

Computer Graphics I - 2D Clipping Assignment

In this assignment, you will implement the Sutherland-Hodgman Clipping Algorithm as presented in the video lectures.

Goals:

- Improve your understanding of 2D clipping
- Complete your suite of 2D drawing algorithms that can be included as part of your midterm.

2D Programming Environment

The programming environment that you will use for this assignment, like the previous 2 assignments, is a set of simple object-oriented classes with implementations in both Java and C++. You are free to use either of the implementations. The classes include:

- `simpleCanvas` – a simple 2D canvas that allows the ability to set a pixel.
- `extenderCanvas` - a special subclass of `simpleCanvas` with functionality for testing out the clipping assignment. Note, this class should only be used for clipping assignment and only for testing purposes!!!
- `clipper` – a simple class that performs clipping operations.
- `clipTest` – the main function for the application.

The only class that you will need to modify is the `clipper` class. For this assignment, you will need to complete the method `clipPolygon()`, implementing the Sutherland-Hodgman Clipping Algorithm. The signature for the `clipPolygon` method is:

```
int clipPolygon(int in, float inx[], float iny[], float outx[],
               float outy[], float x0, float y0, float x1, float
               y1)
```

Where `inx` and `iny` are arrays holding the coords of the vertices of the polygon before clipping. `n` is the number of vertices in this polygon. `outx` and `outy` are arrays that will hold the coords of the vertices of the polygon after clipping. The function should return the number of vertices in the polygon after clipping. You are free to add additional members and methods to the `clipper` class as you see fit, and you may also include additional auxiliary classes (though this is not required). However, you cannot modify the `simpleCanvas`, `extendedCanvas` or `clipTest` classes.

Additional Libraries

The Java implementation of the programming environment makes use of the standard `awt` and `Java2D` classes to access the 2D drawing areas, so no additional libraries or Java files, other than the ones provided by the assignment are required.

However, since C++ does not have a standard 2D graphics environment, you will need some additional libraries

if you choose to use the C++ implementation for your submission. The C++ implementation is built on the SFML (Simple and Fast Multimedia Library), a free, cross-platform and lightweight library for performing common multimedia tasks. It can be downloaded at <http://www.sfm-dev.org/download.php>. Version 1.6 of the libraries are used. There are precompiled binaries for Windows, Mac, and Linux. (NOTE: For Windows, there are precompiled binaries for Visual Studio 2005, and Visual Studio 2008. If you are using Visual Studio 2010, I have precompiled a version of the libraries for this versin of VS, and placed them in the CONTENT area of mycourses.)

Please be sure to refer to the installation instructions for your given platform:

- Windows (Visual Studio) - <http://www.sfm-dev.org/tutorials/1.6/start-vc.php>
- Mac OS X -- <http://www.sfm-dev.org/tutorials/1.6/start-osx.php>
- Linux -- <http://www.sfm-dev.org/tutorials/1.6/start-linux.php>

You can find the files for the assignment [here](#). Note that the distribution contains folders for both the C++ and Java versions of the programming environment.

Submission

Your submission should only include the files for the `clipper` class that you have modified and any other auxillary classaes that you have created. For the Java version, this would mean the file `clipper.java` and for the C++ version, this means the files `clipper.cpp` and `clipper.h`. PLEASE DO NOT SUBMIT code for `simpleCanvas`, `extendedCanvas` or `clipTest` as you should not be changing the code for theses classes.

This assignment is due on Wednesday, January 10, 2013 at 11:59pm. Submissions should be made to the dropbox labeled [2D Viewing and Clipping](#). Note that this is after the break.

Grading

This assignment is worth 10 points distributed as follows:

- 5.1 points for submitting **something** that compiles, runs, and produces a window.
- 0.7 points for each clipped polygin in the test program. There are 7 such polygons.

By the way, the correct output for the test program should look like this:

