CS 4620 PA 1: Eclipse Instructions

1 IDE Setup for CS 4620 Ray I Assignment

This is a guide aimed at people who are familiar with Java, but not familiar with software development. After reading this document, you will be able to use Eclipse to complete the programming assignments in this course.

If you are familiar with Eclipse and know how to import the project, please jump to the last section which describes how to run and test your code.

2 What is an IDE, and why do I need one?

IDE stands for integrated development environment. An IDE is a program that combines a source code editor, a compiler, and usually a debugger. You could solve all of the programming assignments in this course using nothing more than Windows Notepad and javac. But that would be painful. An IDE provides all sorts of nice features that a basic text editor, and the command line compiler, don't.

3 What IDE should I use?

There are several choices for a Java IDE. The course staff recommends Eclipse, and most of the support will be geared towards Eclipse. You are still welcome to use any other IDE that you prefer, but bear in mind that the TA's might not be familiar with IDE's other than Eclipse.

4 How do I get Eclipse?

Eclipse is available under a free, open-source license from http://www.eclipse.org/ You want the Eclipse SDK for the platform of your choice. It's a big file (120 MB for Windows), but is available by bittorrent. It is possible (even probable) that Eclipse is already installed on some of the lab machines, but you would need to check on a machine-by-machine basis.

5 How do I install Eclipse?

Extract the zip file that you downloaded somewhere. It should contain an eclipse directory. Inside that directory, there should be an eclipse executable. Run it. That's all there is! When it prompts you for a workspace directory, select a directory that does NOT contain the framework code.

6 OK, I'm running Eclipse. Now what?

Everything in Eclipse is centered around projects. A project basically corresponds to a related set of software. For example, you will probably setup a project for each programming assignment in this course. The first thing you will need to do is creating a new project. (Note: these instructions are based on Eclipse Helios. The current version is Indigo (3.7), and may be slightly different, but should be pretty much the same).

1. Go to File->Import...
2. Select General->Existing Projects into Workspace, hit Next.
3. Enter the directory where your copy of the assignment code is (you should have already downloaded the assignment code from the course webpage, and extracted the zipfile). Note that this is NOT the ray/ directory in the framework code zip file. It is whatever directory contains the ray/ directory. Note that this should not be on the hard drive of a lab machine – they are not backed up, and are subject to be wiped at any time. If you are using a lab machine, use your network drive to store your files.

```
/wherever-you-put-this-on-your-machine/    <--- project root folder
    ray/
        shader/
        surface/
        math/
        etc.
    TEST_A/
    etc. etc.
```

4. Hit Finish. The project will be imported under your workspace.

7 I have an Eclipse project setup. Now how do I use it to write code?

Another Eclipse notion that you need to understand besides projects is perspectives. A perspective is a way of laying out the frames in the Eclipse window, and what data those frames contain. The Java perspective is the one you will probably want to use.

The Window menu allows you to Open Perspective. There's also a button that does the same thing in the upper right part of the screen. Open the Java perspective if it isn't already open.

On the left side of the screen, you should see the Package Explorer (if you don't, go to Window->Open View->Package Explorer). This shows the tree of java packages in your project. Hit the ▷ sign next to an element to expand it. You should see files, which you can double click to open. Now you can start writing code.

The File->New menu will let you add a new class to your project. Follow the instructions in the dialog box – it's pretty self-explanatory if you are familiar with Java.

Play around with the menus, and right click on things. You'll find features that you like. Most of it is pretty intuitive. And the Eclipse Help files are extensive.

8 I've written some code, and I want to test it. How do I run my application?

First, you should have your "TEST_A" folder next to "ray", which includes the test scene files and solution images.

The main method of the assignment is located in RayTracer.java. Right click on that file in Package Explorer, and choose Run As->Java Application. Nothing will happen, since you need to provide the scene file as a command line argument. This can be done from the Run dialog (Run->Run Configurations...). Choose the Arguments tab, and provide your command line arguments in "Program arguments". For this project, the program expects the names of XML scene files as its arguments (e.g. "TEST_A\four-spheres.xml"). It will render to a PNG file of the same name, but with a ".png" extension. Once the arguments are set up, you can simply press Ctrl-F11 to run it. You can also dispatch multiple scenes to the program by writing the filename of each scene on a separate line in the command line argument. The program will process them one by one and write out each of the generated images.

Compare the images you have with the solution images, if there are visually significant differences, consider checking bugs in your code.

If you have any questions, post it on piazza, or come to office hours, or a help session, or contact the staff mailing list.