CS 211 Lab 8

Asteroids

Winter 2021

Today is an exercise in software archaeology. You will be given a complete program—your goal is to understand it well enough to modify and improve it.

Lab setup

Project setup

For C++ projects, including this lab, the starter code is provided as a ZIP file for you to download: https://nu-cs211.github.io/cs211-files/lab/lab08.zip. Extract the archive file into a directory in the location of your choosing. Once you have your new directory containing the starter files, you can open it in CLion.

Be careful, as CLion will only work correctly if you open the *main project directory* (which has the the CMakeLists.txt in it). If you open any other directory, CLion will create a CMakeLists.txt for you, but it won't work properly.

Getting oriented

Your first step in dealing with a new codebase is to gain a basic understanding of the structure and how it fits together. In particular, you need to understand what significant types (classes, structs, or enum classes) are defined by the code, and how they're related. Are any derived classes of any others (the *is-a* relation)? Do any contain instances of or references to instances of any other (the *has-a* relation)? Of particular interest in this assignment is the inheritance hierarchy, and how it produces the behaviors of different kinds of objects.

Possible goals

Here are some ways you could try modifying the program:

- Instead of ending the game after dying once, keep a life count and give the player multiple chances.
- Add a game-over screen that allows the player to try again.
- Make objects wrap around at the screen edge instead of colliding with it.

Like classic Asteroids.

- Add power-ups, such as an extra life or faster spaceship.
- Rate-limit the firing mechanism to allow *n* shots before requiring an *r* second reloading delay.

Power-ups should spawn randomly and disappear when acquired.