Asteroids Game Report

Yifei Wang

From the process of programming the Asteroids Game. I have gained a lot more experience with OO programming and Java game development. It was not easy to get started at the first time. I spent some time to review other examples of how to build up a game interface. The GUI display was a big slot of programming. Then I realized the relationship between several critical functions. The main function will only call the game loop function, which is called the game play in my functions. This is the important function to keep refreshing the interface because the function while loop that keeps running to display the game. The game loop calls the gameRender and paintScreen functions. Those tow together makes a double buffer to display the screen without a flick. The gameRender uses g2d methods to paint on the graph then the paintScreen paints on the screen. After these two functions, there follows a set of other update functions.

In the game loop there are certain functions to be set up in order to keep the interface display the game which are those update functions . Building the game loop and update functions were very important to have a correct game display. Then I reviewed the notes and drew on paper to display the class structure. This is a very helpful stage while setting up the entire structure of the program. Update functions instructs how the objects will react in the game. For example, the update player ship function will allow the location of player ships to be looped inside the game screen. Easy implementing is just to add of minus the width and length of the screen. Similar to asteroids, alien ship, and small asteroids. Updating bullets got a little bit different. I have two bullets list, one is from player ships and another from the alien ship. The function calCol helps to check if the bullets shot something that should be effected. While looping through the bullet list, if any objects of asteroid, small asteroid or alien ship got shot, the bullet and the object vanish. While checking if the alien ship has a smaller bigger range of getting shot, since the alien ship is a little bit bigger than the asteroids.

After having a basic structure of the program, adding stuff became a little more smooth. I added alien ship after I have my player ship class. Alien ship was not very different than the player ship, except it fires by itself which means the bullets do add up by time rather than adding up by keyboard listener. Also the way it dies is similar to player ship, three shots get the alien ship to die. So I wrote in the alien ship class to have a HP equals to 3. Every time player shot at the alien ship, the HP lost by 1. When Hp is 0, game render functions display no more alien ship.

During the programming, I found out about how OO programming helped to reduce work and implement objects more efficiently. Taking the sound as an example, all the audio is played by implement the SimpleSoundPlayer class. This really reduced work when there are many audio sources are needed in the game. With SimpleSoundPlayer class, the line needed is to new the object in the GUI class. And call the play method of the object. Moreover, all my basic game stuff classes are derived from a base call AstBase. This is another important feature of OO programming. Inherit and Derive. Since all the player ships, asteroids, small asteroids, and bullets have the common fields like x, y location, speed, heading and isAlive. Using inheritance avoids write those common lines again and makes the class structure clear. While using the object, it is clear to refer to the fields and update.

What was hard to implement at the first place was the GUI. After getting the idea of draw the screen and update it was not a problem. Other stuff that took time I faced was writing the update functions. It involves a lot of testing and adjustment. The bullet update includes the players' bullets and alien ship's bullets. Update functions was not difficult if the structure is clear but took time to test and figure out the bug. OO concepts also help with debugging. Since you know which object is giving a unexpected behavior, it is easier to focus on which part of program to fix.

if anything, could have been changed in the project to make it better I think if playing the game once in class and explain here is the alien ship and this is the rogue ship will help understand which object is which at the first place. Anyhow, this is already an instructional idea to implement a Java game and by programming the game this introduced the concept of developing game with OO language to me which is very valuable for the future.