Akashdeep Vasisht

Assignment 2

Pong Game

Java File: Ball

Changes Made:

- Removed unnecessary global variables for individual ball properties (width and height) and combined them into a single variable mBallSize.
    - Simplifies the code and reduces redundancy. Instead of managing two separate variables for width and height, we can use a single variable to represent the ball's size, making the code cleaner and more maintainable
- Made minor optimizations by using compound assignment operators (*=) in the increaseVelocity method.
    - By using these operators, they provide a more concise way to update variables

Java File: Bat

Changes Made:

- Simplified the constructor by directly calculating mXCoord and yCoord.
    - By directly calculating X & Y, we can get rid of creating intermediate variables. This makes the code more straightforward and easier to read.
- Removed an extra set of if statements for handling bat movement.
    - Math.max and Math.min to ensure the bat doesn't go off the screen.
    - Replaced the previous set of if statements for handling bat movement with a more concise approach. Instead of having separate conditions for LEFT and RIGHT, I used Math.max and Math.min to ensure that the bat doesn't go off the screen. This reduces redundancy and makes the code cleaner.

Java File: Pong Activity

Changes Made:

- Created a getDisplaySize helper method
    - If I ever need to get the display size in another part of my code, I can now use the getDisplaySize method without duplicating the same code.

Java File: Pong Game

Changes Made:

- Put the sound initialization code in its own method, initializeSoundPool(Context context)
    - Separating the sound initialization into its own method makes the code more modular. Each method should ideally have a single responsibility. By isolating the sound initialization, I created a clear and reusable function for setting up sound resources.
    - Smaller, well-named methods are easier to read and understand
    - If I need to initialize sound in another part of your code or in another class, you can easily reuse the initializeSoundPool method without duplicating the same code.
    - Separating the sound initialization makes it easier to test the sound-related functionality independently, as you can call initializeSoundPool and test it separately from the rest of the game logic.
- Added mCanvas Line
    - To show my name on the game screen, and adjusted the x and y numbers to make sure it sits in the right corner