15-102 Exploring Programming with Graphics
Homework 6

Start: Wed 6.1.11
Due: Tue 6.7.11
Goal: Control #1

Course Web Site:
www.andrew.cmu.edu/course/15-100mooseNsquirrel

Reading:
As posted on the calendar web page available from the link shown above.

Assignment:
Once again for the second week in a row you are going to take a major step in programming with this homework. This is a really big program for a novice with lots of parts. Plus, you are going to add control syntax to your program. This use of control will allow your program to respond to input by the user. The method of input is up to you. Data must be displayed. Animation is required. There will be lots of global variables. Taken at a whole, this can look overwhelming. Solved in layers one part at a time, it is well within your abilities.

Preparation:
Transfer your **drawInitials(float, float, float, float)** function from your hw4.pde file or hw5.pde file into the hw6.pde file so you can use it. Experiment with the class code to see how the **keyPressed( ) and mousePressed( )** functions works and how the **if/else** constructs work. Explore the **dist( ), map( ), int( ),** and the **random( )** functions in the Processing API – they will be useful.

Animation and Control Specifications:
1. ___Your set of initials must be in the center of the window when your program starts. The initials must not be moving.
2. ___Declare a PImage global variable and initialize to to a picture of your choice. Consult the API for information on working with PImage variables. The picture should be drawn at about the same size as your initials. When your program starts, the picture must be moving and bouncing off the walls. The initial location and the values of the picture's delta variables must be randomly assigned in the setup function. Keep the range of random values reasonable.
3. ___The user must be able to move your initials in four directions (left, right, up, down) with some form of input. The movement speed must be variable and controlled by the user. The input method is up to you.
4. ___When (if) the initials move off screen, there must be an appropriate warning to the user and some form of information or hint displayed to help the user return the initials to the screen.
5. ___There must be a way the user can reset everything: return the initials to the center and reposition the picture at randomly set location and delta values.

Copyright © Jim Roberts
June 2011
Pittsburgh Pa, 15221
All Rights Reserved

It is possible (and desirable) that your initials will collide with the picture. When this happened the following must occur.

6. ___Each collision must be counted.
7. ___When a collision occurs, the delta values of the initials and picture are swapped to simulate the effect of the collision. Details in class.

You must display data – lotsa' data. These values must be labeled and must be presented in a readable and reasonable manner but one that does not significantly interfere with the ongoing animation and controls if any are drawn:

8. The location (x, y) and delta variable values of the initials (4 values)
9. The location (x, y) and delta variable values of the picture (4 values)
10. The number of collisions (1 value)
11. The total time in seconds (displayed as an int value) (1 value)

You must display instructions or labels for the user input:

12. The horizontal and vertical movement of the initials
13. The reset of the location of the initials and the picture

It is illegal to use frameCount or % to keep the move initials and target or to keep them on the screen.
You must use the if/else constructs in this homeowork.

Coding Specifications:

1. ___Use meaningful names for your global variables, functions, and arguments.
2. ___The only primitive code allowed in the **draw( )** function is the call to **background( )**. All other code must be calls of functions that you have defined.
3. ___Each function you define must do a well-defined task. Functions that do a bunch of very different tasks will be marked down.
4. ___Indenting your code is vital if others are to read it and understand it. The style of coding is up to you but there are two caveats:
   - it must be consistent throughout your program
   - we must be able to read it

   There is an auto format option in the tools tab of the Processing IDE. It might help you.

Development:

You will need a bunch of global variables. Jim used thirteen in his code. You will need to use very meaningful names for your variables if you and others are to understand your code.

Copyright © Jim Roberts
June 2011
Pittsburgh Pa, 15221
All Rights Reserved

You have to carefully and thoroughly plan your work. This begins with dividing the problem in to sub-problems. Done in a well-planned way, all of the parts should fit to make the final program. The organization of your program must be visible in the draw( ) function. Here is a print of the **draw( )** function in Jim's code:

```
void draw( )
{
  prepareWindow( );

  movePicture( );
  moveInitials( );

  drawImage( );
  drawInitials( ix, iy, iwd, iht );

  checkInitials( );
  checkCollision( );

  showStats( );
}
```

Here is one possible order to follow as you write the code:

1. _____Define **setup( )** and **draw( )** first.
2. _____Add global variables needed to draw the box and initialize them to random values in **setup( )**. .
3. _____Add global variables to control the amount of movement the box makes in each frame and initialize them to random values in **setup( )**.
4. _____Define a **movePicture( )** function that will move the picture to a new location for each frame **and** keep the picture on screen. Refer to the class code for ideas and hints.
5. _____Define a **drawPicture( )** function that will draw the picture.
6. _____Define a **moveInitials( )**function that will move the initials to a new location for each frame. If you are using the **keyPressed** variable, you *may* need to check it here and respond to key presses by the user.
7. _____Define a **checkInitials( )**function to test to see if the initials are off the screen. This function must have code to tell the user when the initials are off screen and to display some form of hint the user can do to return the initials to the screen.
8. _____Figure out someway to track the user's input to reset the screen. When this input occurs, change the variables as described in specification #5.
9. _____Add a global variable to keep track of collisions.

Copyright © Jim Roberts
June 2011
Pittsburgh Pa, 15221
All Rights Reserved

10. _____Define a **checkCollision( )** function with code to test for a collision. Read the API for the **dist( )** function and look at the class code for examples. Increment the global variable you added in step 9 when a collision occurs.
11. _____When a collision occurs, swap the delta values for the initials and picture.
12. _____Define a **showStats( )**function with code to display the required in information with labels on the screen as listed in specifications 8, 9, 10, and 11. All data must be displayed as int values.
13. Test your code thoroughly.
14. Add instructions if necessary in some reasonable visual manner

This is just a suggested development path. Feel free to use it, modify it, or develop your won. Regardless of what you do, devise a plan and follow it. Do not just throw code at this or it will never work properly if it works at all.

Grading:

There is a lot going on here. Missing major parts of the specification will cost 10% each. Defining functions that do too much will cost 10% for each function. Remember that the definition of your **draw( )** function should resemble Jim's on page 3

Advice:

This is very good practice for the exam on Friday. Start now and make a plan. Use the suggested plan in this document if you are not sure how to proceed. Take it one step at a time. You should have the picture moving on screen today. The movement of the initials should be done by tomorrow. The collision detection and reaction should be done by Friday. This leaves the stats display (including the running time), testing and other loose ends for the weekend.

Copyright © Jim Roberts
June 2011
Pittsburgh Pa, 15221
All Rights Reserved