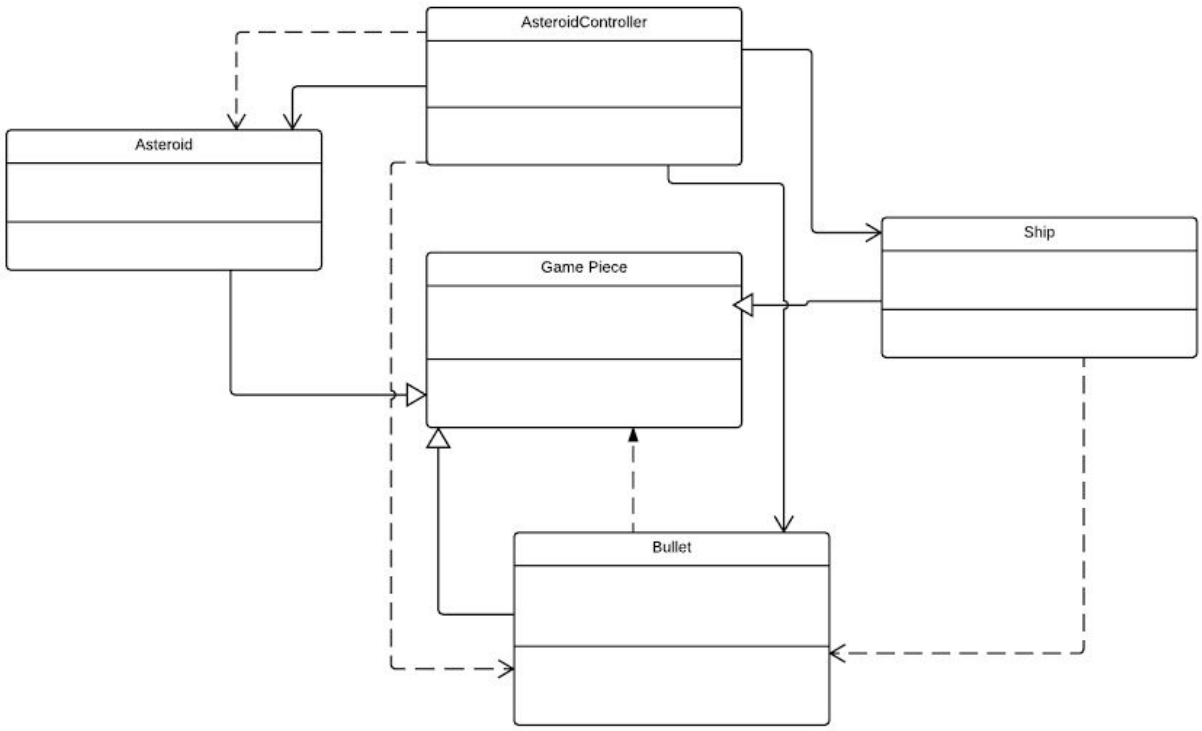
User Manual

Our project was a contemporary overhaul of the classic game Asteroids from 1979. In the game, the player controls a spaceship and must avoid asteroids of varying sizes and speeds by dodging and shooting them. Shooting an asteroid grants points toward the user's score, and destroys the asteroid. If it the asteroid was not the smallest size it can be, it splits into two asteroids of the next smaller size. The game ends when the ship is destroyed by an asteroid, which happens when they collide.

There are many problems we had to face and address while working on this project, and although what follows is mostly the order that we solved them in, a lot of things didn't become perfect until the end of the development. The first and most basic user story was to generate an actual window when the program is launched and place a ship inside that window. Then, we needed to add functionality to the ship, which includes both moving and shooting. The moving part was deceptively complicated, for the movement system in the original game is very unique. The user can rotate the ship, only move forwards, and there are both acceleration and friction present; the ship speeds up, up to a certain point, when the move key is held down, and then slows down when rotating or the move key is released. Also pertaining to the ship was its ability to shoot a bullet that would travel across the screen with a constant velocity, only stopping when it had traveled a certain distance or collided with an asteroid, and to implement a super move, which would destroy all asteroids on screen when used. Next, we needed to have asteroids be added to the screen in random places around the border with random velocities and sizes (from a list of three possible sizes), and have them split upon destruction if they weren't the smallest type

of asteroid. And last but not least, we needed to create a way to constantly check for collisions between the ship, bullets, and asteroids, which proved to be the most difficult problem to solve because of how we created the GUI.



The most important classes in our program are GamePiece and AsteroidController. The GamePiece class is the parent class of the Ship, Asteroid, and Bullet classes, which are all of the different objects that make up the game. They all have very similar instance variables and methods, so it made sense to have them all have the same, large parent class. Their differences are encapsulated within the child classes. The AsteroidController handles button presses as well as the most important aspects of the game. It generates asteroids in the correct places with the correct frequency, it loops through the lists of objects to check for collisions, and it updates the

position of the images onscreen. We go into further detail explaining these classes and their relationships in the design manual, but that's the gist of it.

In order to use the program, the user must first launch it, which will then cause a main menu to appear. If the user would like to view the controls, they can select the "controls" option, which will then bring up a controls menu. The only other options besides that are to either quit, which will kill the program, or play, which will start the game. While playing, the user moves forward by pressing and holding the up arrow key, rotates left and right by pressing and holding the left and right arrow keys respectively, and shoots by pressing the "w" key. If the user wishes to use the super power-up once they have a high enough score, they press the "e" key. Once the game is over, the user can either go back to the main menu or quit the game.