Criterion C: Development

Techniques used to create Asteroids.

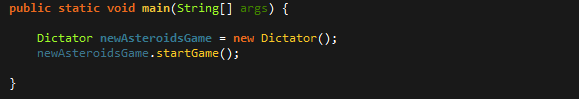
* Game Loop – Page 1, 4
* Graphics and Displaying – Page 6-7

Rest of code is featured in Appendix Classes

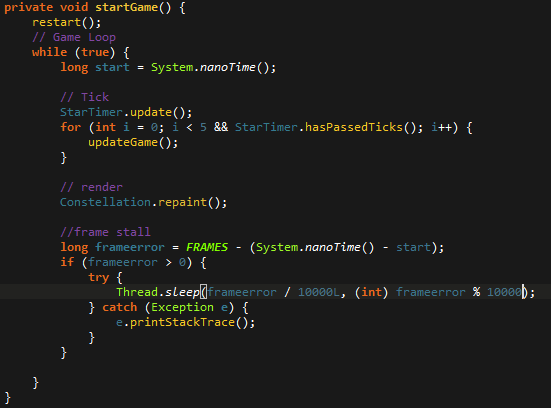
**Game Loop**

For many smaller games, it is easy to use a method called a game loop. This loop usually is a while loop that runs continuously while the application is running. The reason for this type of iterative approach for managing the application is because it allows easy separation between rendering the game, and computation on the game.

The Game Loop starts when the main calls startGame() on a dictator object. The main method is the method run when the application jar is called, however, since it is static, and all the benefits of java come from object oriented computation, not much happens in the main.

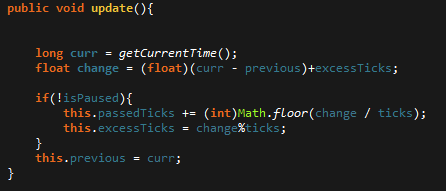
*Illustration 1: Main Method*

As shown in the main, the .startGame() method is called onto a new Dictator object. The startgame method serves to both initiate the game, and start the Game Loop

.

*Illustration 2: Game Loop Method*

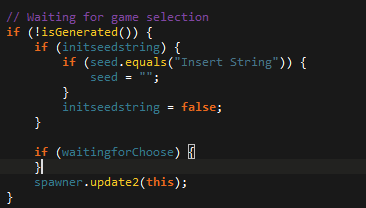
The restart() method serves to instantiate the rest of the variables the Dictator class must have e.g. all the menu booleans. This method is kept not part of the constructor because when a restart is called onto the game, instead of remaking a dictator object again, it just runs restart().



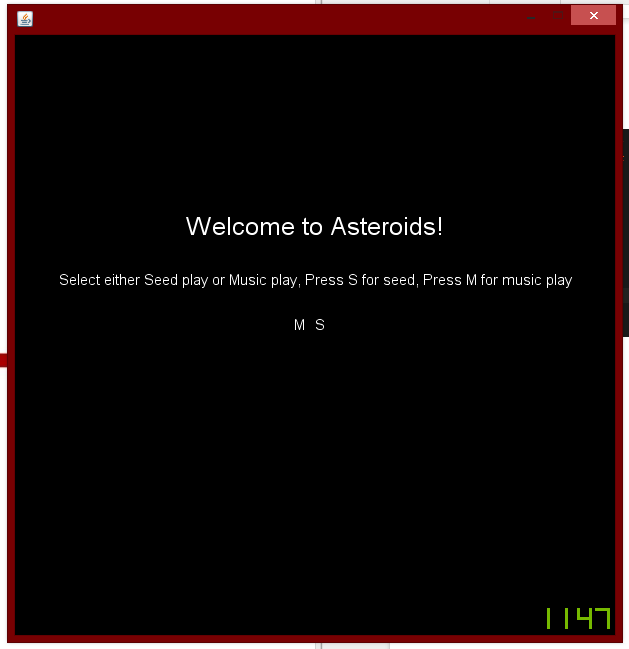
*Illustration 3: Update Method in Watch method*

Next we see the game loop start, shown by the while(true) loop. The for loop stalls the game slightly so that the updateGame() or the “tick” method isn't called to fast to often. The Startimer object is a watch object. When update is called, the time on the StarTimer .is updated

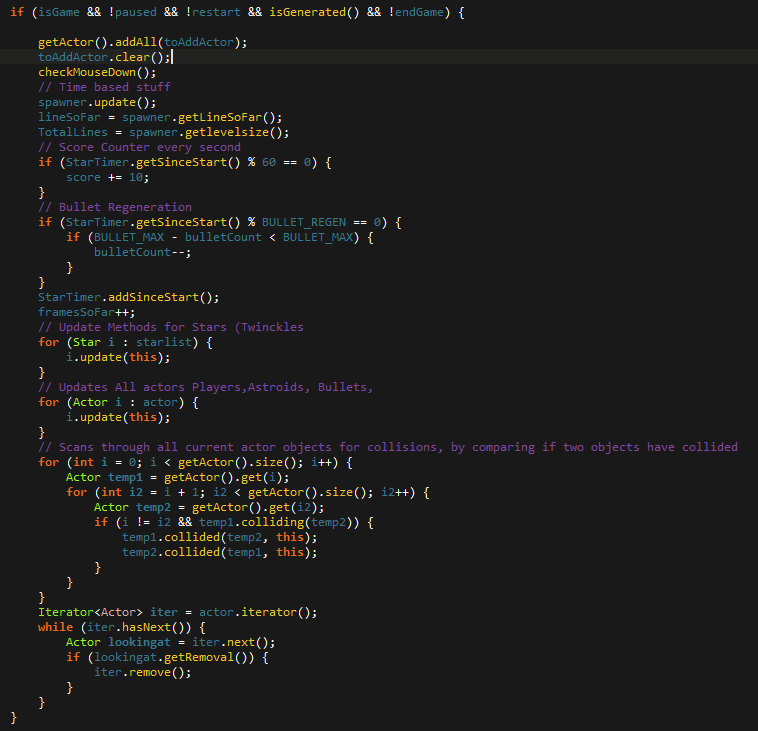
After the tick check, comes the updateGame() method. The update method has two main parts to it's functionality. The pre game, and during game segments



*Illustration 4: Not in game loop, in update()*



*Illustration 5: Not in game mode example*

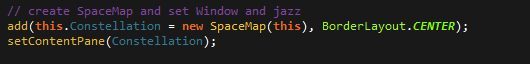
The next part is the during game-play loop. *Illustration 6: GameLoop during gamemode*



*Illustration 7: Game Play Example*

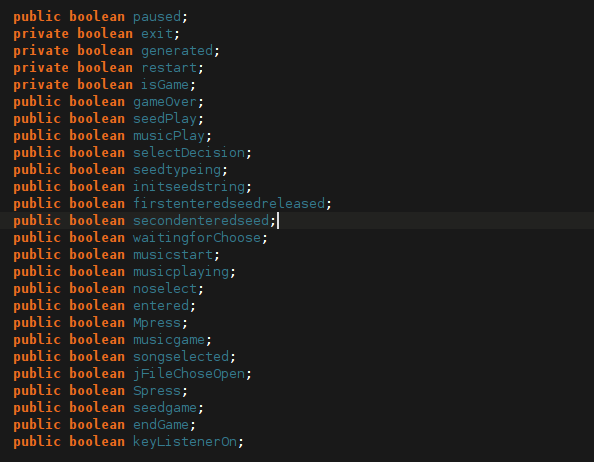
**Graphics**

For graphics, I made a separate class called SpaceMap, which is created in the constructor of the Dictator class.

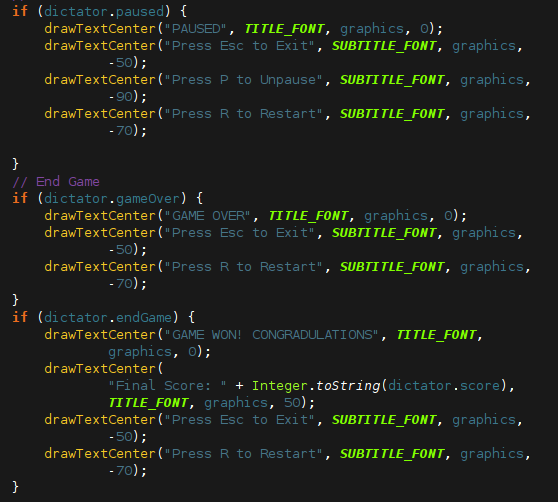


*Illustration 8: Makes a Constellation/spacemap object, then sets it to the contentPane*

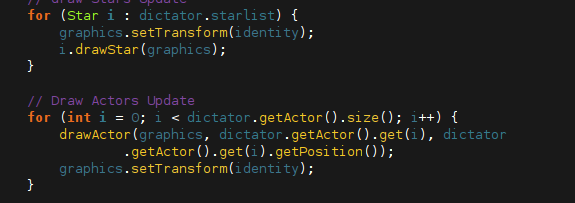
The space map extends Jpanel, another default java class that deals with graphics. Space map, similar to the updateGame() method, has different boolean checkers to decide what to display on the screen. This way, there's separation between rendering a menu, all the game entities, or the pause menu.



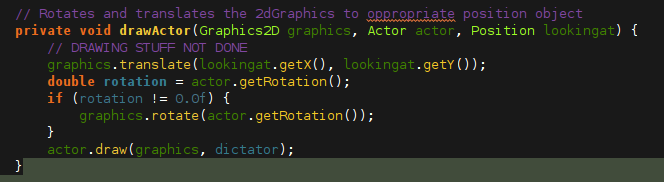
*Illustration 9: Booleans for Menu Graphics*



*Illustration 10: If statement checkers for Graphics*



*Illustration 11: Rendering in Space Map class*



*Illustration 12: drawActor class*