cisc3665, fall 2011 / prof sklar
Lab IV.2: Data collection and analysis
REVISED VERSION!

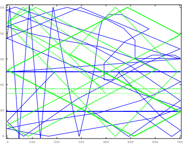
Instructions

This is a lab for unit IV, Nuts & Bolts, and does not need to be submitted.

The purpose of this lab is to give you a chance to learn about data collection.

Instructions

1. Download the sample sketch called **mypong0.zip** from the class web page (under the "syllabus" section for today's class, November 14).
2. Read the information about the game in the header comment, and try playing it a few times.
3. Then, modify the game so that the (x, y) positions of each player are saved in a data file. See below.
4. After you have saved the data in a file, you can analyze it. For example, you could plot the trajectory of each player (see figure on the right). You could also import the data file as a spreadsheet into OpenOffice or Excel and compute some statistics on it.



5. *Challenge Problem #1:* After you have done these steps, you could think about creating a *login* facility for your game. When the game starts up, ask the user to enter their username. You could try using a password too. Look up their username in a password file. You have to decide how you will create the password file. How will you put new users into the file? Will you encrypt the passwords?
6. *Challenge Problem #2:* Write another Processing program that *reads* the data file and uses the points in it to draw a plot, like the one above. Look at the **BufferedReader** section of the Processing Reference for information about and an example of opening a file for reading, and reading from a file.

Writing to files in Processing

Writing to a file in Processing is similar to writing to a file in Java.

First, you need to declare a variable to contain the file object, then bind the object to the name of a file on your computer and open the file:

```
PrintWriter datafile;
...
datafile = createWriter( "myfile.dat" );
```

Second, you can write to the file using the **println** command. For example:

```
datafile.println( x + " " + y );
```

This example writes the values of x and y to the file pointed to by the `datafile` object.

When you are done with the file, you should flush any data that is buffered and hasn't been written to the file, and then close it, as follows:

```
  datafile.flush(); // write any buffered data to the file
  datafile.close(); // close the file
```

Here are a couple of hints for this particular lab:

- There is a Processing variable called **frameCount** which stores the frame number. It starts at 0, which is its value in the `setup()` method. Then, each time the `draw()` method is called, the frame count increments.
- I suggest writing one line to the file each time the `draw()` method is called. I would put on that line the frame count, and the (x, y) position of both balls.

For more information, see the example under the **PrintWriter** entry in the Processing documentation.