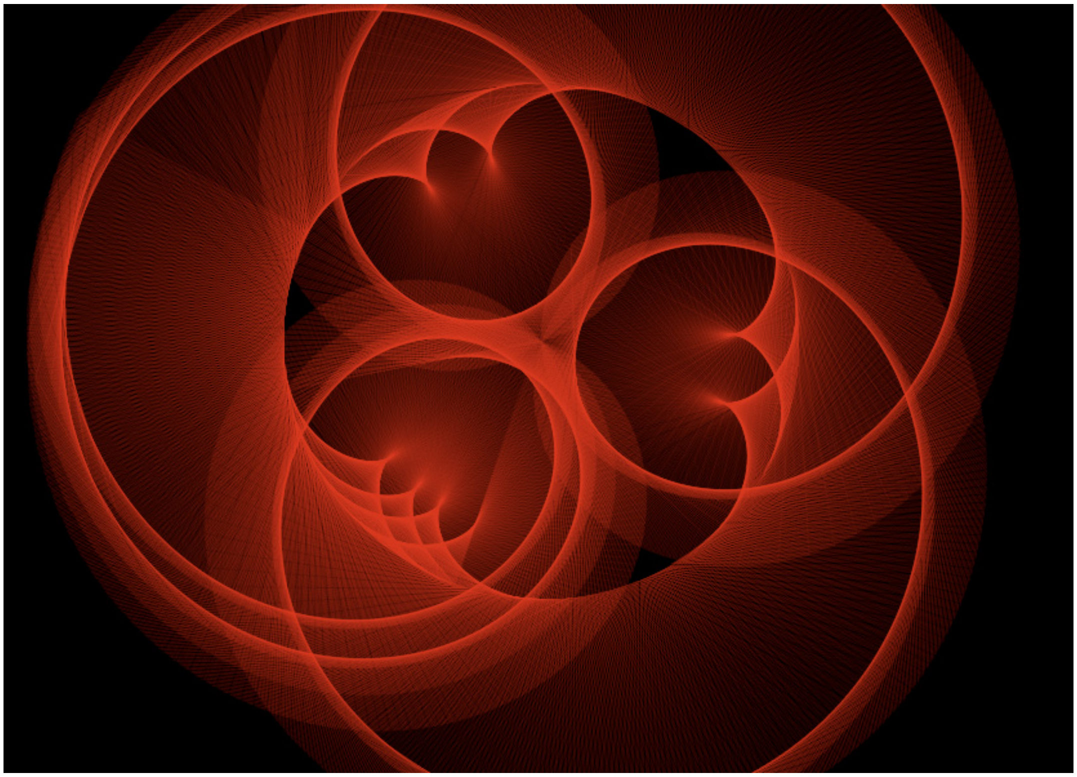
Assignment 1: Generative Art

CS 148 Autumn 2013-2014

Due Date: Monday, September 30th, 2013 at 7pm



Introduction This assignment involves creating a simple OpenGL (Open Graphics Library) application which will give you the opportunity to familiarize yourself with the basics of OpenGL and GLUT (The OpenGL Utility Toolkit). OpenGL is a powerful API (application programming interface) for rendering 2D and 3D computer graphics. Most modern GPUs (graphical processing units) implement the OpenGL interface which subsequently allows the programmer to take advantage of hardware accelerated rendering.

Background Before starting to write your program, you will need to decide what you want to draw using OpenGL. Here are some ideas (and links to examples and definitions in blue) to get you started.

- Spirographs or similar functions (note the ”SEE ALSO” pages linked at the bottom)
- Fractals, drawn to a certain level, see e.g. the Koch Snowflake
- Simple space filling curves
- Simple L-systems, e.g. use them to create simple trees (lines for branches, triangles for leaves)
- Conway’s game of life

- A simple arcade game - Pong, Pacman, Snake...
- XScreenSaver has many good examples
- If you are feeling adventurous, try something with animation and/or particle effects, add physics, and/or try adding some user input. These linked examples were done in JavaScript, but you could easily make something similar with OpenGL and some creativity!.

Implementation

1. Decide which desktop or mobile operating system you will be using to implement the assignment. Example code is available for Windows (Visual Studio), Mac OS X (Xcode, Makefile), or Linux (Makefile). No example code is provided for mobile operating systems (Android, iOS, Tizen, etc.).

2. If you are using a desktop operating system, download the skeleton code and, if necessary, setup GLUT with your development environment. If you have a newer Mac, try opening the Xcode project first as GLUT may be preinstalled. Compile the code and run it. The skeleton code contains a very simple GLUT based OpenGL program that creates a window and draws a blank screen.

3. If you are using a mobile platform, you can either create your own program or port the skeleton code to your mobile device. You will also need to open a window using a method appropriate to your own device (instead of using GLUT).

4. Read up on OpenGL and GLUT and answer the following questions. You will be expected to know the answers to these questions when you present your project to a CA for grading.

    - How is control passed back to the programmer in a GLUT program? How is this set up during initialization?
    - What does the parameter to `glutInitDisplayMode()` in `main()` specify?
    - What do the calls to `glOrtho()` and `glViewport()` in the `reshape()` function accomplish? If the window is to be resized, why might we want to change this?

5. Extend the program to draw something (anything!). Your completed program must meet or exceed the following three requirements.

    (a) Draw using at least two of the OpenGL primitives
    (b) Draw using multiple colors
    (c) Interact with the user to do something within the program. This can be as simple as using the number keys to select one of multiple modes.

Grading This assignment will be graded on the following requirements

- The resulting drawing is sufficiently complex.
- The program draws using at least two of the OpenGL primitives and multiple colors.
- The program incorporates some form of user interaction.
- The student is able to adequetely answer questions asked by the CAs.

according to the following rubric.

- + – Exceeds the requirements via one or more artistic/technical contributions
- ✓ – Meets all of the requirements
- − – Does not meet the requirements but still produces a drawing.
- 0 – The submitted solution does not produce a drawing.

FAQs

- I've used SSH or VNC to connect to a `myth` computer and compiled the code successfully. However, when I try to run it I get the following error.

  ```
  freeglut (./assignment1): OpenGL GLX extension not supported by display ':1.0'.
  ```

  - When using an X based system, you need the GLX extension to be present to run OpenGL programs. The VNC server installed on the `myth` computers doesn't support this. If you have an X server installed locally, you can forward X (including GLX) through your SSH session by adding the `-Y` option to your `ssh` command. On Mac, you can use their X11 server. You just need to install it, run it, and then `ssh` using the `-Y` option. Then, any X program you run remotely will display locally. On Windows you need to install an X server. You can obtain a free one through Cygwin. See Installing Cygwin/X for instructions on installing and using it. Note you'll have to install the version of the server with GLX support (labeled experimental, but seems to work well). You may also have to fiddle with the `startx` script to get it to start the correct X server. Needless to say, this can be a lot of trouble. We encourage you to develop on your personal computer if possible.

- I'm using Visual Studio and Nate Robbins' GLUT and the compiler is saying '`_exit`' is being redefined. How do I fix this?

  - This is due to the fact that GLUT hasn't been updated to work properly with newer versions of Visual Studio. The simplest workaround is to include `stdlib.h` before including `glut.h`.

- In Xcode, I get "`GCC 4.2 is not compatible with Mac OS X 10.4`". How do I fix this?

  - Try changing the Active SDK to from 10.4 to 10.5 or 10.6 in the Overview dropdown.