

Practical No.1

(A) Aim: Setup DirectX 11, Window Framework and Initialize Direct3D device.

Code:

```
using System; using
System.Collections.Generic; using
System.ComponentModel;
using System.Data; using
System.Drawing; using
System.Text; using
System.Windows.Forms; using
Microsoft.DirectX; using
Microsoft.DirectX.Direct3D;

namespace Prac1a
{
    public partial class Form1 :
Form
    {
        Microsoft.DirectX.Direct3D.Device device;

        public Form1()
        {
            InitializeComponent();
            InitDevice();
        }
        public void InitDevice()
        {
            PresentParameters pp = new PresentParameters();
pp.Windowed = true;
            pp.SwapEffect = SwapEffect.Discard;
            device = new Device(0, DeviceType.Hardware, this,
CreateFlags.HardwareVertexProcessing, pp);
        }
        public void Render()
        {
            device.Clear(ClearFlags.Target, Color.Orange, 0, 1);
device.Present();
        }
        private void Form1_Paint(object sender, PaintEventArgs e)
        {
            Render();
        }
    }
}
```

```
}  
}  
}
```

Output:



Practical No.1

(B) Aim: Loading models into DirectX 11 and rendering.

Code:

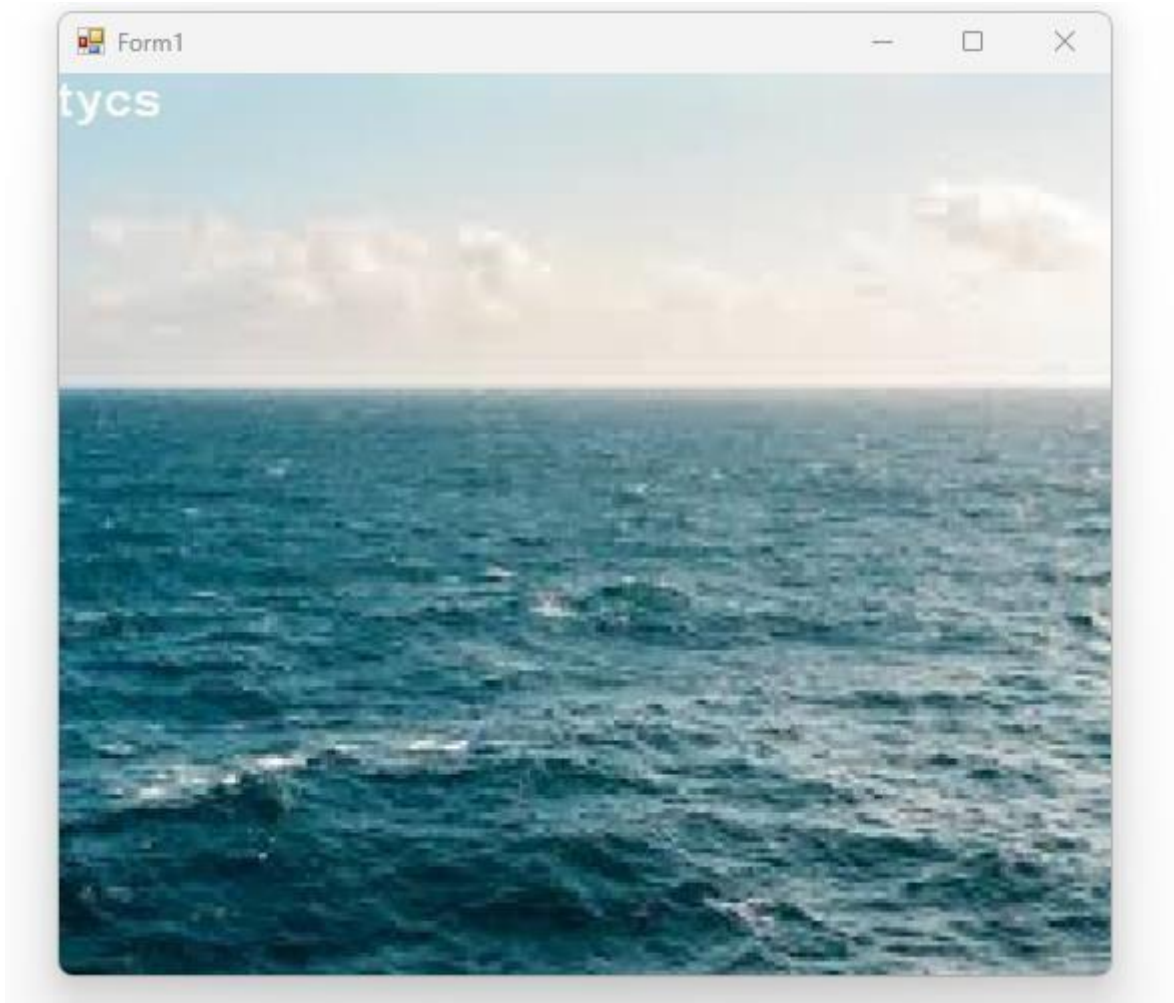
```
using System; using
System.Collections.Generic; using
System.ComponentModel;
using System.Data; using
System.Drawing; using
System.Text; using
System.Windows.Forms; using
Microsoft.DirectX; using
Microsoft.DirectX.Direct3D;

namespace Pracamit
{   public partial class Form1 :
Form
    {
        Microsoft.DirectX.Direct3D.Device device;
        Microsoft.DirectX.Direct3D.Texture texture;
        Microsoft.DirectX.Direct3D.Font font;    public
        Form1()
        {
            InitializeComponent();
            InitDevice();
            InitFont();
            LoadTexture();
        }    private void
        InitFont()
        {
            System.Drawing.Font f = new System.Drawing.Font("Arial", 16f,
            FontStyle.Regular);    font = new
            Microsoft.DirectX.Direct3D.Font(device, f);
        }
        private void LoadTexture()
        {
            texture = TextureLoader.FromFile(device, @"C:\Users\student\Desktop\sea.jpg",
            400, 400, 1, 0, Format.A8B8G8R8, Pool.Managed, Filter.Point, Filter.Point,
            Color.Transparent.ToArgb());
        }
    }
}
```

```
    }
    private void InitDevice()
    {
        PresentParameters pp = new PresentParameters();
        pp.Windowed = true;
        pp.SwapEffect = SwapEffect.Discard;
        device = new Device(0, DeviceType.Hardware, this,
        CreateFlags.HardwareVertexProcessing, pp);
    }
    private void Render()
    {
        device.Clear(ClearFlags.Target, Color.CornflowerBlue, 0, 1);
        device.BeginScene();
        using (Sprite s = new Sprite(device))
        {
            s.Begin(SpriteFlags.AlphaBlend);
            s.Draw2D(texture, new Rectangle(0, 0, 0, 0), new Rectangle(0, 0,
            device.Viewport.Width, device.Viewport.Height), new Point(0, 0), 0f, new Point(0, 0),
            Color.White);
            font.DrawText(s, "tycs", new Point(0, 0), Color.White);
        }
        s.End();
        device.EndScene();
        device.Present();
    }

    private void Form1_Paint(object sender, PaintEventArgs e)
    {
        Render();
    }
}
```

Output:



Practical No. 2

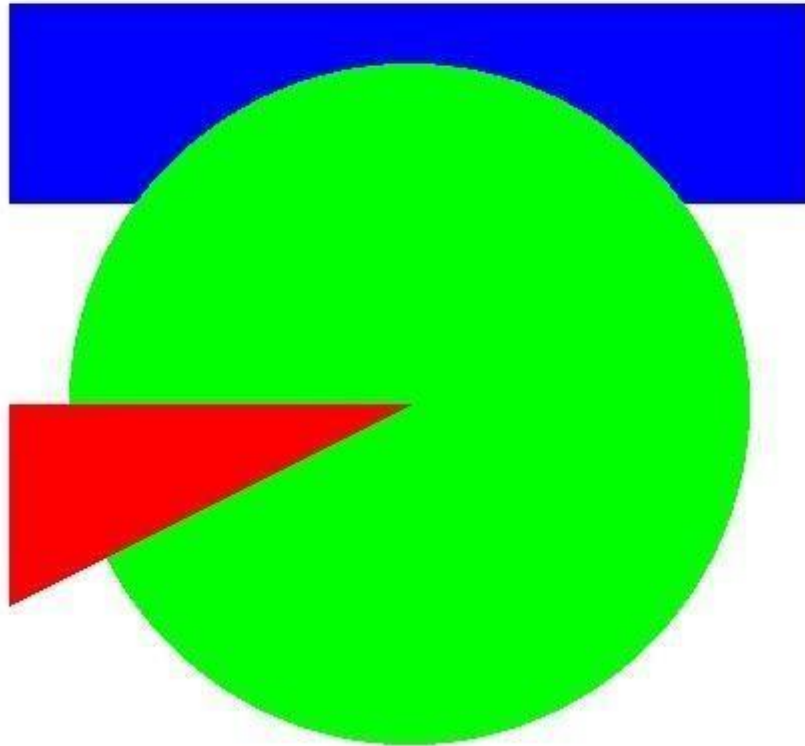
Aim: Learn Basic Game Designing Techniques with pygame.

Code:

```
import pygame
pygame.init()
window = pygame.display.set_mode((600, 600))
window.fill((255, 255, 255))
pygame.draw.rect(window, (0, 0, 255), [100, 100, 400, 100], 0)
pygame.draw.circle(window, (0, 255, 0), [300, 300], 170, 0)
pygame.draw.polygon(window, (255, 0, 0), [[300, 300], [100, 400], [100, 300]])
pygame.display.update()
```

 Output:

pygame window



Practical No. 3

Aim: Develop Snake Game using pygame.

Code: import

pygame

import time

import random

snake_speed = 20

window_x = 720 window_y

= 480

black = pygame.Color(0, 0, 0) white

= pygame.Color(255, 255, 255) red =

pygame.Color(255, 0, 0) green =

pygame.Color(0, 255, 0) blue =

pygame.Color(0, 0, 255)

pygame.init()

pygame.display.set_caption("Snake Game") game_window =

pygame.display.set_mode((window_x, window_y))

fps = pygame.time.Clock()

snake_position = [100, 50] snake_body = [[100, 50],

[90, 50], [80, 50], [70, 50]]

Name:- Jyoti Munnilal Yadav

Roll No : 74

```

fruit_position = [random.randrange(1, (window_x//10)) * 10, random.randrange(1,
(window_y//10)) * 10] fruit_spawn
= True direction = 'RIGHT'
change_to

```

```

= direction

```

```

score = 0

```

```

def show_score(choice, color, font, size):    score_font =
pygame.font.SysFont(font, size)    score_surface =
score_font.render('Score: ' + str(score), True, color)    score_rect =
score_surface.get_rect()    game_window.blit(score_surface,
score_rect)

```

```

def game_over():    my_font = pygame.font.SysFont("times new roman",
50)    game_over_surface = my_font.render('Your Score is: ' + str(score),
True, red)    game_over_rect = game_over_surface.get_rect()
game_over_rect.midtop = (window_x/2, window_y/4)
game_window.blit(game_over_surface, game_over_rect)
pygame.display.flip()    time.sleep(2)    pygame.quit()    quit()

```

```

while True:    for event in
pygame.event.get():        if event.type ==
pygame.KEYDOWN:            if event.key
== pygame.K_UP:
                change_to = 'UP'            if
event.key    ==    pygame.K_DOWN:
change_to = 'DOWN'            if event.key ==
pygame.K_LEFT:

```

```
        change_to = 'LEFT'        if
event.key == pygame.K_RIGHT:
    change_to = 'RIGHT'

    if change_to == 'UP' and direction != 'DOWN':
direction = 'UP'    if change_to == 'DOWN' and
direction != 'UP':        direction = 'DOWN'    if
change_to == 'LEFT' and direction != 'RIGHT':
    direction = 'LEFT'    if change_to == 'RIGHT'
and direction != 'LEFT':
direction = 'RIGHT'

    if direction == 'UP':
        snake_position[1] -= 10
    if direction == 'DOWN':
snake_position[1] += 10    if
direction == 'LEFT':
snake_position[0] -= 10    if
direction == 'RIGHT':
snake_position[0] += 10

    snake_body.insert(0, list(snake_position))    if snake_position[0] ==
fruit_position[0] and snake_position[1] == fruit_position[1]:
        score += 10
        fruit_spawn = False
    else:
snake_body.pop()

    if not fruit_spawn:
```

```
fruit_position = [random.randrange(1, (window_x // 10)) * 10,
random.randrange(1, (window_y // 10)) * 10]  fruit_spawn = True
game_window.fill(black)

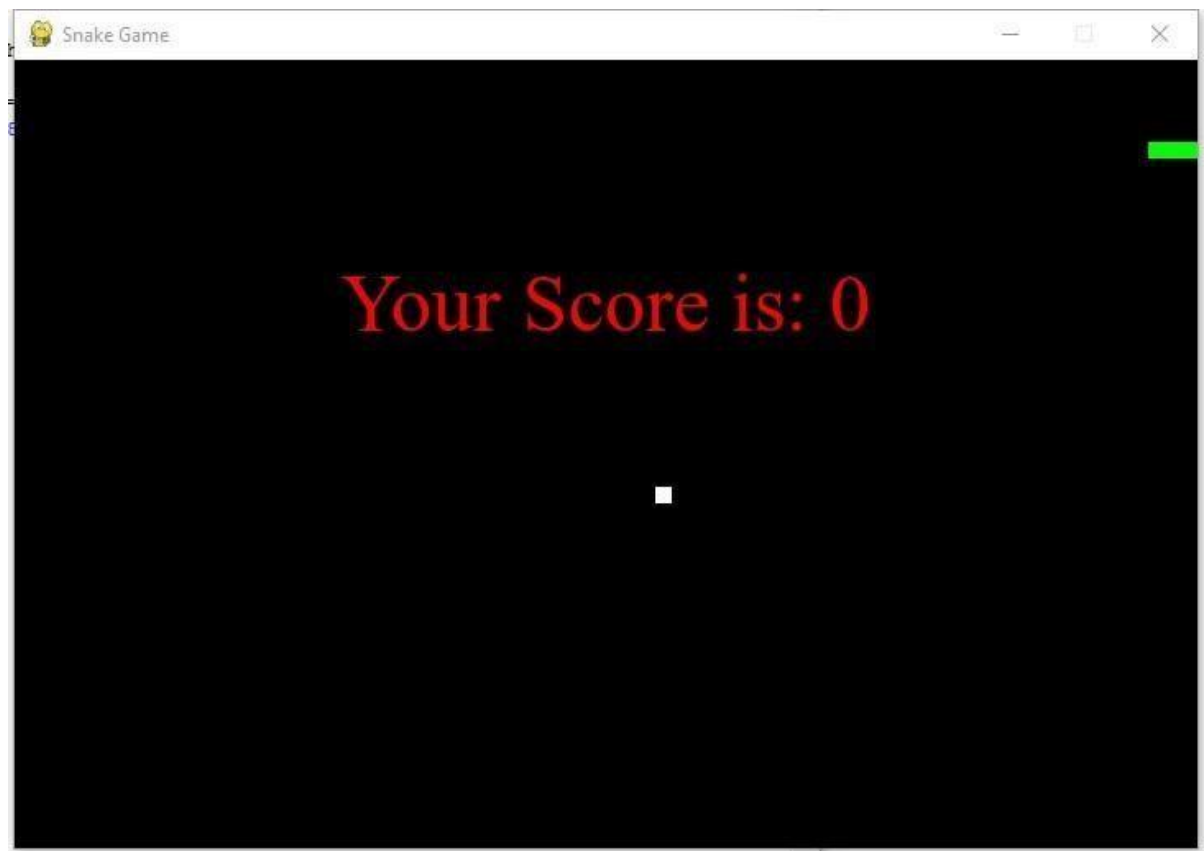
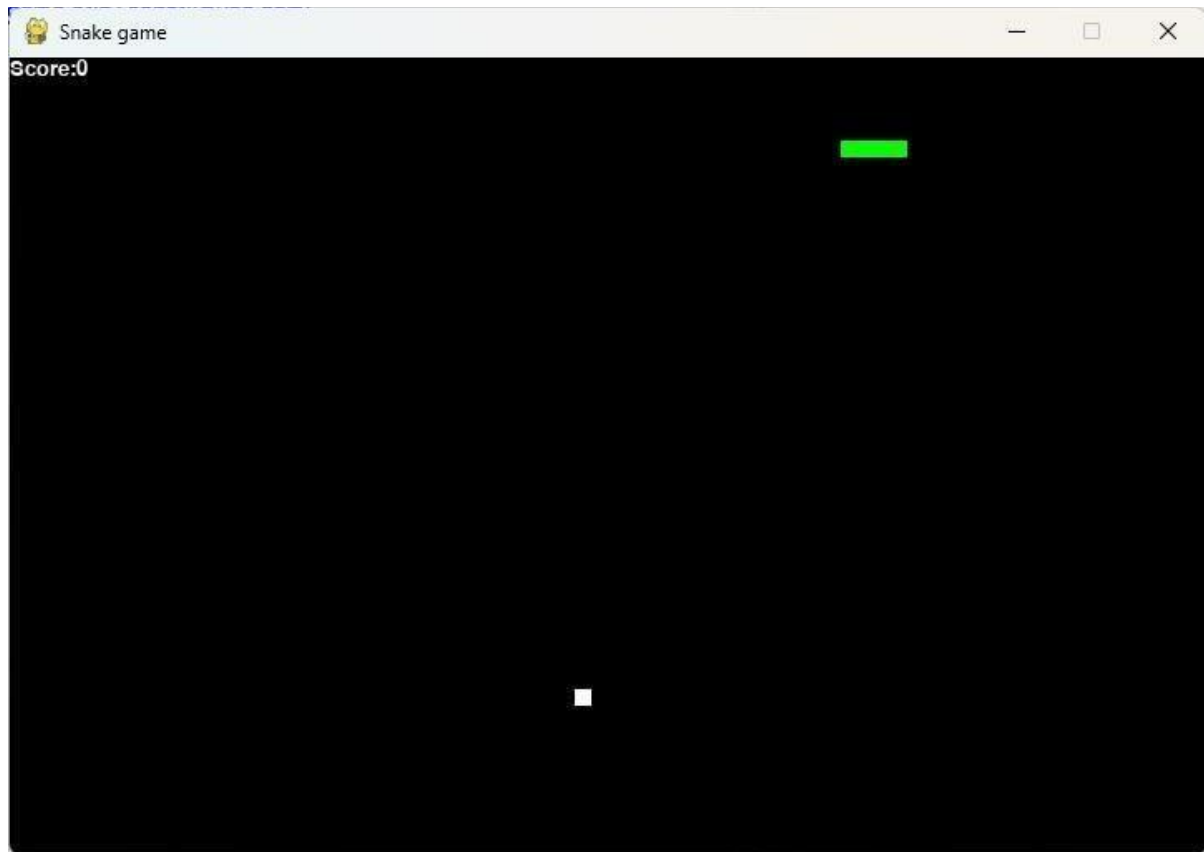
for pos in snake_body:    pygame.draw.rect(game_window, green,
pygame.Rect(pos[0], pos[1], 10, 10))

pygame.draw.rect(game_window, white, pygame.Rect(fruit_position[0], fruit_position[1],
10, 10))

if snake_position[0] < 0 or snake_position[0] > window_x - 10:
game_over()  if snake_position[1] < 0 or snake_position[1]
> window_y - 10:
game_over()

for block in snake_body[1:]:    if snake_position[0] == block[0] and
snake_position[1] == block[1]:
game_over()

show_score(1, white, "times new roman", 20)
pygame.display.update()
fps.tick(snake_speed) Output:
```



Practical No.4

Aim: Create 2D Target Shooting Game.

Code:

```
import pygame          import
random import math from pygame
import mixer

pygame.init()

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

clock = pygame.time.Clock()

pygame.display.set_caption("space shooter")

background = pygame.image.load('Space.jpg')

score_val = 0
scoreX = 5 scoreY
= 5 font =
pygame.font.Font('freesansbold.ttf', 64)

game_over_font = pygame.font.Font('freesansbold.ttf', 64)

def show_score(x, y):    score =
font.render("Points:"+str(score_val),
                True, (255,255,255))
screen.blit(score, (x, y))
```

```
def game_over():    game_over_text =
game_over_font.render("GAME OVER",
                        True, (255,255,255))
screen.blit(game_over_text,(190, 255))

mixer.music.load("music.mp3") mixer.music.play(-1)

playerImage = pygame.image.load("NRG.png") playerImage
= pygame.transform.scale(playerImage, (75, 75)) player_X =
370 player_Y = 523 player_Xchange = 0

invaderImage = []
invader_X = [] invader_Y
= [] invader_Xchange = []
invader_Ychange = []
no_of_invaders = 8

for num in range(no_of_invaders):
invaderImage.append(pygame.image.load("enemy.png"))
invaderImage[num] = pygame.transform.scale(invaderImage[num], (65, 65))
invader_X.append(random.randint(30, 737))
invader_Y.append(random.randint(22, 180)) invader_Xchange.append(1.2)
invader_Ychange.append(30)

bulletImage = pygame.image.load("lazer.png") bulletImage
= pygame.transform.scale(bulletImage, (30, 45)) bullet_X = 0
bullet_Y = 50 bullet_Xchange = 0 bullet_Ychange = 3
bullet_state = "rest"
```

```
def isCollision(x1, x2, y1, y2):  
    distance = math.sqrt((math.pow(x1 - x2,2)) +  
                          (math.pow(y1 -  
y2,2)))    if distance <= 50:  
return True    else:        return False
```

```
def player(x, y):  
screen.blit(playerImage,(x - 16, y + 10))
```

```
def invader(x, y, i):  
screen.blit(invaderImage[i], (x, y))
```

```
def bullet(x, y):    global  
bullet_state  
screen.blit(bulletImage, (x, y))  
bullet_state = "fire"
```

```
running = True while  
running:  
screen.fill((0,0,0))  
screen.blit(backgrou  
nd, (0, 0))    for event  
in pygame.event.get():  
if event.type == pygame.QUIT:  
    running = False  
  
    if event.type == pygame.KEYDOWN:  
if event.key == pygame.K_LEFT:
```

```

        player_Xchange = -1.7        if
event.key == pygame.K_RIGHT:
        player_Xchange = 1.7        if event.key ==
pygame.K_SPACE:        if bullet_state == "rest":
bullet_X = player_X        bullet(bullet_X,
bullet_Y)        bullet_sound =
mixer.Sound("lazermp3.mp3")
bullet_sound.play()        if event.type == pygame.KEYUP:
        player_Xchange = 0

    player_X += player_Xchange
for i in range(no_of_invaders):
    invader_X[i] += invader_Xchange[i]

    if bullet_Y <= 0:
bullet_Y = 600        bullet_state
= "rest"        if bullet_state ==
"fire":        bullet(bullet_X,
bullet_Y)        bullet_Y -=
bullet_Ychange        for i in
range(no_of_invaders):        if
invader_Y[i] >= 450:        if
abs(player_X-invader_X[i]) <
80:        for j in
range(no_of_invaders):
        invader_Y[j] = 2000        explosion_sound
= mixer.Sound("explosion.mp3")        explosion_sound.play()
game_over()        break

```

```

        if invader_X[i] >= 735 or invader_X[i] <= 0:            invader_Xchange[i]
*= -1            invader_Y[i] += invader_Ychange[i]            collision
=            isCollision(bullet_X,            invader_X[i],
bullet_Y, invader_Y[i])            if collision:
score_val += 1            bullet_Y = 600
bullet_state = "rest"            invader_X[i] =
random.randint(64,736)            invader_Y[i] =
random.randint(30,200)
invader_Xchange[i] *= -1

```

```

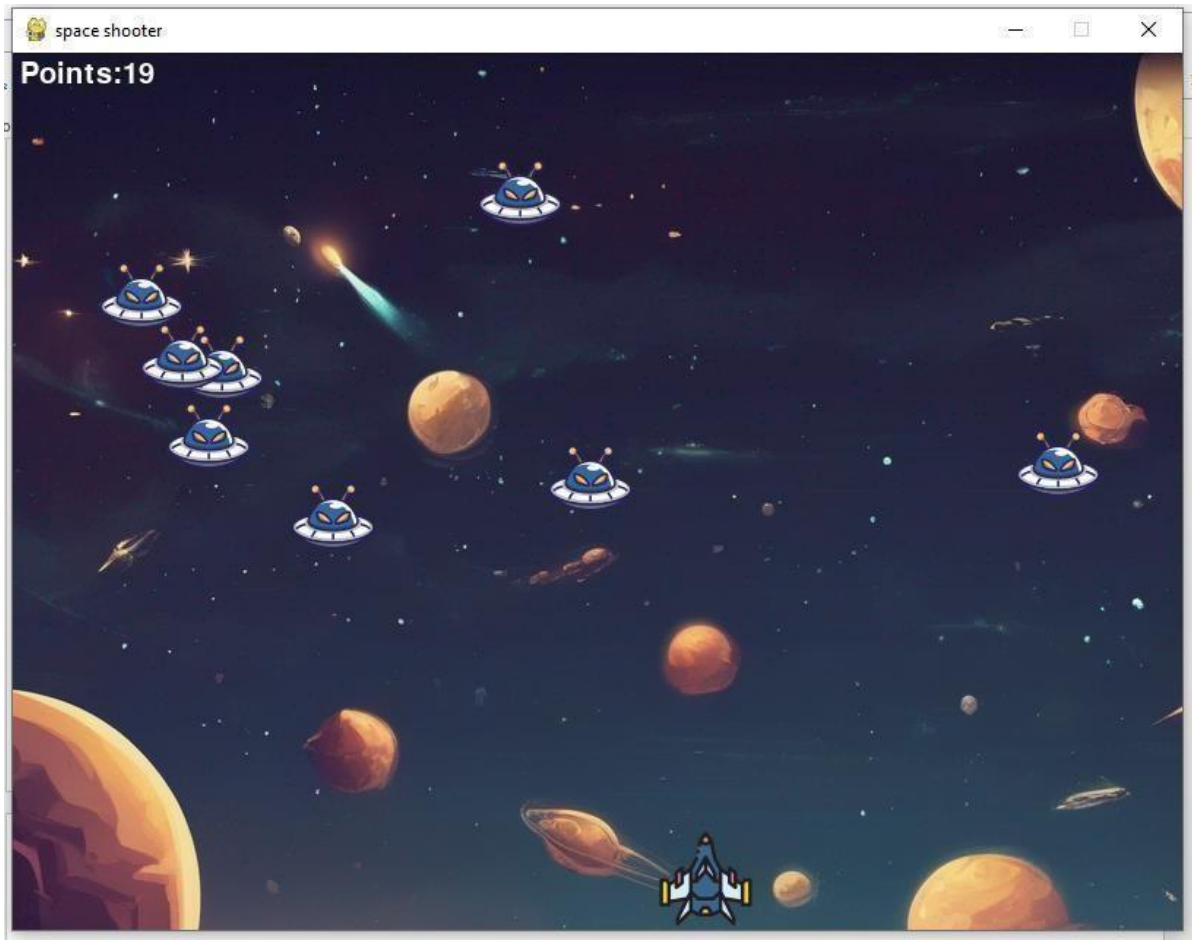
    invader(invader_X[i], invader_Y[i], i)

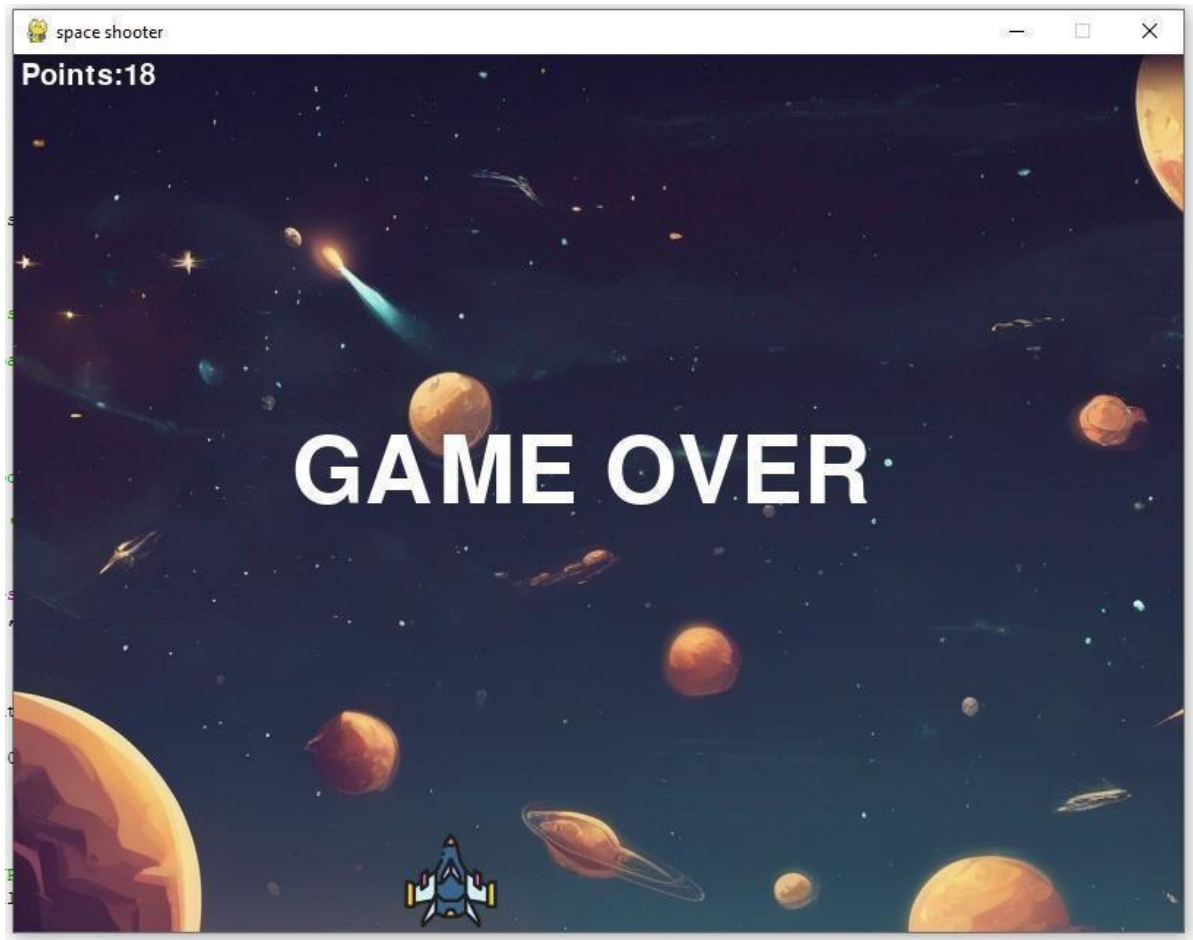
```

```

    if player_X <= 16:
player_X = 16; elif
player_X >= 750:
player_X = 750
player(player_X,
player_Y)
show_score(scoreX,
scoreY)
clock.tick(120)
pygame.display.update(
) Output:

```





Practical No.5

Aim: Create 2D Infinite Scrolling Background.

Code: import math import pygame as py

```
py.init()
```

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

```
FrameHeight = 400
```

```
FrameWidth = 600
```

```
py.display.set_caption("Infinite Scrolling in Pygame") screen  
= py.display.set_mode((FrameWidth, FrameHeight))
```

```
bg = py.image.load("img.jpg").convert()
```

```
scroll = 0
```

```
tiles = math.ceil(FrameWidth / bg.get_width()) + 1
```

```
while True:
```

```
    clock.tick(33)
```

```
    i = 0    while
```

```
    i < tiles:    screen.blit(bg, (bg.get_width() * i +  
    scroll, 0))
```

```
    i += 1
```

```
    scroll -= 6
```

```
if abs(scroll) > bg.get_width():  
    scroll = 0  
  
for event in py.event.get():  
if event.type == py.QUIT:  
    quit()  
  
py.display.update()  
  
py.quit()
```

Output:



Practical No.6

Steps for Camera Shake.

1. Open Unity.
2. Create new project.
3. Choose Universal 3D template > Name the project > create project.
4. After opening unity > choose windows > choose 'package management' and select "package manager".
5. Right click on hierarchy > select 3D object and add cube and plane > adjust the cube on the top of the plane.
6. Right click on hierarchy and add cinemachine camera.
7. Go to cube > Add components > search and add "cinemachine Imulse source" and then > search and add "Rigidbody".
8. In "cinemachine Imulse source" > Impulse type choose "Legacy" > Frequency gain and amplitude gain enter value "2" > Decay "s=1".
9. Now choose Cinemachine Camera in hierarchy and add "Cinemachine Impulse Listener".
10. Right click on Assets > Create > Scripting > Empty C# script.
11. Double click on the script > Visual studio > Copy & paste the given code > and save the code.
12. Go to unity > Your code will be visible on the inspector panel.
13. Run.

Aim: Create Camera Shake Effect in Unity.

Code:

```
using System.Collections; using
System.Collections.Generic;
using UnityEngine;
```

```
public class camerashake : MonoBehaviour {
public Transform cameraTransform; private
Vector3 _originalPosOfCam; public float
shakeFrequency;      // Use this for
initialization void Start () {
        _originalPosOfCam = cameraTransform.position;
    }
```

```
        // Update is called once per frame
void Update () {            if
(Input.GetKey(KeyCode.S))
```

Name:- Jyoti Munnalal Yadav

Roll No : 74

```
        {  
            CameraShake();  
        }  
        else if(Input.GetKeyUp(KeyCode.S))  
        {  
            StopShake();  
        }  
    }  
    private void CameraShake()  
    {  
        cameraTransform.position = _originalPosOfCam +  
Random.insideUnitSphere*shakeFrequency;  
    } private void  
StopShake()  
    {  
        cameraTransform.position = _originalPosOfCam;  
    }  
}
```

Output:



Practical No.7

Aim: Design and Animate Game Character in Unity.

Code:

```
using System.Collections; using
System.Collections.Generic;
using UnityEngine;

public class animate : MonoBehaviour {
public float speed;
Rigidbody2D rb2d;

    // Use this for initialization
void Start () {
    rb2d = GetComponent<Rigidbody2D>();
}

    // Update is called once per frame    void FixedUpdate () {
float moveHorizontal = Input.GetAxis("Horizontal");
float moveVertical = Input.GetAxis("Vertical");
var movement = new
Vector2(moveHorizontal,moveVertical).normalized*speed*Time.deltaTime;

    rb2d.AddForce(movement);
}
}
```

Output:



Practical No.8

Aim: Create Snowfall Particle effect in Unity Output:

Steps for Snowfall in Particle

Create a snowfall effect using Unity's Particle System.

Steps:

1. Open Unity and create a new project.
2. Choose a template (e.g., Universal 3D) and name your project.
3. In the Unity editor, go to GameObject > Effects > Particle System to create a new particle system.
4. Rename the particle system to "Snowfall".
5. In the Inspector panel, adjust the particle system settings:
 - Duration: Set to a suitable value (e.g., 5 seconds).
 - Looping: Enable looping.
 - Start Lifetime: Set to a suitable value (e.g., 5 seconds).
 - Start Speed: Set to a suitable value (e.g., 1-2 units/second).
 - Start Size: Set to a suitable value (e.g., 0.1-0.5 units).
6. Add a Renderer to the particle system:
 - In the Inspector panel, click on the Renderer module.
 - Set the Material to a suitable snowflake material (or create a new one).
7. Adjust Emission:
 - In the Inspector panel, click on the Emission module.
 - Set the Rate to a suitable value (e.g., 100 particles/second).
8. Add a Force Field to simulate wind:
 - Go to GameObject > 3D Object > Force Field.
 - Adjust the force field settings to simulate a gentle breeze.
9. Run the simulation:
 - Click the Play button in the Unity editor to see the snowfall effect.

