Snowfall display using Pygame in Python

Last Updated : 11 Sep, 2021

Not everybody must have witnessed Snowfall personally but wait a minute, What if you can see the snowfall right on your screen by just a few lines of creativity and Programming.

Turtle Tkinter Matplotlib Python Imaging Library Pyglet Python Numpy Pandas Python Database

Pygame.

Steps for snowfall creation

1. Importing modules

First, we need to import the Pygame module by using the command.

```
import pygame
```

Also, along with Pygame, we will also need random module. Python has a built-in module that you can use to make random numbers just by importing random module.

```
import random
```

2. Initialize the game engine

It simply means choose the colors you want to use. In programming World, Whatever you can think you can make. At the end of the article, you will find green snowfall on the white background.

Python3

```
# initialize
pygame.init()

# chosen colours will be used
# to display the output
WHITE = [255, 255, 255]
GREEN = [0, 255, 0]
```

3. Specify the size of the screen

It can be a new number depending upon the resolution of your system.

Python3

```
# specify the size

SIZE = [400, 400]
screen = pygame.display.set_mode(SIZE)
```

4. Assign a name to your snowfall window screen

The name given can be seen on the left corner of the output window.

Python3

```
# caption for output window

pygame.display.set_caption("Programming World of GFG")
```

5. Create an empty array for your snowfall

Python3

```
snowFall = []
```

6. Looping to get snowfall positions

Make a loop and run to 50 times and add a snowfall in a random x,y position using random Module.

Python3

```
for i in range(50):
    x = random.randrange(0, 400)
    y = random.randrange(0, 400)
    snowFall.append([x, y])
```

7. Track time

Create an object to help track time

Python3

```
# object to track time

clock = pygame.time.Clock()
```

8. Set criteria for snowfall occurrence

Snowfall should occur until the user presses the close button and for this inside **while** loop, use a **for** loop.

Python3

```
# loop till the close button is pressed
done = False

while not done:

  # User did something
    for event in pygame.event.get():

        # If user clicked close
```

```
        if event.type == pygame.QUIT:

          # Flag that we are done so
          # we exit this loop
            done = True
```

9. Set the screen background :

Python3

```
screen.fill(WHITE)
```

10. Process the snowfall

Now use a for loop to process each Snowfall in the list :

Python3

```
for i in range(len(snowFall)):
```

11. Draw the snowfall

Python3

```
pygame.draw.circle(screen, GREEN, snowFall[i], 2)
```

12. Adding movement

Python3

```
# Move the snowFall down one pixel
        snowFall[i][1] += 1
```

```
        # If the snowFall has moved off the bottom of the screen
        if snowFall[i][1] > 400:

            # Reset it just above the top
            y = random.randrange(-50, -10)
            snowFall[i][1] = y

            # Give it a new x position
            x = random.randrange(0, 400)
            snowFall[i][0] = x

    # Go ahead and update the screen with what we've drawn.
    pygame.display.flip()
    clock.tick(20)
pygame.quit()
```

And Yes, Green snowfall has started!!

Complete Program

Python3

```
import pygame
import random
pygame.init()

WHITE = [255, 255, 255]
GREEN  = [0,255,0]
SIZE = [400, 400]

screen = pygame.display.set_mode(SIZE)
pygame.display.set_caption("Programming World of GFG")

snowFall = []
for i in range(50):
    x = random.randrange(0, 400)
    y = random.randrange(0, 400)
    snowFall.append([x, y])

clock = pygame.time.Clock()
done = False
while not done:

    for event in pygame.event.get():
```

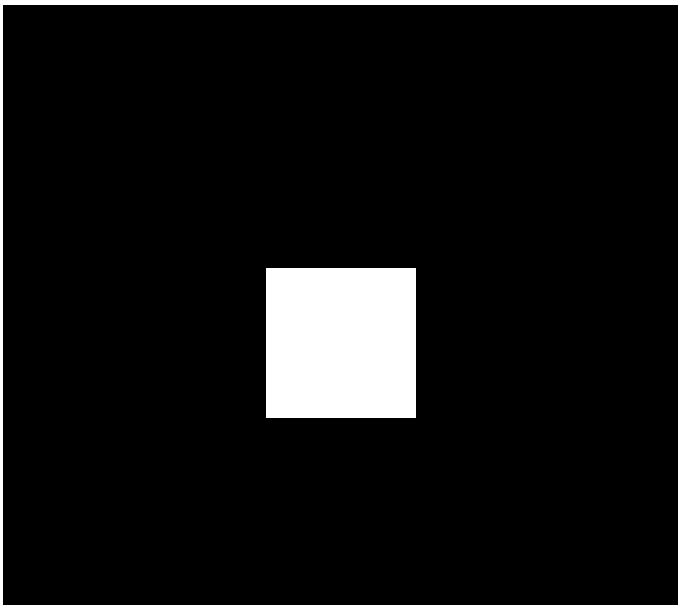
```
        if event.type == pygame.QUIT:
            done = True
    screen.fill(WHITE)
    for i in range(len(snowFall)):
        pygame.draw.circle(screen, GREEN, snowFall[i], 2)

        snowFall[i][1] += 1
        if snowFall[i][1] > 400:
            y = random.randrange(-50, -10)
            snowFall[i][1] = y

            x = random.randrange(0, 400)
            snowFall[i][0] = x

    pygame.display.flip()
    clock.tick(20)
pygame.quit()
```

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!

A anshi...

7

Previous Article

Python | Playing audio file in Pygame

Next Article

Rhodonea Curves and Maurer Rose in Python

Similar Reads

Display Snowfall using arcade library in Python

In this article, we are going to display snowfall using the arcade Python Package. The Arcade library is a modern Python Module used widely for developing 2D...

4 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

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

How to create walking character using multiple images from sprite sheet...

In this article, we will cover how to create a walking character using multiple images from a sprite sheet using Pygame. Any video or films that are made are...

5 min read

Slide Puzzle using PyGame - Python

Slide Puzzle game is a 2-dimensional game i.e the pieces can only be removed inside the grid and reconfigured by sliding them into an empty spot. The slide...

15 min read

Python - Drawing design using arrow keys in 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...

3 min read

Binary Search Visualization using Pygame in Python

An algorithm like Binary Search can be understood easily by visualizing. In this article, a program that visualizes the Binary Search Algorithm has been...

4 min read

Ternary Search Visualization using Pygame in Python

An algorithm like Ternary Search can be understood easily by visualizing. In this article, a program that visualizes the Ternary Search Algorithm has been...

5 min read

Building Space Invaders Using PyGame - Python

In this article, we're going to build a Space Bullet shooter game using PyGame in Python. The used file can be downloaded from here. Approach: Import the...

13 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