



Introduction to pygame

Last Updated : 08 Oct, 2020

Game programming is very rewarding nowadays and it can also be used in advertising and as a teaching tool too. Game development includes mathematics, logic, physics, AI, and much more and it can be amazingly fun. In python, game programming is done in pygame and it is one of the best modules for doing so.

Installing pygame:

Pygame requires Python; if you don't already have it, you can download it from python.org. Use python 3.6.1 or greater, because it is much friendlier to newbies, and additionally runs faster.

The best way to install pygame is with the pip tool (which python uses to install packages). Note, this comes with python in recent versions. We use the `--user` flag to tell it to install into the home directory, rather than globally.

```
python3 -m pip install -U pygame --user
```

To see if it works, run one of the included examples:

```
python3 -m pygame.examples.aliens
```

If it works, we are ready to go!

Once you have installed pygame, you're ready to create your very first pygame

instance.

Python3

```
# import the pygame module
import pygame

# import pygame.locals for easier
# access to key coordinates
from pygame.locals import *

# Define our square object and call super to
# give it all the properties and methods of pygame.sprite.Sprite
# Define the class for our square objects
class Square(pygame.sprite.Sprite):
    def __init__(self):
        super(Square, self).__init__()

        # Define the dimension of the surface
        # Here we are making squares of side 25px
        self.surf = pygame.Surface((25, 25))

        # Define the color of the surface using RGB color coding.
        self.surf.fill((0, 200, 255))
        self.rect = self.surf.get_rect()

# initialize pygame
pygame.init()

# Define the dimensions of screen object
screen = pygame.display.set_mode((800, 600))

# instantiate all square objects
square1 = Square()
square2 = Square()
square3 = Square()
square4 = Square()

# Variable to keep our game loop running
gameOn = True

# Our game loop
while gameOn:
    # for loop through the event queue
    for event in pygame.event.get():

        # Check for KEYDOWN event
```

```
if event.type == KEYDOWN:

    # If the Backspace key has been pressed set
    # running to false to exit the main loop
    if event.key == K_BACKSPACE:
        gameOn = False

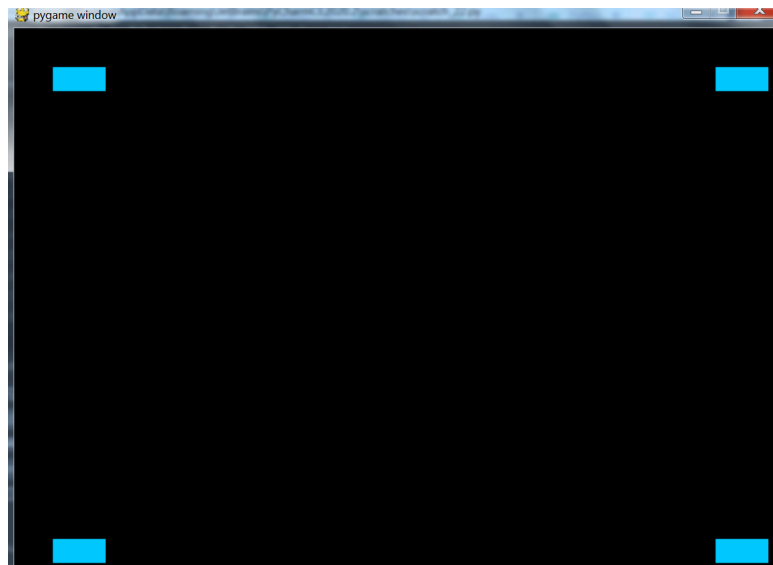
# Check for QUIT event
elif event.type == QUIT:
    gameOn = False

# Define where the squares will appear on the screen
# Use blit to draw them on the screen surface
screen.blit(square1.surf, (40, 40))
screen.blit(square2.surf, (40, 530))
screen.blit(square3.surf, (730, 40))
screen.blit(square4.surf, (730, 530))

# Update the display using flip
pygame.display.flip()
```

The above Python code is a simple pygame script to draw four cyan color squares.

Output:



Don't worry! We will learn a lot more about how to make an object move, how to add animation effects, buttons, music and much more in the next few articles but for now let's understand this basic code which contains all the necessary pygame elements.

```
import pygame

from pygame.locals import *
```

In the first few lines, we import pygame and pygame.locals which is necessary to do before using any module in python.

Sprite, Surf, and Rect:

[Python Basics](#) [Interview Questions](#) [Python Quiz](#) [Popular Packages](#) [Python Projects](#) [Practice Python](#) [AI Wit](#)

by extending the sprite class.

Surf: Surfaces are like blank sheets of paper on which we draw. Our screen object is also a Surface. They can hold images as well.

Rects: Rectangular area that we define on a surface.

Python3

```
class Square(pygame.sprite.Sprite):

    def __init__(self):
        super(Square, self).__init__()
        self.surf = pygame.Surface((25, 25))
        self.surf.fill((0, 200, 255))
        self.rect = self.surf.get_rect()
```

Note: We are using RGB format of color coding in which we can form different colors by giving Red, Blue, and Green color values in the range of 0-255. A tuple of (0, 0, 0) is the color black as there is no color present and a tuple of (255, 255, 255) is the color white.

In our code, we are extending the sprite class so that we can use *surfs* and *rects* to draw our squares. We have made a surface of dimension 25x25px and filled it with a color: (0, 200, 255)

pygame.init() :

Python3

```
pygame.init()

screen = pygame.display.set_mode((800, 600))

square1 = Square()
square2 = Square()
square3 = Square()
square4 = Square()
```

The above lines of code initialize pygame using the command `pygame.init()` which is necessary to use the pygame module commands. After that, we define our screen object and its dimensions in pixels. Then in the next lines, we initialize our four squares.

Game Loop

Python3

```
while running:
    for event in pygame.event.get():
        if event.type == KEYDOWN:
            if event.key == K_BACKSPACE:
                running = False
        elif event.type == QUIT:
            running = False
```

The most important part of the game development code is the game loop. This loop continuously runs in the background until the user ends the game or the game is over. Right now our game loop is not much to look forward to. It only tracks two basic events. We can add more events according to the game state,

user inputs, and much more but the point to remember here is that the game loop must end after some condition else the user will be stuck in the game loop forever.

Blit and Flip –

Blit: Blit keyword is used to draw a surface on another surface. In simple words when we draw a surface we just *blit* it onto some other surface.

Flip: It is used to update the entire screen after everything is drawn. Remember that the flip only works after drawing all the necessary surfaces otherwise, it will update nothing.

Python3

```
screen.blit(square1.surf, (40, 40))
screen.blit(square2.surf, (40, 530))
screen.blit(square3.surf, (730, 40))
screen.blit(square4.surf, (730, 530))

pygame.display.flip()
```

In the above lines of code first, we *blit* four squares on the screen, and then we *flip* to make them appear.

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!

C Chait... [+ Follow](#)



28

Previous Article

[PyGame Tutorial](#)

Next Article

[Getting Started with Pygame](#)

Similar Reads

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

How to create a text input box with Pygame?

In this article, we will discuss how to create a text input box using PyGame. Installation Before initializing pygame library we need to install it. This library c...

3 min read

How to add moving platforms in PyGame

Prerequisite: Drawing in Pygame In this article, we will learn how we can add moving platforms to our game using PyGame in Python. Creating a Platform We...

5 min read

How to get keyboard input in PyGame ?

While using pygame module of Python, we sometimes need to use the keyboard input for various operations such as moving a character in a certain direction. To...

3 min read

Pygame - Surface

When using Pygame, surfaces are generally used to represent the appearance of the object and its position on the screen. All the objects, text, images that we...

6 min read

Python | Display images with PyGame

Pygame is a cross-platform set of Python modules designed for writing video games. It includes computer graphics and sound libraries designed to be used...

2 min read

Python | Display text to PyGame window

Pygame is a cross-platform set of Python modules designed for writing video games. It includes computer graphics and sound libraries designed to be used...

6 min read

Python | Drawing different shapes on PyGame window

Pygame is a cross-platform set of Python modules designed for writing video games. It includes computer graphics and sound libraries designed to be used...

3 min read

PyGame Set Mouse Cursor from Bitmap

In this article, we are going to see how to set the mouse cursor from bitmap using the PyGame module in Python. What is PyGame? It is a cross-platform set of...

5 min read

How to create an empty PyGame window?

Pygame window is a simple window like any other window, in which we display our game screen. It is the first task we do so that we can display our output onto...

2 min read

Article Tags : [Python](#)

Practice Tags : [python](#)



Corporate & Communications Address:-
A-143, 9th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305) | Registered Address:- K 061,
Tower K, Gulshan Vivante Apartment,
Sector 137, Noida, Gautam Buddh
Nagar, Uttar Pradesh, 201305



[Company](#)

[Languages](#)

About Us
Legal
In Media
Contact Us
Advertise with us
GFG Corporate Solution
Placement Training Program
GeeksforGeeks Community

Python
Java
C++
PHP
GoLang
SQL
R Language
Android Tutorial
Tutorials Archive

DSA

Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
Top 100 DSA Interview Problems
DSA Roadmap by Sandeep Jain
All Cheat Sheets

Data Science & ML

Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths
Data Visualisation
Pandas
NumPy
NLP
Deep Learning

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

Python Tutorial

Python Programming Examples
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question
Django

Computer Science

Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths
Software Development
Software Testing

DevOps

Git
Linux
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD

Interview Preparation

Competitive Programming
Top DS or Algo for CP
Company-Wise Recruitment Process
Company-Wise Preparation
Aptitude Preparation
Puzzles

System Design Bootcamp

Interview Questions

School Subjects

Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar
Commerce
World GK

GeeksforGeeks Videos

DSA
Python
Java
C++
Web Development
Data Science
CS Subjects

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved