect_color = (255,0,0)

# CREATING CANVAS
canvas = pygame.display.set_mode((500,500))

# TITLE OF CANVAS
pygame.display.set_caption("Show Image")

image = pygame.image.load("Screenshot.png")
exit = False

while not exit:
    canvas.fill(color)
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            exit = True

    pygame.draw.rect(canvas, rect_color,
                     pygame.Rect(30,30,60,60))
    pygame.display.update()
```

Output :



Looking to dive into the world of programming or sharpen your Python skills? Our **Master Python: Complete Beginner to Advanced Course** is your ultimate

guide to becoming proficient in Python. This course covers everything you need to build a solid foundation from fundamental programming concepts to advanced techniques. With **hands-on projects**, real-world examples, and expert guidance, you'll gain the confidence to tackle complex **coding challenges**. Whether you're starting from scratch or aiming to enhance your skills, this course is the perfect fit. Enroll now and master Python, the language of the future!

Previous Article

Introduction to pygame

Next Article

How to Install Pygame on Windows ?

Similar Reads

Python for Game Development: Getting Started with Pygame

For a variety of uses, including web development, data research, automation, and, more and more, game creation, Python has grown to be an immensely popular...

5 min read

Adding Collisions Using pygame.Rect.colliderect in Pygame

Prerequisite: Drawing shapes in Pygame, Introduction to pygame In this article, we are going to use pygame.Rect.colliderect for adding collision in a shape usin...

3 min read

Getting width and height of an image in Pygame

Prerequisites: Pygame To use graphics in python programs we use a module called Pygame. Pygame provides high functionality for developing games and...

3 min read

Python | Getting started with SymPy module

SymPy is a Python library for symbolic mathematics. It aims to become a full-featured computer algebra system (CAS) while keeping the code as simple as...

4 min read

Python | Getting started with psycopg2-PostGreSQL

PostgreSQL is a powerful, open source object-relational database system. PostgreSQL runs on all major operating systems. PostgreSQL follows ACID...

1 min read

Getting started with Python for Automated Trading

Automated Trading is the terminology given to trade entries and exits that are processed and executed via a computer. Automated trading has certain...

3 min read

Getting Started with Python OpenCV

Computer Vision is one of the techniques from which we can understand images and videos and can extract information from them. It is a subset of artificial...

15+ min read

Getting started with Django

Python Django is a web framework that is used to create web applications very efficiently and quickly. Django is called a battery included framework because it...

15+ min read

Getting Started on Heroku with Python

Heroku is a cloud platform as a service supporting several programming languages where a user can deploy, manage and scale their applications. It is...

3 min read

Getting Started with Competitive Programming in Python

Python is a great option for programming in Competitive Programming. First off, its easy-to-understand and concise grammar enables quicker development and...

11 min read

Article Tags : Python Python-PyGame

Practice Tags : python