

## 30862 Write Up

I have learned many things from making this game. When I first learned OO programming, I felt that PO programming is much better. I thought that PO is very simple and straightforward. You just need to program what you want in order. Why build that many meaningless objects to do the same thing? But after making this game, I found that OO programming has many advantages. It is more organized, gives the game a nice view. Objects can have their own attributes and function, so it is very comfortable when making things such as spaceships, bullets, etc. Also inheritance is awesome, objects that have similar attributes can be inherited from one base class. This makes the game even more organized. It reduces the amount of code as well. Through making this game, I mastered on what should be implemented as an object, and how to use them in game. Actually I also found that many things in Java are actually objects, such as String, Integer and even System. The only bad thing about OO is that it is slow. But for making a game, speed is not a big deal!

When I just started making the game, I found that I had no idea about how to make a graphical game. It is really hard to think of if you have no experience on it. Then I studied the code on kleene's website and found the way to do it: you need objects for any movable things on the screen. The objects need to store their own status like speed, position, color, etc.

And to make them visible on the screen, you need a “draw” function that be called in gameloop. And this “draw” function will draw object at their own positions. So actually everything you see on screen is merely a drawing, not a “real thing”. There also need to be an “update” function that updates the status of all objects. This “update” function will call every object’s own “update” function.

The second thing I found to be hard was Save & Load. You need to analyze what are needed to be written to file. These things must be sufficient to restore the game. When load, you need to parse out all information, and use them to restore the game. You need to store and load everything with absolutely no mistakes. Because when you load a game, the game must resume playing as if you didn’t save the game (i.e bullets remain their flying directions, etc). If you miss one single attribute, the loaded game can be wrong or even errors can be thrown.

At last, what can I do better if I have time? First I want to fix save and load dialogs, I don’t know why but they are a little bit awkward. I will also optimize the collision detection. Java’s built-in collision detection uses the inner rectangle of a polygon, which is very imprecise. And last but not lease, I will improve my UI even more! Even if my UI is very cool already!