Process work

Workflow of drawing background

8. The program initializes the window using Raylib with a width of 800 and height of 600, setting the title to “PONG” and the target frames-per-second (FPS) to 60.

2. The `Setup()` function is called, creating an instance of the `Player1` class and assigning it to the `p1` variable.

3. Inside the main loop, while the window should not close, the program proceeds to draw on the canvas.

4. The canvas is cleared with a black background, and a white line is drawn in the middle of the canvas at y-coordinate 300.

5. The `Update()` function is called, which in turn triggers the player’s movement, collision detection with the walls, and rendering of the player on the canvas.

6. The canvas is updated with the new frame, displaying any changes made during the update.

7. Steps 4 to 6 are repeated continuously as the game loop continues running.

8. If the window is closed, the program exits the loop and closes the window using Raylib.

Workflow of ball collision with screen edges

1. The game initializes by setting up the window with dimensions 800x600 and the title "Pong" using Raylib. The target frames-per-second (FPS) is set to 60.

2. The initial positions and speeds for the circle are defined, with the circle starting at the center of the screen.

3. In a continuous loop controlled by `Raylib.WindowShouldClose()`, the circle's position is updated based on its speed in the horizontal and vertical directions.

4. Collision detection with the screen boundaries is implemented to ensure the circle stays within the screen. If the circle reaches the edges, its direction is reversed accordingly.

5. Within each iteration of the loop, the canvas is cleared with a black background, and the circle is drawn at its updated position using `Raylib.DrawCircle()`.

6. The loop continues to update the circle's position, handle collisions, and redraw the circle on the canvas.

7. This loop repeats until the window is closed by the user, triggering the termination of the game loop.

8. Upon closing the window, the game window is closed using `Raylib.CloseWindow()`.