



[Turtle](#) [Tkinter](#) [Matplotlib](#) [Python Imaging Library](#) [Pyglet](#) [Python](#) [Numpy](#) [Pandas](#) [Python Database](#)

How to add Custom Events in Pygame?

Last Updated : 17 Jun, 2021

In this article, we will see how to add custom events in [PyGame](#).

Installation

PyGame library can be installed using the below command:

```
pip install pygame
```

Although PyGame comes with a set of events (Eg: KEYDOWN and KEYUP), it allows us to create our own additional custom events according to the requirements of our game. Custom events increase the control and flexibility we have over our game. A custom event is the same as creating a User-defined event.

Syntax:

```
<event_name> = pygame.USEREVENT + 1
```

Example:

```
# Here ADDITION and SUBTRACTION is the event name
```

```
ADDITION = pygame.USEREVENT + 1
```

```
SUBTRACTION = pygame.USEREVENT + 2
```

Now, how do we publish our custom events once they are created? This can be done in two ways:

- Using `pygame.event.post()` method.
- Using `pygame.time.set_timer()` method.

Using `pygame.event.post()` method

We can directly post our events using `pygame.event.post()` method. This method adds our event to the end of the events on the queue. In order to execute this, we need to convert our event to Pygame's event type inorder to match the attributes of the post method and avoid errors.

Syntax:

Step 1 – Convert event into event datatype of pygame

```
ADD_event = pygame.event.Event(event)
```

Step 2 – Post the event

```
pygame.event.post(ADD_event) # event_name as parameter
```

Using `pygame.time.set_timer()` method

Broadcasting the event periodically by using PyGame timers. Here, we'll be using another method to publish the event by using `set_timer()` function, which takes two parameters, a user event name and time interval in milliseconds.

Syntax:

event_name, time in ms

```
pygame.time.set_timer(event, duration)
```

Note: In this, we don't need to convert the user-defined event into PyGame event datatype.

Now to create a plot with custom events firstly the attributes for the screen should be set as per requirement. Then create an event and convert it to PyGame event datatype. Now add code for your operations that will generate a custom event.

In the given implementation both of the approaches have been handled.

Program :

Python3

```
# Python program to add Custom Events
import pygame

pygame.init()

# Setting up the screen and timer
screen = pygame.display.set_mode((500, 500))
timer = pygame.time.Clock()

# set title
pygame.display.set_caption('Custom Events')

# defining colours
WHITE = (255, 255, 255)
RED = (255, 0, 0)
GREEN = (0, 255, 0)
BLUE = (0, 0, 255)

# Keep a track of active variable
bg_active_color = WHITE
screen.fill(WHITE)

# custom user event to change color
CHANGE_COLOR = pygame.USEREVENT + 1

# custom user event to inflate default
# box
ON_BOX = pygame.USEREVENT + 2

# creating Rectangle
box = pygame.Rect((225, 225, 50, 50))
grow = True

# posting a event to switch color after
# every 500ms
```

```
pygame.time.set_timer(CHANGE_COLOR, 500)

running = True
while running:

    # checks which all events are posted
    # and based on that perform required
    # operations
    for event in pygame.event.get():

        # switching colours after every
        # 500ms
        if event.type == CHANGE_COLOR:
            if bg_active_color == GREEN:
                screen.fill(GREEN)
                bg_active_color = WHITE
            elif bg_active_color == WHITE:
                screen.fill(WHITE)
                bg_active_color = GREEN

        if event.type == ON_BOX:

            # to inflate and deflate box
            if grow:
                box.inflate_ip(3, 3)
                grow = box.width < 75
            else:
                box.inflate_ip(-3, -3)
                grow = box.width < 50

        if event.type == pygame.QUIT:

            # for quitting the program
            running = False

    # Posting event when the cursor is on top
    # of the box
    if box.collidepoint(pygame.mouse.get_pos()):
        pygame.event.post(pygame.event.Event(ON_BOX))

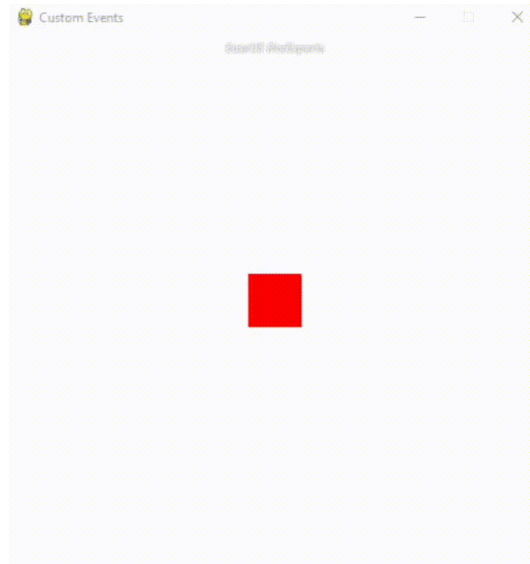
    # Drawing rectangle on the screen
    pygame.draw.rect(screen, RED, box)

    # Updating Screen
    pygame.display.update()

    # Setting Frames per Second
    timer.tick(30)
```

```
pygame.quit()
```

Output :



In the above implementation, we have used the **.post()** method to inflate/deflate the box when the cursor is on the top of the box and **.set_timer()** method to switch background color after every 500ms.

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!

A iama...



5

Previous Article

How to draw a rectangle with rounded corner in PyGame?

Next Article

How to get keyboard input in 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 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 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 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

Introduction to pygame

Game programming is very rewarding nowadays and it can also be used in advertising and as a teaching tool too. Game development includes mathematic...

5 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

Article Tags : [Python](#) [Python-PyGame](#)

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

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

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

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

Computer Science

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

System Design

High Level Design
Low Level Design

Languages

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

Data Science & ML

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

Python Tutorial

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

DevOps

Git
Linux
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

Interview Preparation

Competitive Programming
Top DS or Algo for CP

UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

School Subjects

Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar
Commerce
World GK

Company-Wise Recruitment Process
Company-Wise Preparation
Aptitude Preparation
Puzzles

GeeksforGeeks Videos

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

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved