Python Game (pygame)

Session 1: Introduction to Pygame

- **Overview of Pygame and its features: Explain that Pygame is a set of Python modules designed for writing video games. Discuss its features such as graphics rendering, event handling, and sound support.**
- **Setting up Pygame: Provide step-by-step instructions on how to install Pygame using pip and set up the development environment in popular IDEs like PyCharm or Visual Studio Code.**
- **Importing Pygame and initializing: Demonstrate how to import the Pygame module and initialize it using pygame.init().**
- **Creating a basic Pygame window: Guide learners through the process of creating a simple window using pygame.display.set_mode().**
- **Resizing window in Pygame: Show how to dynamically resize the Pygame window using pygame.display.set_mode() with the flags parameter.**
- **Changing screen background color in Pygame: Explain how to change the background color of the Pygame window using pygame.Surface.fill().**
- **Changing the name and icon of a Pygame window: Illustrate how to set the title and icon of the Pygame window using pygame.display.set_caption() and pygame.display.set_icon().**
- **Understanding the game loop: Introduce the concept of the game loop, explaining its role in updating the game state, handling events, and rendering graphics.**

Session 2: Drawing Shapes

- **Drawing objects and shapes in Pygame: Provide examples and code snippets for drawing various shapes such as rectangles, polygons, circles, ellipses, elliptical arcs, and straight lines using Pygame's drawing functions (pygame.draw.rect(), pygame.draw.polygon(), pygame.draw.circle(), etc.).**
  Discuss parameters such as position, size, color, and line thickness.

Session 3: Event Handling

- **Keyboard events (KEYDOWN, KEYUP)**: Explain how to handle keyboard events using the **pygame.event.get()** method and the **pygame.KEYDOWN** and **pygame.KEYUP** constants. Show how to process key presses and releases to control game characters.
- **Mouse events (MOUSEBUTTONDOWN, MOUSEBUTTONUP, MOUSEMOTION)**: Introduce mouse event handling with **pygame.MOUSEBUTTONDOWN**, **pygame.MOUSEBUTTONUP**, and **pygame.MOUSEMOTION** events. Demonstrate how to detect mouse clicks, button releases, and mouse movement.
- **Implementing basic controls for a game character**: Combine keyboard and mouse event handling to control the movement and actions of a game character, such as moving left, right, jumping, or shooting.

Session 4: Working with Images

- **Displaying images with Pygame**: Teach learners how to load and display images onto the Pygame window using **pygame.image.load()** and **pygame.Surface.blit()**.
- **Rotating and scaling images using Pygame**: Show how to rotate and scale images using **pygame.transform.rotate()** and **pygame.transform.scale()**.
- **Flipping images**: Explain how to flip images horizontally and vertically using **pygame.transform.flip()**.
- **Moving an image with the mouse in Pygame**: Demonstrate how to implement interactive image movement by tracking mouse position and updating image coordinates accordingly.
- **Using the mouse to scale and rotate an image in Pygame**: Extend interactive image manipulation to include scaling and rotating based on mouse input.

Session 5: Working with Text

- **Displaying text to a Pygame window: Show how to render text on the Pygame window using pygame.font.Font and pygame.font.SysFont.**
- **Creating a text input box with Pygame: Guide learners through the process of creating a text input box where users can type text using keyboard input.**
- **Using text as buttons: Illustrate how to create interactive buttons using text labels and mouse events, allowing users to click on them to trigger actions.**

Session 6: Pygame - Sounds and Time

- **Creating sound effects: Explain how to load and play sound effects using pygame.mixer.Sound.**
- **Playing sounds with Pygame: Show how to play background music and sound effects using pygame.mixer.music and pygame.mixer.Sound.play().**
- **Playing music with Pygame: Introduce background music playback using pygame.mixer.music.load() and pygame.mixer.music.play().**
- **Pygame time: Discuss various time-related functions in Pygame such as pygame.time.get_ticks(), pygame.time.wait(), pygame.time.delay(), and pygame.time.Clock for controlling frame rate and timing within the game loop.**