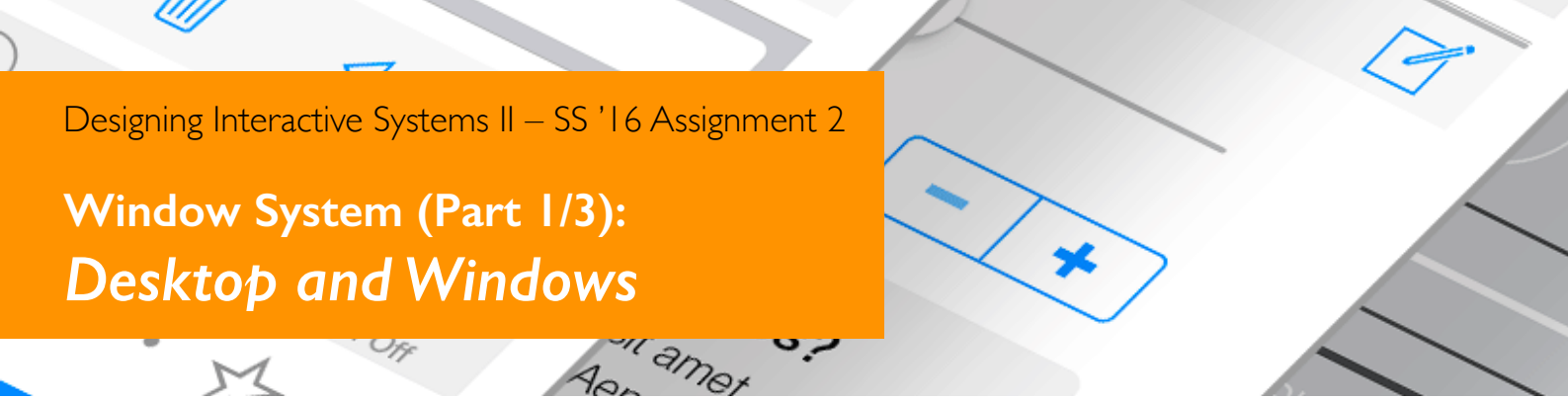

Designing Interactive Systems II – SS '16 Assignment 2

Window System (Part 1/3): *Desktop and Windows*

Due: Monday, May 2nd, 2016, **8:00 am**, in groups of 2 students via L2P.

Σ 10 points

Description
This is the first in a multi-week assignment where you will build your own window system and toolkit in Java. Please note that the next two assignments will build upon your code from this assignment.

Tasks

1. Desktop and Windows (4.5 points)

Your first task is to create the "desktop" for your window system. Write a `WindowSystem` class that puts up an empty desktop when created. You can subclass from the `GraphicsEventSystem.class` (included in the file `GES.jar` provided by this assignment), passing in the size of the desktop, which will create and draw the empty rectangle for you. The `GraphicsEventSystem` constructor accepts two int arguments for the width and height of the desktop.

- **`GraphicsEventSystem(int i, int j);`**

Extend your `WindowSystem` class to handle drawing in a resolution-independent way, using an "abstract coordinate system" with values between 0.0 and 1.0. To draw a line in the abstract coordinate system, implement a method

- **`void drawLine(float StartX, float StartY, float EndX, float EndY)`**

that accepts float values ranging from 0.0 to 1.0. You may find the following methods in `GraphicsEventSystem` useful:

- **`void handlePaint()`**
- **`void drawLine(int inStartX, int inStartY, int inEndX, int inEndY)`**
- **`void setColor(Color inColor)`**

You will need to override the `handlePaint()` in your `WindowSystem` class, and do all of your drawing in there. The `handlePaint()` method is inherited from `GraphicsEventSystem`, and currently does nothing.

Then, create a `SimpleWindow` class (the name "Window" is already taken by Java). Your `SimpleWindow` class doesn't need to have much functionality yet, but you will need to augment your `WindowSystem` class so that it has the ability to keep track of a collection of `SimpleWindow` objects.

Finally, create a test program, `MyApp`, that uses your `WindowSystem` class. Your test program should display a single line from (0.2, 0.3) to (0.8, 0.7) on your "desktop".

Hints and Tips:

- To access the `GraphicsEventLibrary.class` (in `GES.jar`), add the following line to your code:

```
import de.rwth.hci.Graphics.GraphicsEventSystem;
```

  To compile, you need to set the classpath to `GES.jar`, e.g.:

```
javac -cp ./GES/GES.jar ./Code/WindowSystem.java
```

  given that, in your current folder, the `GES.jar` is stored in the folder `GES` and `WindowSystem.java` is stored in the folder `Code`.

- Implement a private method that converts between the "desktop" and the "abstract coordinate system". Be sure that you preserve the aspect ratio of your desktop.
- You should also be concerned with how to organize your basic data structures – drawing and event handling will take place in an upcoming assignment.
- Do not use any Java Swing API's in any of your code. While all of the features that you will be implementing in your window system are already provided by Java Swing, the purpose of this assignment is for you to learn how window systems like Java Swing are implemented. Your code should depend only on `GraphicsEventSystem`. You may use any basic data structures that Java provides, such as Points, Arrays, etc., however. You may also import `java.awt.Color` for objects of type `Color` passed to `setColor()`.
- Do not decompile `GraphicsEventSystem`. We're not giving you the source code for this class precisely because your solution should not depend on how it is implemented.

2. Testing Your Understanding (2.5 points)

Answer the following questions:

1. How does your `WindowSystem` keep track of `SimpleWindows`? If you used a data structure, specify which one (e.g., array, linked list, hash table).
2. Justify your specific design and/or choice of data structures, in particular how it would affect the following:
   - adding/removing windows
   - drawing windows in front-to-back order
   - finding a specific window given an arbitrary (x, y) (desktop) coordinate
   - overall code complexity

Keep in mind that there is no single correct answer here. What we're more interested in seeing is whether or not you have gone through the thought process, not whether you have found the "perfect" solution.

3. Expert Question – for reaching a total grade of up to 1.0 (3 points)

For extra credit, write a (separate) Java Swing program (`ExtraCreditApp.java`) that contains a custom widget that extends `JComponent`. This widget should simply draw a horizontal line whose length can be adjusted by holding down the mouse button and dragging up (to make the line longer) or dragging down (to make the line shorter). The purpose of this exercise is to understand the relationship between event handling and drawing, which will help you in upcoming assignments. Describe what you learned for the design of your own event handling system. Oracle has some good documentation on this here:

http://docs.oracle.com/javase/tutorial/uiswing/painting/index.html

Submission

Use the following folder structure for your code:

<**groupname**> (the name of the root folder is your group number)

- |– **Code** (Put the files `WindowSystem.java`, `SimpleWindow.java`, and `MyApp.java` here.)
  - |– **Expert** (Put the java file(s) for the Expert Question including `ExtraCreditApp.java` here.)
- |– `README.pdf`

You do **not** need to submit the file `GES.jar`. Compress the folder `<groupname>` and submit the resulting **`groupname.zip`** in a group of two through the L2P classroom. Be sure to document your source code! An incomplete or missing documentation in the source code will lead to point deductions. Include a **`README.pdf`** (not more than 1 DIN A4 page, 12pt font size) that contains:

- names and matriculation numbers of the two group members in the header
- answers to the questions (see Testing Your Understanding)
- non-obvious things you did in your code (if any)
- anything that you think makes your design particularly optimized and/or elegant

Be sure that you use the folder structure and file names mentioned above. We will compile and run your code using a script. We will **not** start installing your favorite IDE or any other library you might want to use. Please note that assignments that do not meet the above submission criteria will not be graded. Be prepared to discuss your solution in the next lab. If you copy code without proper reference we will consider this as plagiarism.

Grading

The assignment will be graded on the following scale:

Points:	10	9	8	7	6	5	4	3	2	1	0
Grade:	1.0	1.3	1.7	2.0	2.3	2.7	3.0	3.3	3.7	4.0	5.0

Late assignments will *not* be graded.