



**Student Name:** Pankaj Katoch

**Branch:** MCA(AI&ML)

**Semester:** 1st

**Subject Name:** PYTHON PROGRAMMING

**UID:** 24MCI10215

**Section/Group:** MCA(AIML)-4A

**Date of Performance:** 20-10-2024

**Subject Code:** 24CAH-606

## **TITLE: -      Serpent Challenge: Eat to Survive!**

### **Aim of the practical:**

The purpose of the project is to develop a 2D Snake Game in Pygame with more focus on the player controls and the game mechanics, and also on collision detection. This undertaking makes practical improvement of programming skills based on basic principles of game building which involves operating with events, drawing of graphics, and controlling the state of the game to develop a rich interactive experience which helps in understanding the basic tenets of coding.

### **Task to be done:**

- First, we have to import the pygame and random library.
- We have to set screen size and display the snake size and food.
- After we have to add the clock function for the time.
- We have to give the command for the directions.
- We have to use pygame font module for the score.

### **CODE FOR EXPERIMENT:**

#### **Input:-**

```
import pygame
import random
```

```
pygame.init()
```

```
# Constants width, height =
800, 600
```

```
game_screen = pygame.display.set_mode((width, height)) pygame.display.set_caption("Snake
Game")
```

```
# Snake and food initialization snake_x,
snake_y = width // 2, height // 2
change_x, change_y = 0, 0

food_x = random.randrange(0, width // 20) * 20
food_y = random.randrange(0, height // 20) * 20

clock = pygame.time.Clock() snake_body
= [(snake_x, snake_y)]
score = 0

# Font for score display font =
pygame.font.Font(None, 36)

def display_snake():    global snake_x, snake_y,
food_x, food_y, score    snake_x = (snake_x +
change_x) % width
    snake_y = (snake_y + change_y) % height

    if (snake_x, snake_y) in snake_body[1:]:
print("GAME OVER!!!")
        return False # Game over, return False

    snake_body.append((snake_x, snake_y))

    if (food_x == snake_x and food_y == snake_y):
food_x = random.randrange(0, width // 20) * 20
food_y = random.randrange(0, height // 20) * 20
score += 1 # Increase score when food is eaten    else:
        del snake_body[0]

    game_screen.fill((0, 0, 0))    pygame.draw.rect(game_screen, (0, 255,
0), [food_x, food_y, 20, 20])    for (x, y) in snake_body:
pygame.draw.rect(game_screen, (255, 255, 255), [x, y, 20, 20])

# Render and display the score
score_surface = font.render(f'Score: {score}', True, (255, 255, 255))
game_screen.blit(score_surface, (10, 10))
```

```
pygame.display.update()
return True # Game continues, return True

# Main game loop running = True
while running:
    events = pygame.event.get()
    for event in events:
        if event.type == pygame.QUIT:
            running = False
        elif event.type == pygame.KEYDOWN:
            if event.key == pygame.K_LEFT and change_x == 0:
                change_x = -20
            elif event.key == pygame.K_RIGHT and change_x == 0:
                change_x = 20
            elif event.key == pygame.K_UP and change_y == 0:
                change_y = -20
            elif event.key == pygame.K_DOWN and change_y == 0:
                change_y = 20

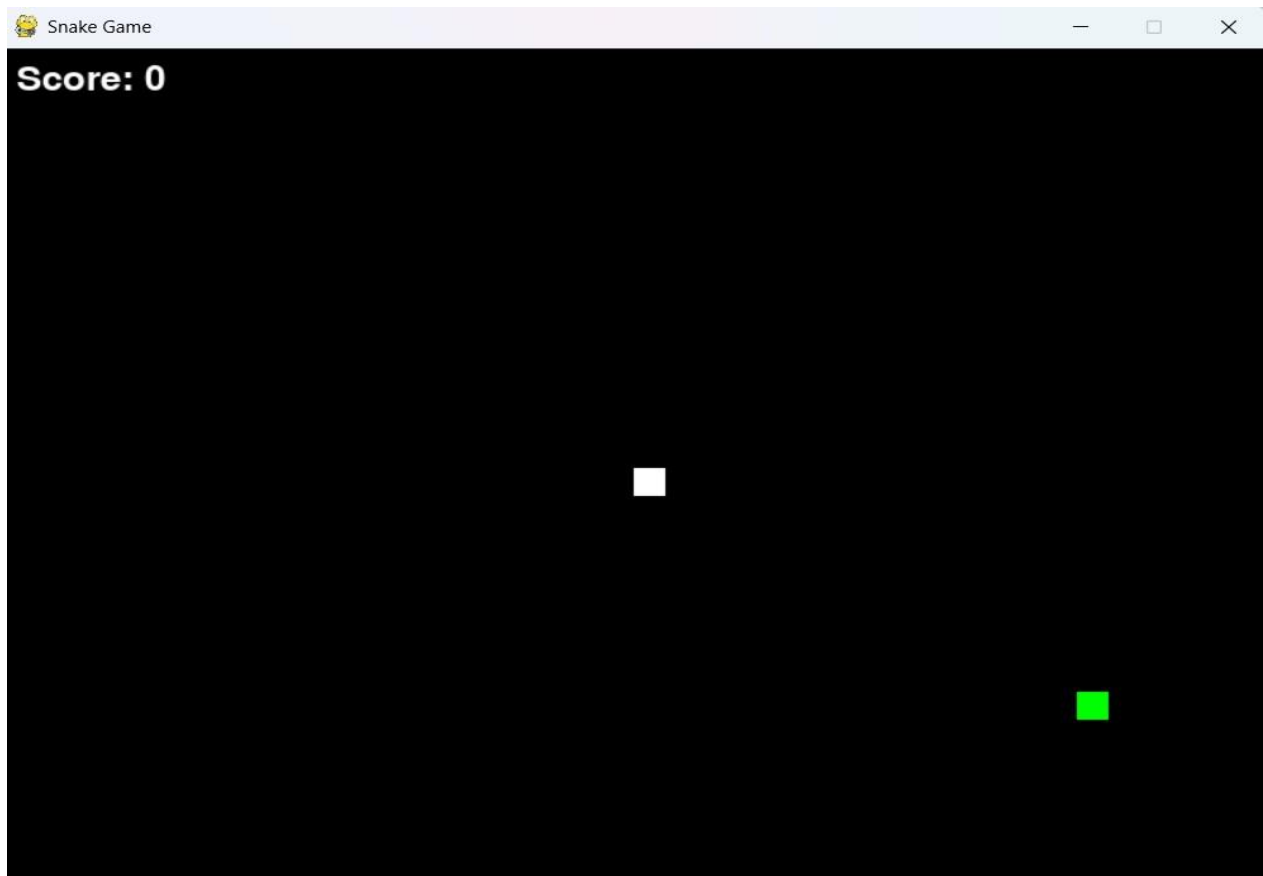
    if not display_snake(): # Check if game should continue
        break # Exit the loop if game is over

    clock.tick(10)

# Game over message
game_screen.fill((0, 0, 0))
game_over_surface = font.render(f'GAME OVER! Your score: {score}', True, (255, 255, 255))
game_screen.blit(game_over_surface, (width // 2 - game_over_surface.get_width() // 2, height // 2))
pygame.display.update()

# Wait for a few seconds before quitting
pygame.time.wait(3000)
pygame.quit()
```

**Output: -**



**Learning outcomes (What I have learned):**

- Learned how to use the pygame library.
- Learned how to thick of the body.
- Learned about clock tick function and uses.
- Understand the basics of pygame module.
- Learned hoe some basic concepts to create 2d games.