

## CS4815 Week05 Lab Exercise

**Lab Objective:** We will complete our introduction to OpenGL over the next two weeks by

- looking at the use of output primitives (`GL_LINE_STRIP`), etc.
- looking at a chart drawing program for plotting some silly data, and
- modifying its behaviour to react to menu and keyboard events

Here's a **quick summary** of the tasks:

- ❶ Copy the source code for this week's lab from the class directory `~cs4815/labs/week05`
- ❷ Take a look at a function for plotting circles efficiently
- ❸ Use the provided makefile to compile *two* source files in to *one* executable
- ❹ Run the program and be prepared to be un-Dazzled
- ❺ Modify the program so that it switches between chart-drawing modes
- ❻ Although there is no handin due this week what you do in the lab will be crucial for next week's handin.

### In Detail

❶ Following the procedure of previous weeks create a directory for this week's lab in `~/cs4815/labs/week05`. Now copy over from the class directory the two source files `chart.cc` and `circlePlot.cc`. These are to be found in `~cs4815/labs/week05`.

❷ One of the two files you have been provided with this week is called `circlePlot.cc` and it contains a function, `circleMidpoint()`, that takes a centre-point and a radius, and plots the corresponding circle. This is a very interesting method for plotting a circle as it takes huge advantage of the symmetry in plotting circles in order to speed up the rendering. This

is achieved in `circlePlotPoints()` where 8 different pixels can be coloured just from one known point. Note also the function `setPixel()` which “colours” a pixel.

③ The source code you are being given this week is C++ , although only barely: a very simple class that defines a point in the  $xy$ -plane makes it C++ as opposed to C. We will use `.cc` to denote C++ files and the compiler we will use for compiling C++ programs is `g++` . If you look at the `Makefile` you will see where we specify `CXX=g++` .

What is also new about this week’s set-up is that the executable comes in two parts: the main bulk of it is in a file called `chart.cc` and this is the one you should do all of your work on.

In terms of getting the program to be runnable you will need to compile together the two source files to create an executable, `chart`. A way to do this is with the command:

```
g++ chart.cc circlePlot.cc -l GL -l GLU -l glut -o chart
```

This takes all of the source code in both files, syntax checks them, generates machine code for them and searches in the `glut` library for “anything that wasn’t supplied” in the two source files. From the point of program development the purist would say that this generates unnecessary work: since we will never change `circlePlot.cc` why should we repeatedly have to syntax check it and generate machine code? The solution here is to syntax check it and generate the machine code (called compilation) once and save it in *object* form; this is achieved with the `-c` flag to the compiler and results in a `.o` file being generated (`.obj` in the Devil’s tongue). As we change `chart.cc` we *will* need to compile it each time. Then we *link* the two object files together to create the executable. This is summarised in these three steps:

```
g++ -c circlePlot.cc
```

```
g++ -c chart.cc
```

```
g++ chart.o circlePlot.o -l GL -l GLU -l glut -o chart
```

The first command is done once only, the other two are done any time you make a change to `chart.cc`.

While all of the above works very well we can now begin to see the power of a properly set up `makefile`. If we modify, say, one of the `.cc` files we need to recompile it and link the `.o` version to recreate the executable. Alternatively we just type `make` and the `makefile` manages the process for us.

④ Run the program and note how the bar chart is not fully displayed in the window. Will

resizing <sup>1</sup> the window fix this problem? Whether it does or not, this is not a proper solution. Can you fix this? Hint: don't do anything until you have read and edited/rerun the program as per the next item.

⑤ You will see from the code in `chart.cc` that there are 2+ chart-drawing routines. There are functions `barChart()`, `pieChart()` and an empty (stub) `lineChart()`. Comment out the call to `barChart()` and replace it with the call to `pieChart()`. Remake and rerun. You should have the same data presented without labels this time. Try resizing the window. Why is the circle becoming elliptical? Can you fix this? Hint: the `glViewport();` function does the job. See `glut07.cc` of last week for its use.

The second and final task for today is to set up a menu system so that the type of chart can be selected from a menu; the data should be redrawn in the selected format on exiting the menu. Refer to last week's sample code for how to tackle this. As an add-on, add a keyboard handling callback so that hitting 'c' or 'b' will toggle between the two styles of chart, also. You should add a `quit()` function that can be used by the menu, or the keyboard, 'q'.

---

<sup>1</sup>Clicking on the bottom right corner and dragging.