CPSC 1045: Lab 6

Functions, variable scope and program structure

Introduction

In this lab you will learn about functions, the next lab will cover event driven programming with functions. Functions allow us to organize blocks of code that we write and giving these blocks of code names. This allows code to be reusable and can keep blocks of code isolated from one another. In this lab you will write several functions and then use then in your main program.

Part A: writing and using functions

An example of how the lab should work can be found at:
http://mylinux.langara.bc.ca/~kimlam/lab6/lab6a.html

Note: The steps are recommendation and you are free to proceed, as you like.
For part A only **drawObject()** can have side-effects, and that side effect is limited to drawing on the canvas.

Step 1: Converting to radians

Write a function called **toRad(angleDeg)** that converts degrees to radians. Using this function, you can specify your angles in degrees. The formula to convert degrees to radians is as follows.

$$\theta_{rad} = \theta_{deg}\frac{\pi}{180}$$

Step 2: Writing argument checking functions

Write a function called **isValidNumber(inputNum, validMin, validMax).** The function will take in a number and return true if the number is between **validMin** and **validMax**. The function will return false, if the number is outside the range or **inputNum** if is not a number.

Step 3: Writing drawing functions

Write a function **drawObject(ctx,x,y)** which draws a shape from a previous lab centered at **x** and **y**, you can choose the shape. If you modify the local coordinate system in anyway, be sure to use **save** and **restore** so that coordinate system entering the function is restored before the function exits.

Step 4: Putting it together

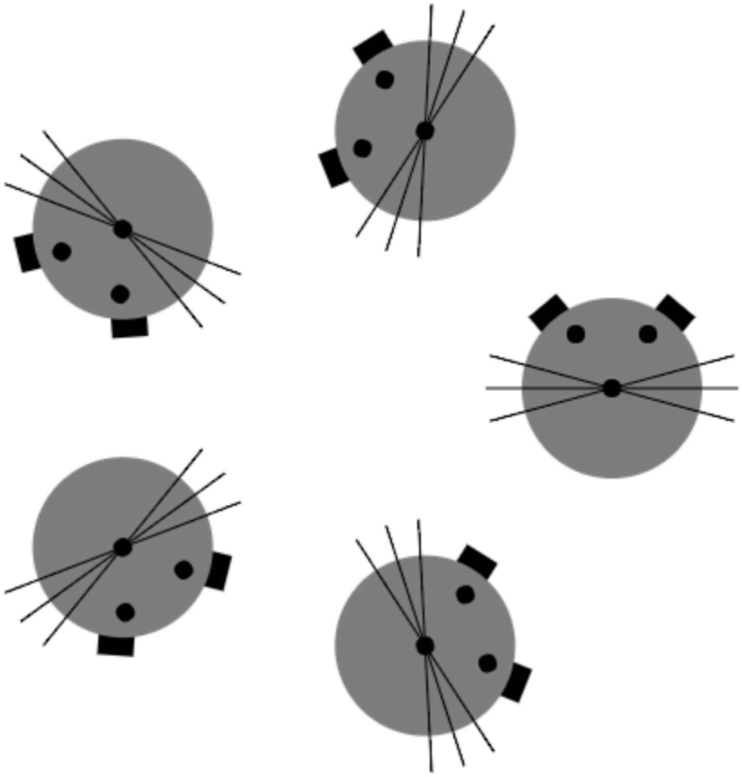
Write an app that:

1. Prompts the user for the number of objects to draw, N.

2. Verify that the input is valid; if it is not valid inform the user that the input is not valid and ask them for the value again.
3. Prompts the user for a radius, R.
4. Verify that the input is valid; if it is not valid inform the user that the input is not valid and ask them for the value again. R cannot be too large or your shape will be drawn outside the canvas. R cannot be negative.
5. Draw N objects on a circle R away from the center of the canvas. The objects should all be equally spaced along the circle.

The example below has R = 150 and N = 5

Lab 6 solution



Part B: Interactive Story

Write an interactive story, using prompt. An interactive story is where the story changes based on the user's choices.

1. You story should give the user at least two choices,
2. There should be four possible outcomes to your story, one for each path through your story.
3. Each chapter of your story should be in it's own function, this will make it easier for you to debug you program as you can invoke each chapter separately in the console.
4. Once the user has made all their choices, the complete story should be displayed on the HTML page.

   A sample interactive story can be found at the following link: http://mylinux.langara.bc.ca/~kimlam/lab6/lab6b.html