

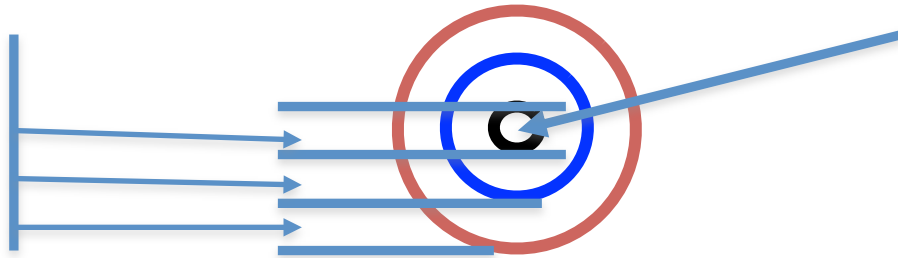
CISC 110 Lab 2

Work on this lab during the lab time. Be sure you understand everything well, in order to prepare for the course tests and final project.

In this lab, you will write further code that could be used as part of a shooter game, in which a player is being flown around in a plane and must shoot a stationary bull's eye target, building on your work from Lab 1.

Part 1: If-Else Statements

The target is composed of three concentric rings: an outer ring, a middle ring, and the inner circle (Bull-s eye). The diameter of the inner circle and the width of both the middle ring and outer ring is 20 pixels. The center point of the target is at position (100, 150).



You will write the code that needs to be executed whenever the player shoots at the target. In the game, there is a single player, their score (initially 0), two levels, and the speed of the plane they're in (which increases when they reach the second level).

Here is what you need to do:

1. Download the file Lab1.fla (the link labeled Lab 1 Solution) from the course web page, or start with your solution to Lab 1, whichever you prefer.
2. As in Lab 1, assign and display the value of the player's last shot, using a trace statement. However, don't always assume the player has hit a bull-s eye. Change the assignments to sometimes show the player has hit the middle or outer ring or not hit the target at all and check if the rest of the program specified in the next steps works correctly for each case.
3. Increase the score if the player has hit the target. Increase the score according to which part of the target is hit. If the bull's eye is hit, increase the score by the amount in the constant `BULLS_EYE`. If the middle or outer ring is hit, increase the score by the amount in `MIDDLE_RING` or `OUTER_RING`.

To simplify this, assume the bull's eye has been hit if both the x value of the player's shot is equal to the x value of the center of the target plus or minus 10 pixels and the y value of the player's shot is equal to the y value of the center of the

target plus or minus 10 pixels (since the center circle's diameter is 20 pixels). This is actually the square that contains the center circle; we'll see how to be accurate later in the course. Calculate the middle and outer ring similarly.

4. Determine whether the player has reached the next level. If the score is greater than 50, increase the player's level by setting level 1 to false and level 2 to true and double the speed (since at a higher level the plane goes faster making it more difficult to shoot the target).
5. Check if the score and level is updated correctly when the player hits different parts of the target. Do this by changing the x and y values of the player's shot and then displaying on the screen the player's score and the values of the level variables. Include multiple shots, so level 2 is sometimes reached.
Your output after one shot might look like this:

```
Last shot: ( 95,170 )  
Ania's score: 30  
Level 1: true  
Level 2: false
```

Part 2: Using Functions and Defining Simple Functions

Now let's try using some methods in the Math class. Assume that we now have 3 players, all shooting at the target, and that scores can be any ints, e.g, 34, 52, and 29.

1. Define 3 variables to store the scores of three players: p1Score, p2Score, and p3Score. Give these variables some sample scores to try them out.
2. Write an if-else statement that determines which of the three scores is the largest, and thus the winner. If p1Score is the largest, player 1 is the winner; if p2Score is the largest, player 2 is the winner; if p3Score is the largest, player 3 is the winner. Assume that there are no ties. Display the result with a trace statement.
3. Now accomplish the same result as step 2, but instead of writing an if-else statement, use the Math.max method. Display the result with a trace statement.
4. Next find the average of the three scores, so each player can see where they stand in relation to the others. To find the average, add the three scores and then divide the result by 3. Since scores are integers, the result of the division should be rounded to the nearest integer, so use the Math.round method. Again, display the result. Use a variable of type Number to hold the average, so you can see the result with the fractional component if you don't round the number.
5. Define your own simple function that displays a congratulatory message of your choice whenever a bulls-eye is scored, using trace statements. Add a line to your script that calls this function where you check whether a bulls-eye has been scored.