

Animation with the Timer Class

Due Date: 11:59 PM, September 10, 2019

Objectives

When you complete this assignment you will be able to:

- Write a Java program to display an animation using AWT and Swing.

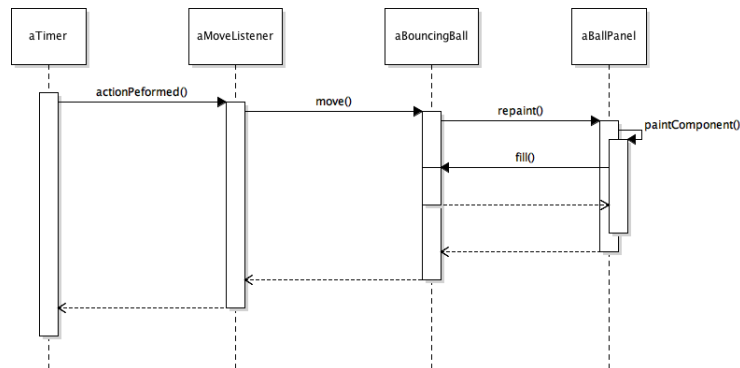
Assignment

In this assignment you are given a program which displays two balls in its frame which has a black background. The two balls are instances of the **BouncingBall** class which is a subclass of **SmartEllipse** which adds the capabilities of moving a specific number of pixels in the x and y directions as well as the ability to bounce off of the edges of the frame. Your lab assignment is to finish adding a timer to the application which will initiate the ball movement and repainting the **JPanel**. The speed specifies the number of pixels the ball moves in the x and y directions that the block is currently moving.

The **BouncingBall** class implements the **Mover** interface. So, it must implement a **move** method. Since changes to the ball object are not visible to the user until the **JPanel** is repainted, the **BouncingBall** class must maintain a reference to the **JPanel** it is drawn on as a peer object. So, the sequence becomes:

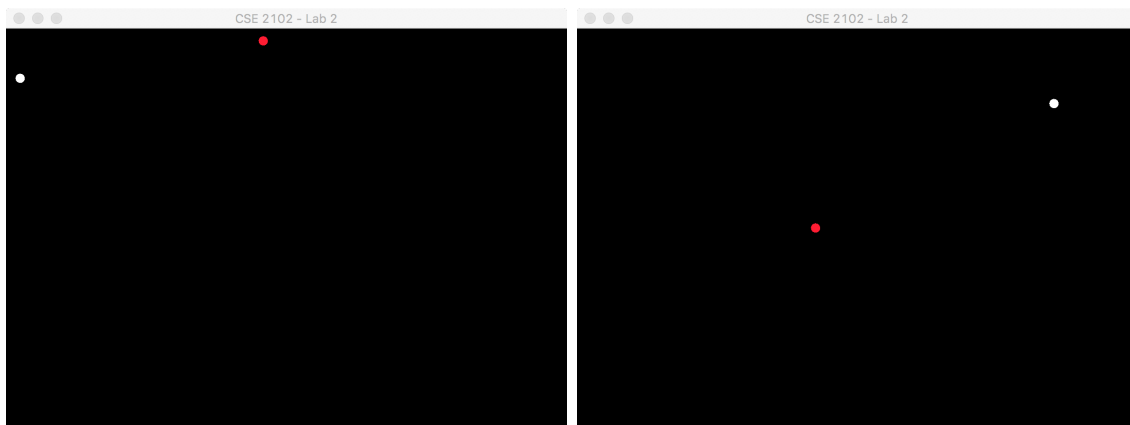
1. The timer object, in this case an instance of **MoveTimer**, generates an **ActionEvent** and invokes **actionPerformed** on the **MoveListener** which is a private inner class in the **MoveTimer** class.
2. The **actionPerformed** method is responsible for sending a **move** message to the **BouncingBall**.
3. The ball moves itself to the next position and must tell its panel to refresh. So, it sends a **repaint()** message to the **BallPanel**.
4. The **BallPanel** **repaint()** method invokes **paintComponent()** on itself which delegates redrawing the balls to the ball objects by calling the **fill** method.

See the following sequence diagram which illustrates how the timer is used to initiate the actions of moving the ball(s) and refreshing the **JPanel**.



The specific changes that need to be made are finishing the MoveTimer class' constructor and adding the code to create and start the timer to the BouncingBall class' constructor.

The following screenshots show the program at different points during the program's execution with the ball objects in different positions.



Grading

You are free to get whatever help you need in lab class to complete the assignment correctly. But, you must type in the code yourself. If you are unable to turn the work in during the lab class, you must complete it and submit it by 11:59PM that evening.