Homework #5
08-600 Java & J2EE Programming

Assigned 10/1/2014
Due 10/13/2014 by 11:59pm

Please put your program in a file called Game.java. To turn your homework in, put Game.java and any other files in your solution into a zip file called AndrewID_HW5.zip. Upload this zip file to Blackboard using the link provided on the Blackboard page for Homework 5. Please follow all the relevant conventions for indentation and naming. Put a comment at the top of each of your Java files that includes your name, Andrew ID, course number, and date. Do not put your code into a package.

Overview

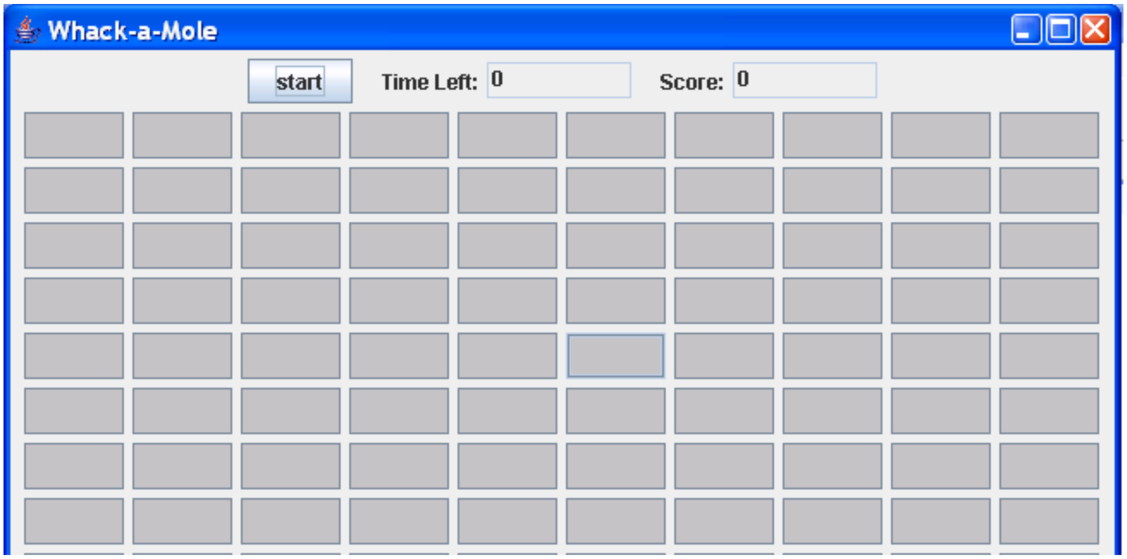
Develop a Whack-a-Mole game using Swing. Allow a player to play the game for 20 seconds before the game stops. Clicking on a mole gives you a point.

Part I

40 points

The figure below shows a mock up of a Whack-a-Mole computer game. Build the GUI which must include a “Start” button, a label and text field for the time, and a label and text field for the score. There must be many buttons, each representing a mole and it’s “mole hole”. You can choose the number, placement, and content of the buttons, as long as there are at least 10 buttons (representing 10 holes, each hole containing a mole).

There must be an “up” configuration for buttons indicating that the mole has popped his head out of his hole. (In Part IV, clicking on an “up” button will give the player a point.) Similarly, there must be a “down” configuration for buttons to indicate that the mole’s head is not popped up. (Clicking on a “down” button will not give the player a point.) At the start of the game, all buttons are in the “down” configuration.



Time Left: 0

Score: 0

Part II

20 points

When the start button is clicked:

(a) Disable the start button so that it cannot be clicked again until the game is over. (Do this in the actionPerformed() method while you are starting up the game.)

(b) create and start a timer thread.

The timer thread must display a countdown clock in seconds from 20 to 0 in the time text field. The timer thread must also re-enable the start button 5 seconds after the game ends. Roughly these are the steps:

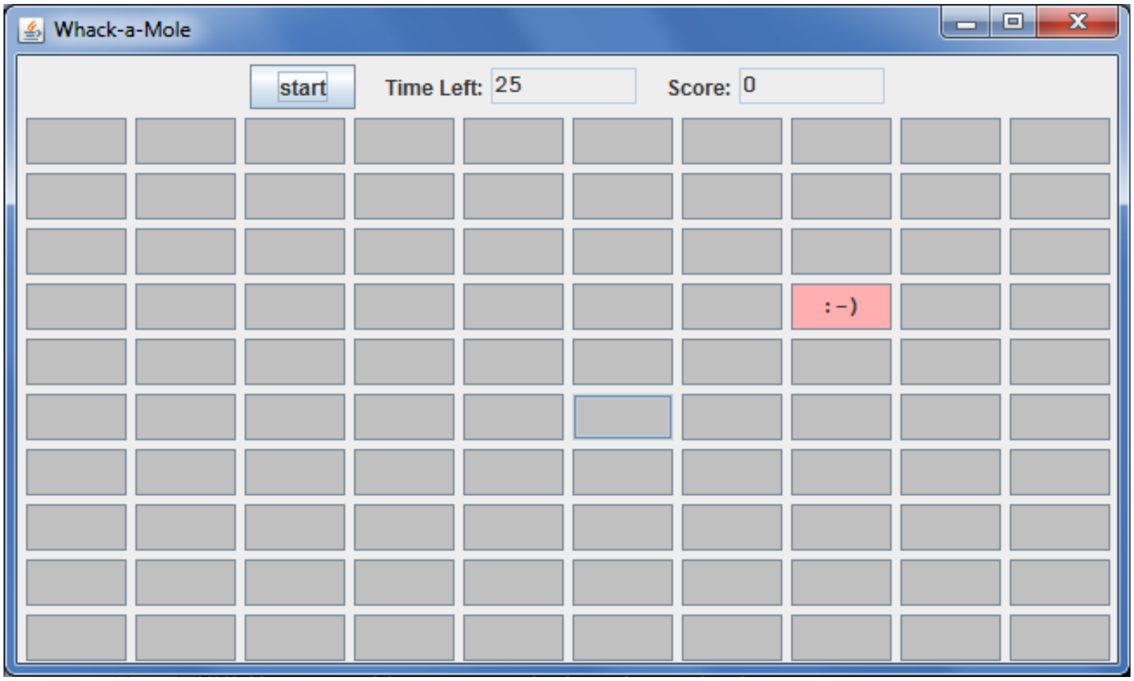
```
1. Set count to 20
2. Display count
3. While count is greater than zero
       o Sleep for one second
       o Subtract one from count
       o Display count
4. Sleep for five more seconds
5. Enable start button
```

Part III

20 points

Also in the **`actionPerformed()`** method (after the timer thread is created and started), create and start a mole thread for each of the buttons. Each mole thread should cause its “mole” to pop up by changing the button into the “up” configuration. The mole should pop up for a random amount of time that’s more than ½ second and less than 4 seconds. You can fine-tune the “up” time as necessary (e.g., You could decide ½ second is too short for the minimum up time, so, for example, you might choose to make the “up” time range from 1 to 4 seconds.) Once a mole goes down, the mole must stay down for at least two seconds before popping up again. A mole may not pop up once the timer reaches zero. It’s up to you whether to make the all the moles go down when the timer gets to zero. (The Lights program in Lecture 9 shows an example of generating random integer numbers.)

Here’s an example of what the mole might look like when it’s “up”. You can use different colors, text, and/or graphics for the mole buttons.



Time Left: 25

Score: 0

Part IV

20 points

When a mole button is clicked, check to see if the mole is “up” in this mole hole. If an “up” mole button is clicked:

1. Increment the score and display it in the score text field.
2. Change the button to display an indication that the mole has been hit. For example, perhaps the button would then show a frowny face :-(
3. After some delay, the button should return to the “down” configuration. (This can be the remaining time the mole thread would have left the button “up” if were not clicked.)

Note: If the button is clicked multiple times when a particular mole is up, the score should not increment more than once. Also, if the timer is zero (and the button is still “up”), the score should not increment.