Assignment 8

GUI application for animation images

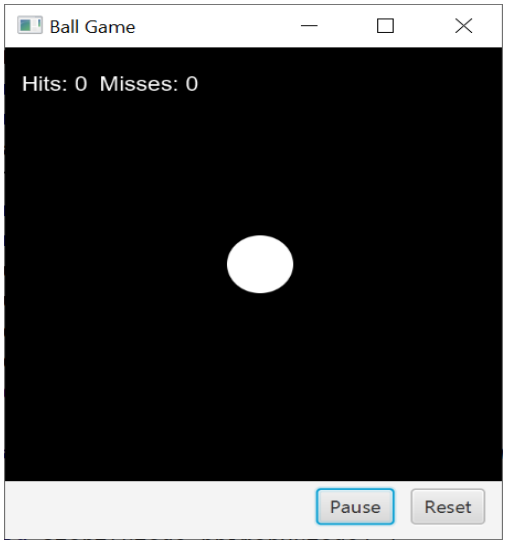
Objectives:

- Build a graphical application for animating images
- Implement mouse event listeners and animation.

Program:

In this assignment, you are going to build a simple clicking game.

Please follow the following steps:



Step 1:

Create a JavaFX Application with a central graphics pane and buttons below it. The graphics pane should have a black background. You will need to have data members for the number of hits, misses, and other needed information. It will start by displaying the current number of hits and misses at the top of the panel. Padding and alignment should be similar to the example image/video. Choose the size for the ball and graphics pane to be similar to the example.

Step 2:

Create an AnimationTimer inner class. When the game starts, the first circle should appear and move across the window horizontally. When it completely leaves the window, increase the misses count by one. Then another circle appears and moves across the screen.

When the user has missed 5 balls, the game should end and display a game over message in the middle of the graphics pane. Their final hits/missed count should still be visible and/or be included in the game over message.

Step 3:

Create event handlers to make the buttons function. The reset button should completely reset the game as if the application was just started. It should do this if the game is running, paused, or ended.

The pause button should pause the game in progress and if clicked again, the game should continue. When the game is paused, the animation should stop and clicking on the ball should have no effect. The pause button should also have no effect if the game has ended.

Step 4:

Setup an event handler for when the mouse is clicked. If the user clicks on a circle, they have scored a hit and the circle should disappear. When the user scores a hit, the next circle's speed should be **slightly** higher. If the user misses, do nothing. The misses' count only increases when the circle leaves the scene.

When the game is paused or ended, clicking does nothing.

Hints:

Do not place all of your code directly into the start method and event handler/animation methods. Create at least two useful helper methods for code that needs to be called from multiple locations. One example, many of the same things need to happen when the application is started and when the reset button is clicked.

Bonus:

Add your own addition(s) to the game. **Choose from one or more of the following or make up your own. Clearly document what your addition is at the top of your java file.**

- Add an animation when the circle is hit. The circle gets smaller, breaks up, etc.
- Multiple circles on the screen at the same time.
- Add a nice background (easy, so it is not worth a lot, but fun)

Change the ball into an object like the one you have used in previous labs. (You can use an object you have drawn in a previous lab) As before, do your own thing. Do not copy anything from someone else.

Submit a single ZIP (Compressed) file to BrightSpace containing:

The source code for the program.

Marking Scheme

- [10] JavaFX basics
- [10] Layout with appropriate Texts
- [20] Button Event Handler.
- [20] Clicking Event Handler
- [20] Animation
- [10] Break code into functions
- [5] Documentation: Javadoc for all the methods headers, purpose of the program, inline comments
- [5] Programming Style: Meaningful variable names and constants following the conventions, consistent indentation.
- [10] Bonus