PYTHON PROJECT

- PROJECT NAME : **Bouncing Ball Simulator**

- SUBMITTING TO :

1. Mr. Mayur Dev Sewak
   General Manager , Operations
   Eisystems Services

2. Ms. Mallika Srivastava
   Trainer , Programming & Algorithms,
   Eisystems Services

1. SUBMITTING BY:

ANUJ MISHRA

Content Table

List of Figures

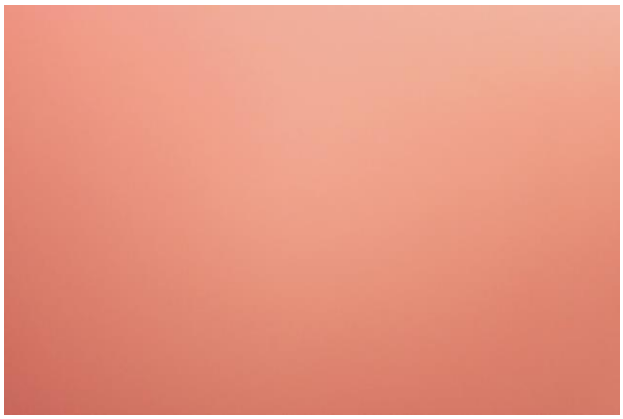
1. Screenshots shows the working of program:

    - Caption : Screenshot1, Screenshot2, Screenshot3, Screenshot4, Screenshot5
    - Page No : 8-9

2. Images and Files were used in the Project

    - Python File.py

    - Ball.jpg



    - Background.jpg

Abstract of Project

Project Title : Bouncing Ball Simulator

- It's most fun and most instructive to work on this simulation from *first principles* without using any "fancy" modules. The pygame module is a brilliant tool to create such physical simulations as it's relatively straightforward to use, and it gives you full control.

- Pygame is an open-source Python library for making multimedia applications like games built on top of the excellent SDL library.

- Using only relatively basic knowledge of the laws of motion and the effect of gravity on a falling ball, you've been able to create a reasonably realistic simulation of a bouncing ball in Python.

- Python is a multipurpose language and can be used in almost every field of development. Python can also be used to develop different type of game. Let's try to develop a simple Bouncing Ball Simulator.

Project Summary

Project Title: Bouncing Ball Simulator

- This is as simple as you can get for a bouncing animation. First, we see importing and initializing pygame is nothing noteworthy. The import pygame imports the package with all the available pygame modules. The call to pygame.init() initializes each of these modules.

- This simulation of Bouncing ball looks more attractive and with the help of pygame module another game can be created to play in endless loop.

Objectives Of Project

- To understand how to implement a bouncing algorithm, it provides you an essential knowledge to understand how the computer controls the trajectory of a sprite (e.g. ball) on the screen.
- The aim of the ball is to bounce randomly from one position to another.
- All five balls just move randomly from left to right and right to left and strike to the wall.

System Requirement Used

1. Windows 10 pro
2. Python 3.10.4
3. VS CODE
4. Command prompt for pip file installation

Text code/program

Code(a).

File Edit Selection View Go Run Terminal Help Bouncing ball simulation.py -

battery noti.py | Bouncing ball simulation.py

C: > Users > anujs > Desktop > Bouncing ball simulation.py > ball > move_ball

```python
#This program shows the simulation of 5 balls bouncing under gravitational ac
#It is also accompanied by eleastic collission with walls of the container.
import pygame,time,random
pygame.init()
screen=pygame.display.set_mode((800,600))
background=pygame.image.load('background-img.jpg')
pygame.display.set_caption('Ball Bounce Simulation')
class ball:
    ball_image=pygame.image.load('ball.png')
    g=1
    def __init__(self):
        self.velocityX=4
        self.velocityY=4
        self.X=random.randint(0,768)
        self.Y=random.randint(0,350)

    def render_ball(self):
        screen.blit(ball.ball_image, (self.X,self.Y))
    def move_ball(self):
        #changing y component of velocity due to downward acceleration
        self.velocityY+=ball.g
        #changing position based on velocity
        self.X+=self.velocityX
        self.Y+=self.velocityY
        #collission with the walls lead to change in velocity
        if self.X<0 or self.X>768:
            self.velocityX*=-1
        if self.Y<0 and self.velocityY<0:
            self.velocityY*=-1
            self.Y=0
        if self.Y>568 and self.velocityY>0:
            self.velocityY*=-1
            self.Y=568
Ball_List=[ball(),ball(), ball(), ball(), ball()]
#The main program loop
running=True
while running:
    for event in pygame.event.get():
```

⊗ 0 ⚠ 0

Code(b).

Visual Studio Code

Bouncing ball simulation.py

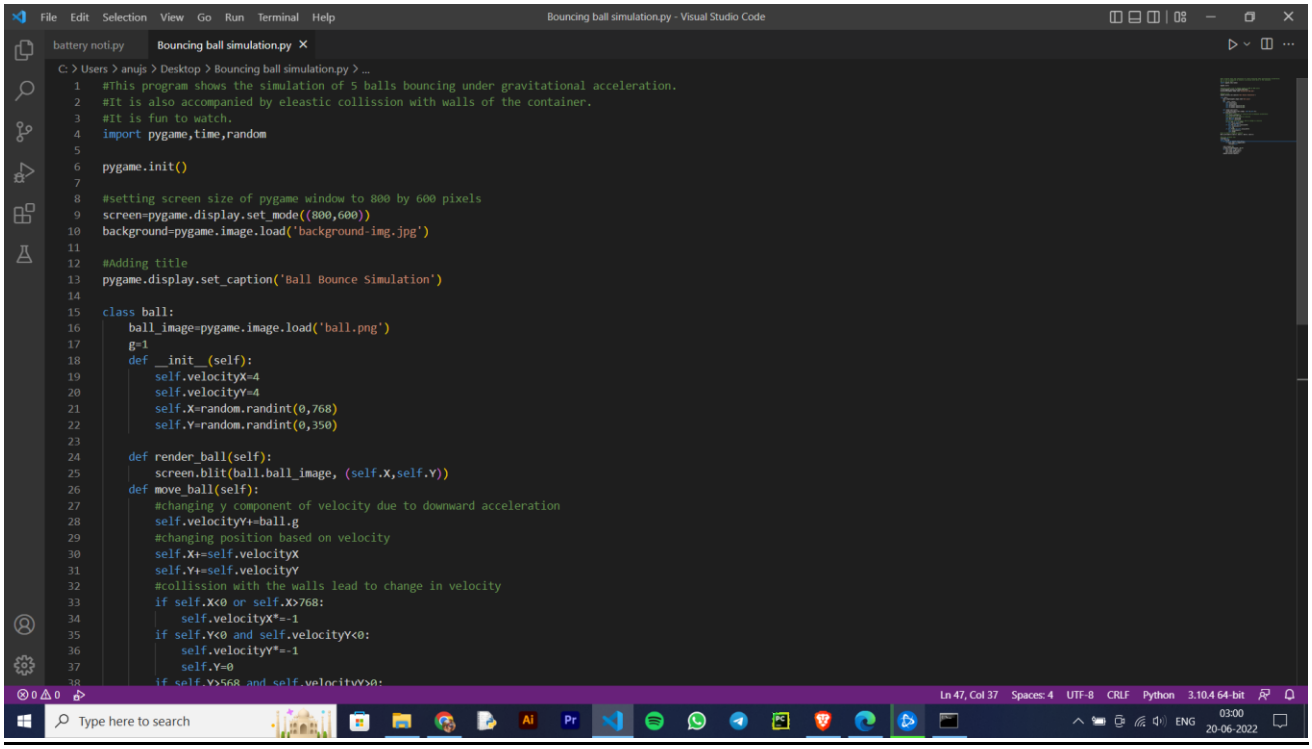
C: > Users > anujs > Desktop > Bouncing ball simulation.py > ball > move_ball

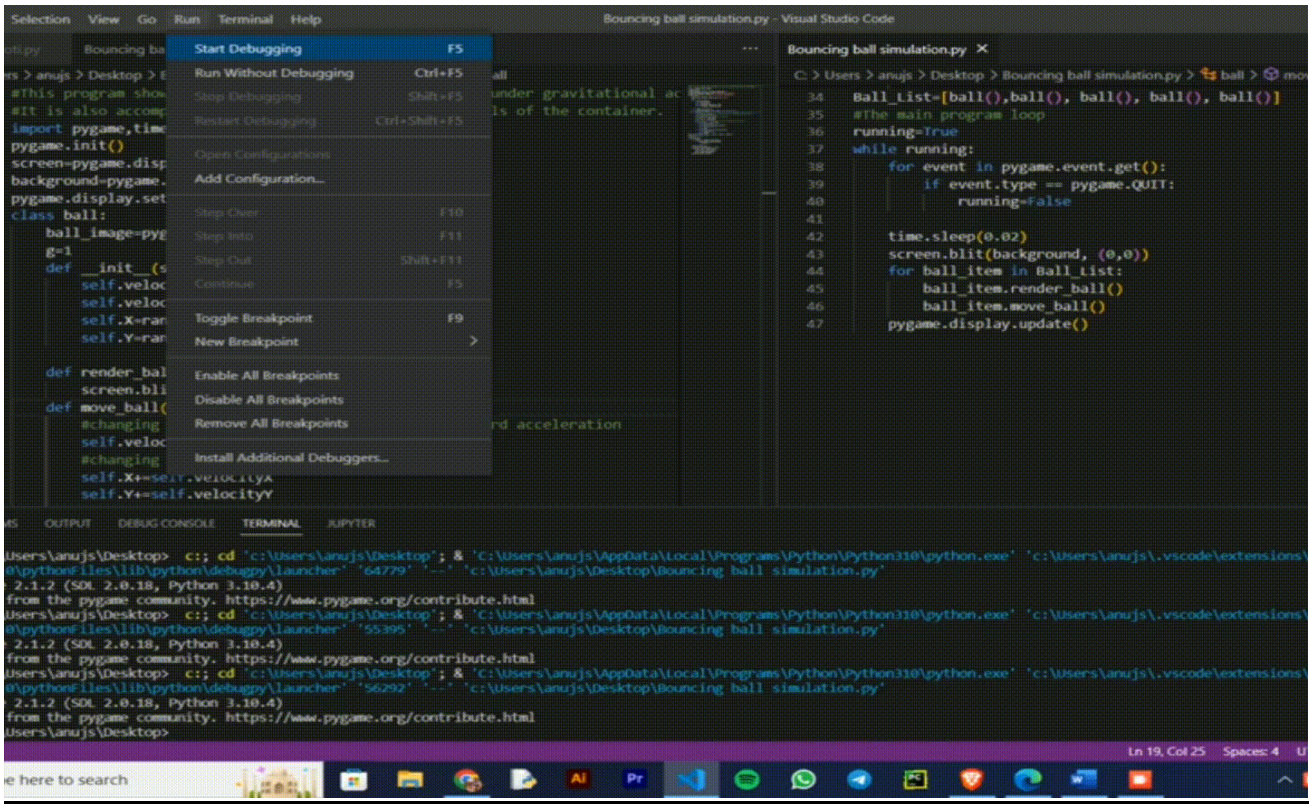
```python
Ball_List=[ball(),ball(), ball(), ball(), ball()]
#The main program loop
running=True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running=False

    time.sleep(0.02)
    screen.blit(background, (0,0))
    for ball_item in Ball_List:
        ball_item.render_ball()
        ball_item.move_ball()
    pygame.display.update()
```

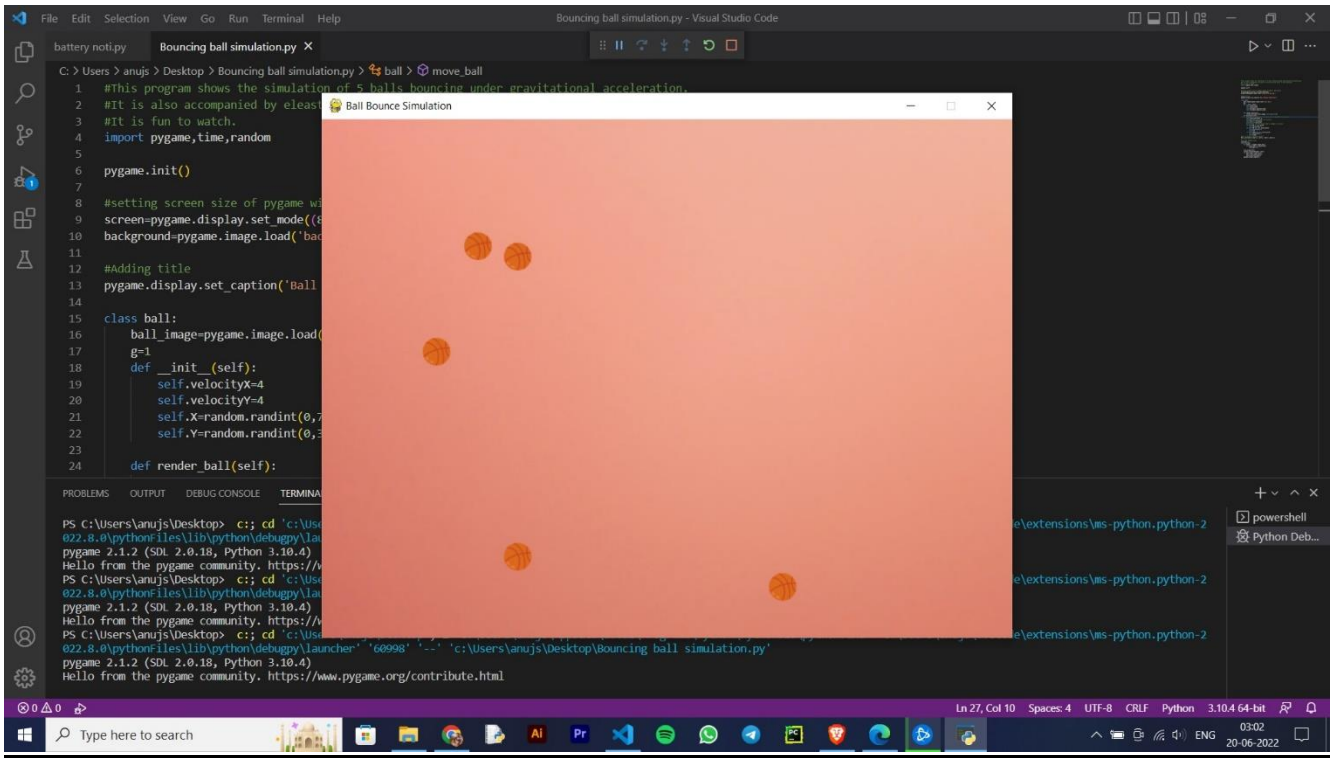
Ln 19, Col 25 Spaces: 4 UTF-8 CRLF Python 3.10.4 64-bit

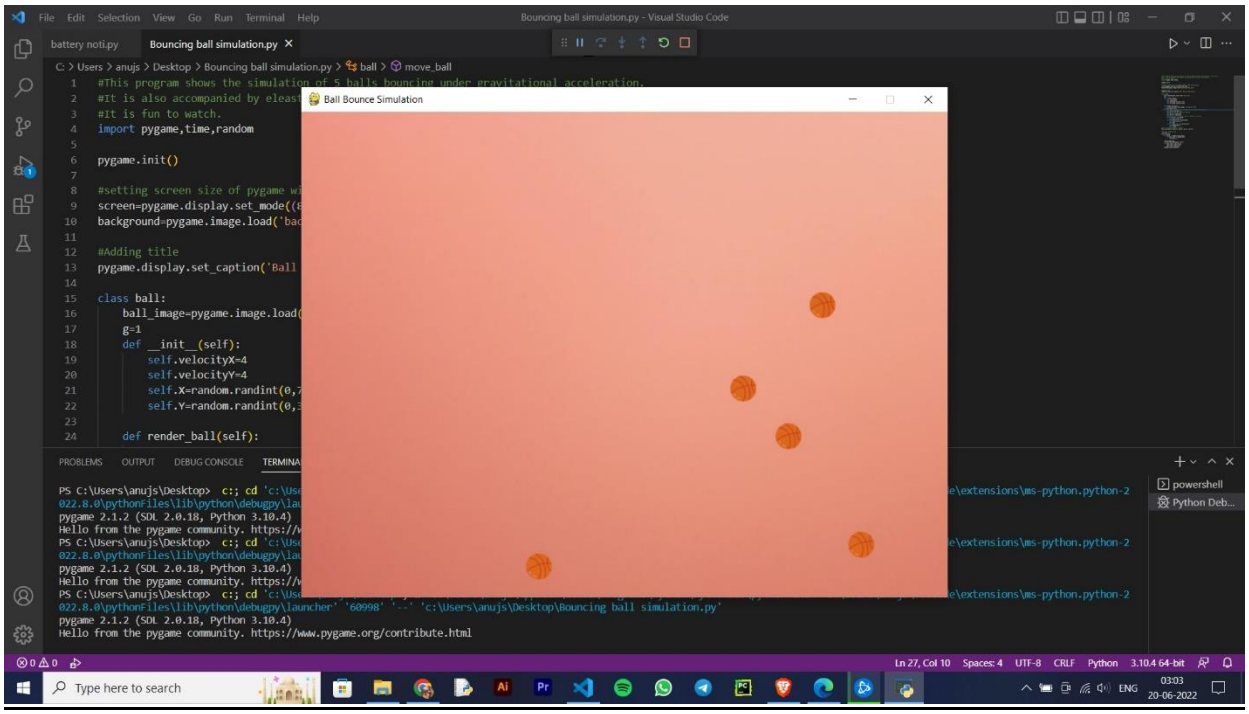
Screenshots of program





Screenshots of simulation..





References

- YouTube
- Wikipedia
- Brave 1.39.122
- And books

Thank you