OceanExplorer	- OceanMap	Ship	Pirate
Pane root	<pre>int[][] oceanGrid = new int[25][25]</pre>	Point currentLocation	Point currentLocation
 Main() void start(Stage oceanStage) throws Exception void loadShipImage() void loadPirateImage() void drawMap() void startSailing() 	 public OceanMap(int dimens, int numIslands) void drawMap(ObservableList <node> root, int scale)</node> public void placeIslands(ObservableList st<node> root, int scale, int x, int y)</node> 	 public Ship(OceanMap oceanMap) public Point getShipLocation() void goEast() void goWest() void goNorth() void goSouth() 	 public Pirate(OceanMap) public Point getPirateLocation() void move() public void update(Observable o, Object arg)
<u></u>			
	drawMap	getShipLocation	getPirateLocation
	void drawMap(ObservableList <node> root, int scale</node>	<pre>public Point getShipLocation()</pre>	public Point getPirateLocation()

Design Decisions

For Homework 4, I followed the guided and given methods and functions. The only difference between the given code snippets and my implementation was the use of the islandMap() method. Instead, everything for the islandMap method was included in the drawMap class, drawing all of the ocean spots, pirate ships, and regular ships together. As for drawing the pirate ships, the images were loaded in a function in the OceanExplorer() class, but the pirate ship activity was implemented in its own class. I did this because first I attempted to load the image in a function in the OceanExplorer class but was experiencing issues with getting the image to load and move properly. So, a Pirate class was created to load the image the same way as the Ship class and its movement was modified. I couldn't completely figure out how to properly get the pirates to move, but I learned from the Observer method from the cat and mouse example to better indicate following. Additionally, I allowed for the pirates to move diagonally in the case that those spaces were available.