CS 101 Test Program #1

Due: Monday, October 24 at 11:55 PM

A test program is a program that you complete on your own, without the help of others. It is a form of take-home exam. You may consult your text, your notes, your lab work, or our on-line examples and web pages, but use of any other source is forbidden. You may not discuss these problems with anyone aside from the lab instructors and course professors.

Complete each of the following problems, commenting your code well.

You are encouraged to adapt the code from your labs or our class examples.

Your work will be automatically collected from your **dev/cs101/TP1** folder. Please be sure your files are in the right place. Late days cannot be used for the test program. Your assignment will lose 10 points for each 15 minutes that it is late. To avoid a late penalty, do not work on the test program after 11:55 PM.

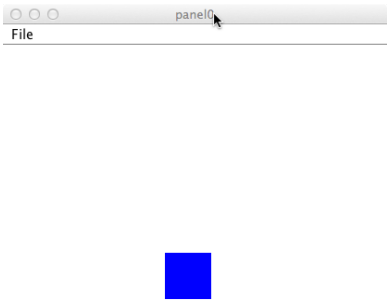
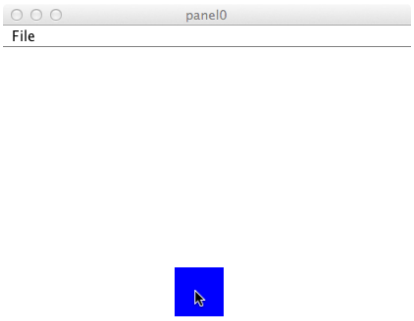
You can see movies of each of these programs executing on the course website.

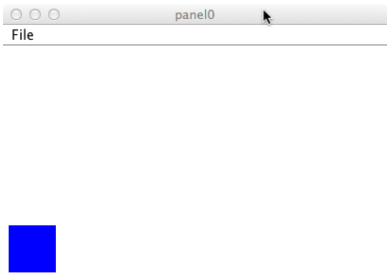
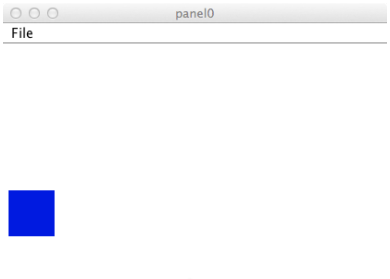
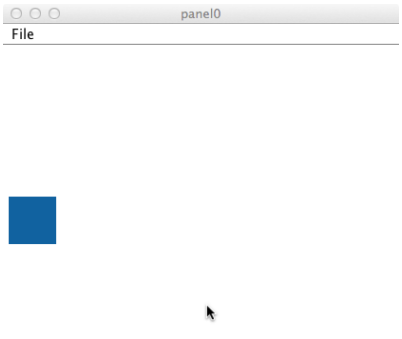
Program 1: Running Box

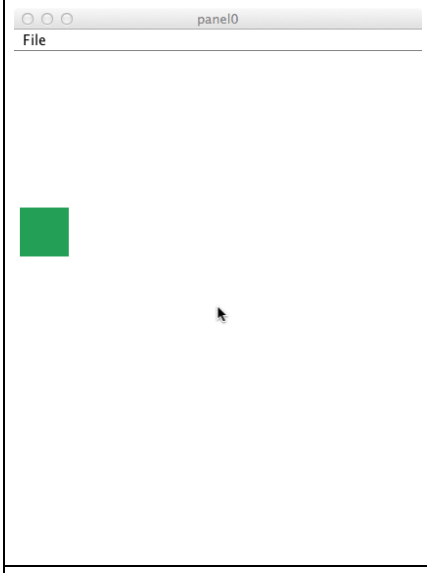
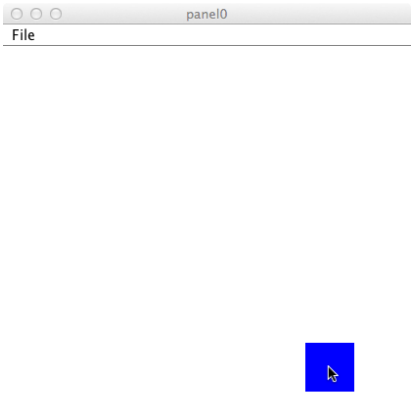
Put the solution to this problem in a project named **Box** inside your **TP1** folder.

When your program starts, it should draw a draggable, pure, blue box centered on the canvas. When the user clicks on the canvas (outside the box), the blue component of the box's color should decrease by 20 and the green component of its color should increase by 20. When it cannot get any more green and any less blue, it should stop changing color on a click. When the user clicks on the box, it should "run away" by jumping to a random location on the canvas (the entire box should still be visible) and it's color should change back to blue. The box's color should also return to blue when it is dragged.

Here are some screenshots:

	Initially the box is blue.
	When the user clicks the box, …

	… the box moves to a different location on the canvas (chosen randomly). The entire box should still be seen on the canvas.
	This image and the next 2 show the box getting less blue and more green as the user clicks the canvas multiple times.
	

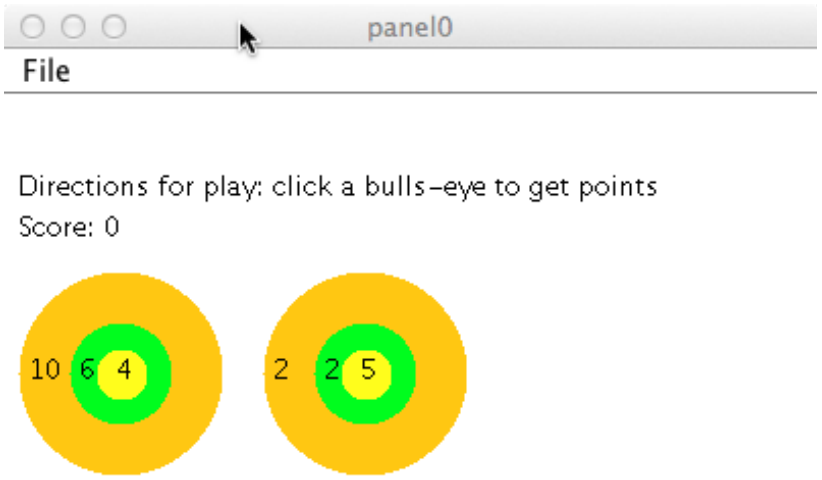
	
	Clicking or dragging the box, changes the box color back to the original blue color it had when the program started to run.

You should be able to write this program using a single class.

Program 2: Dart Game

Put the solution to this problem in a project named **DartGame** inside your **TP1** folder.

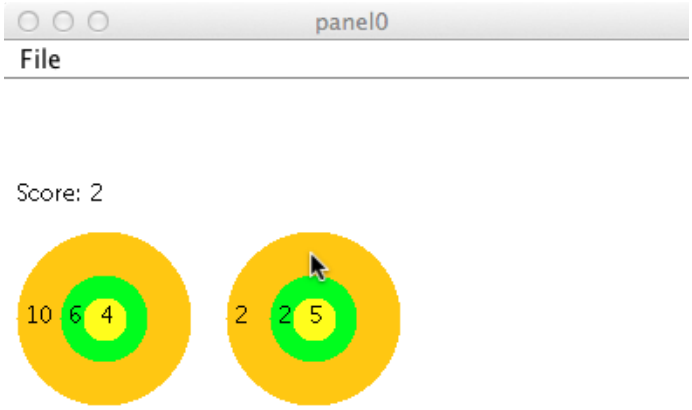
For this problem, you will write a program to simulate a Dart game. When the game begins, 2 bulls-eyes, a message saying "Directions for play: click a bulls-eye to get points", and a message giving the current score are visible on the canvas as depicted in the image below.

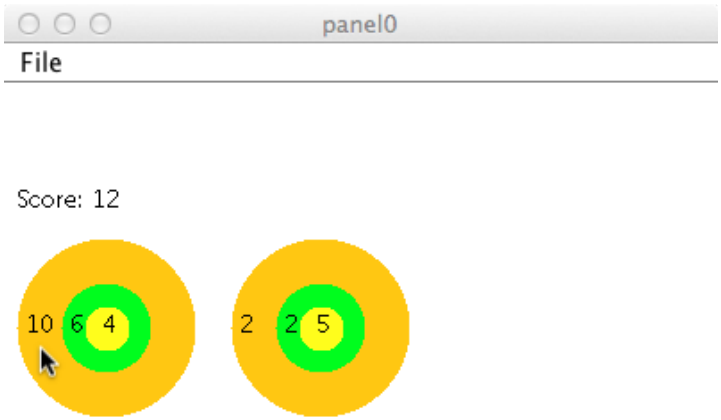


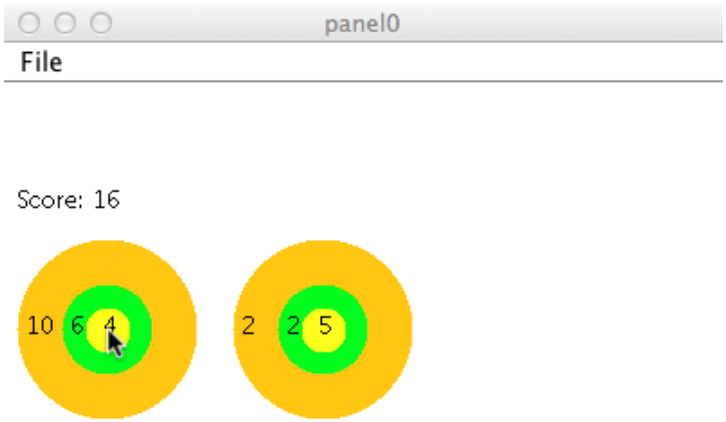
As shown above, each ring of a bulls-eye should be worth from 1-10 points. The point values for each ring should be assigned randomly. A label should display the number of points for each ring.

Users can accumulate points by clicking inside any of the rings of a bulls-eye, a user's score should increase by the amount shown on the clicked upon ring. If the user misses and the canvas is clicked outside both bulls-eyes, remove 1 point. The first time a user clicks a bulls-eye, the directions message should disappear. Anytime a user's score changes, the onscreen display of the user's score should be updated. When the mouse exits the window, the score is reset to 0 and the directions are displayed again.

The following image and the next 3 show the updated score associated with individual mouse clicks.


panel0
File
Score: 2
10 6 4
2 2 5


panel0
File
Score: 12
10 6 4
2 2 5


panel0
File
Score: 16
10 6 4
2 2 5

For this problem you will write 2 classes:

1. BullsEye:
   A BullsEye object made of 3 concentric circles. The outer circle is orange, the middle circle is green, and the center is yellow as shown above. The only positional information the BullsEye constructor should be given is where the *center* of the BullsEye should be on the canvas. It should also be told the size of the outer circle. The code for calculating each target contribution to the score after a click should be situated in BullsEye.

2. DartGame:
   This class is your window controller.

Grading

Grading of style

10	Compiles without error
5	Comments
5	Boolean conditions
5	If statements
5	Variables: Instance / local / parameter
5	public / private
5	Good choices for names
5	Good use of constants
5	Indentation
50	**Total**

Grading of correctness

Program 1:

2	Starting screen
5	Dragging
5	Random location change when clicked
3	Reset box color on drag
5	decrease blueness & increase greenness on canvas click
5	Stop increasing when maximally green and maximally blue
25	**Total**

Program 2:

5	Starting screen
5	score update and removal of directions
5	Random assignment of the ring scores
10	BullsEye class
25	**Total**