Lecture 11: Pygame, Snake & Paint

1. Project 1: Snake Game

The Snake game is a classic arcade game where a snake moves around the screen, eating food and growing in size. The game ends when the snake collides with the screen boundary or itself.

Step 1: Initialize Pygame and Set Up the Screen

We start by initializing Pygame, setting the screen size, and defining colors.

```
import pygame
import random

# Initialize Pygame
pygame.init()

# Screen settings
WIDTH, HEIGHT = 500, 500
```

```
screen = pygame.display.set_mode((WIDTH, HEIGHT))
pygame.display.set_caption("Snake Game")

# Colors
WHITE = (255, 255, 255)
GREEN = (0, 255, 0)
RED = (255, 0, 0)
BLACK = (0, 0, 0)

# Clock for controlling FPS
clock = pygame.time.Clock()
```

Step 2: Define the Snake and Food

We define the snake’s properties and functions for movement.

```
# Snake variables
snake_pos = [100, 50]
snake_body = [[100, 50], [90, 50], [80, 50]]
direction = 'RIGHT'
change_to = direction

# Food position
food_pos = [random.randrange(1, WIDTH//10) * 10, random.randrange(1, HEIGHT//10) * 10]
food_spawn = True
speed = 15
```

Step 3: Handle Events

We handle key inputs to control the snake’s movement.

```
def handle_events():
    global change_to
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            quit()
        elif event.type == pygame.KEYDOWN:
            if event.key == pygame.K_UP and direction != 'DOWN':
                change_to = 'UP'
            elif event.key == pygame.K_DOWN and direction != 'UP':
                change_to = 'DOWN'
            elif event.key == pygame.K_LEFT and direction != 'RIGHT':
                change_to = 'LEFT'
            elif event.key == pygame.K_RIGHT and direction != 'LEFT':
                change_to = 'RIGHT'
```

Step 4: Update Snake Position

The snake moves in the direction specified by the user.

```
def move_snake():
    global direction, snake_pos, food_spawn

    if change_to == 'UP':
        snake_pos[1] -= 10
    elif change_to == 'DOWN':
        snake_pos[1] += 10
    elif change_to == 'LEFT':
        snake_pos[0] -= 10
    elif change_to == 'RIGHT':
        snake_pos[0] += 10

    # Snake body growing mechanism
    snake_body.insert(0, list(snake_pos))
    # If the snake eats the food
    if snake_pos == food_pos:
        food_spawn = False
    else:
        snake_body.pop()

    if not food_spawn:
        food_pos[:] = [random.randrange(1, WIDTH//10) * 10, random.randrange(1, HEIGHT//10) * 10]
    food_spawn = True
```

Step 5: Draw Everything

We render the snake, food, and check for collisions.

```
def draw():
    screen.fill(BLACK)

    # Draw snake
    for pos in snake_body:
        pygame.draw.rect(screen, GREEN, pygame.Rect(pos[0], pos[1], 10, 10))

    # Draw food
    pygame.draw.rect(screen, RED, pygame.Rect(food_pos[0], food_pos[1], 10, 10))

    pygame.display.flip()
```

Step 6: Check for Collisions

The game ends if the snake hits the screen boundaries or itself.

```
def check_game_over():
    if snake_pos[0] < 0 or snake_pos[0] > WIDTH-10 or snake_pos[1] < 0 or snake_pos[1] > HEIGHT-10:
        pygame.quit()
        quit()

    for block in snake_body[1:]:
        if snake_pos == block:
            pygame.quit()
            quit()
```

Step 7: Main Loop

```
while True:
    handle_events()
    move_snake()
    check_game_over()
    draw()
    clock.tick(speed)
```

Project 2: Paint Application

This is a simple drawing program where users can draw with their mouse.

Step 1: Initialize Pygame and Set Up the Screen

```
import pygame

# Initialize Pygame
pygame.init()

# Screen settings
```

```
WIDTH, HEIGHT = 800, 600
screen = pygame.display.set_mode((WIDTH, HEIGHT))
pygame.display.set_caption("Paint App")

# Colors
WHITE = (255, 255, 255)
BLACK = (0, 0, 0)
RED = (255, 0, 0)
BLUE = (0, 0, 255)

# Brush settings
brush_size = 5
brush_color = BLACK

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

Step 2: Handle User Input

```
def handle_events():
    global brush_color, brush_size
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            quit()
        elif event.type == pygame.KEYDOWN:
            if event.key == pygame.K_r:
                brush_color = RED
            elif event.key == pygame.K_b:
                brush_color = BLUE
            elif event.key == pygame.K_UP:
                brush_size += 2
            elif event.key == pygame.K_DOWN:
                brush_size = max(1, brush_size - 2)
```

Step 3: Draw with Mouse

```
def draw():
    mouse_pressed = pygame.mouse.get_pressed()
    mouse_pos = pygame.mouse.get_pos()

    if mouse_pressed[0]:  # Left button is pressed
        pygame.draw.circle(screen, brush_color, mouse_pos, brush_size)

    pygame.display.flip()
```

Step 4: Main Loop

```
screen.fill(WHITE)  # Background color
while True:
    handle_events()
    draw()
    clock.tick(60)
```

4. Summary

- **Snake Game:**
  - Handled movement, food eating, collisions.
  - Used game loops to update frames.
- **Paint App:**
  - Used mouse events to draw.

- Allowed color and brush size changes.