CS 370: Computer Graphics Final Project: Fireworks

Craig Earley, Johnathan Hicks, Edward Ly, Eli Ramthun

December 11, 2016

Building and running the program

We have provided a Makefile. Run 'make' on the home server for the Earlham Computer Science Department, or a similarly-configured Unix environment to make all necessary binaries. Run it using '`./firework`' on the command line or double-clicking in a GUI.

Project design

This project adapts the fireworks class provided in a link by the instructor to a click-based interactive graphical tool. Implementing this formed the code base, which was relatively straightforward.

After that was done we participated in the process of iterative software development: create a version that works, add a feature, add another feature, and so on. This program was easiest to design this way, as design and implementation questions both came up and were resolved as we worked through the current version of the project.

File structure

We use the following open-source code as the basis for the firework class:

- `firework.h`, `firework.cpp` (which compiles into `firework.o`)

We wrote the following code to use that class:

- `demo.cpp`: compiled against the `firework.o` object, becomes the firework binary

We also have a Makefile and a `README` which we used as a group reference for the specifications of the project. For full edit history, please see the Earlham CS Gitlab repo `graphics/fireworks`.

Completed and future work

We completed a program that displays fireworks in a window with a random color scheme, a fixed acceleration, and a particular pattern. We then expanded it to include a menu for selecting the size of the particles in the firework, then another to change the colors of the firework.

The program needs more user control in the future. Acceleration and pattern could also be controllable by a user. This would allow a range of creativity in operating the program. In particular we imagine more submenus and potentially keyboard actions to affect firework choices, but an on-screen panel would be better-looking and more intuitive.

Roles of group members

Most of this project was a group effort. In particular, we did a substantial amount of pair-programming, which contributed to relatively quick development of the project.

As individuals, our roles were as follows (alphabetical by last name):

- Craig Earley opened the git repo, procured the open-source code, and wrote this report. Because of illness, he played a smaller role in development in the middle of the project.

- Johnathan Hicks contributed to development of the software at all stages and to producing the presentation.

- Edward Ly wrote the much of the main body of code and led the in-class demo.

- Eli Ramthun contributed to the development of the software at all stages and wrote code for the menu and its submenus.

Thus we produced the fireworks.