Nicolas C. Broeking & Josh Rahm
CSCI 4448
April 8, 2014

Project Part 3: Progress Report

Summary

Nic Broeking has implemented the classes surrounding application events. The game main function will run a GameApplication that inherits from RiskApplication. The game application has a main loop that currently asks the CLEventDelegate to get a command. The CLEventDelegate gets the command and gives it to a CommandHandler that inherits from Handler. The CommandHandler will create an Event and the main loop will end if it is a Quit Event.

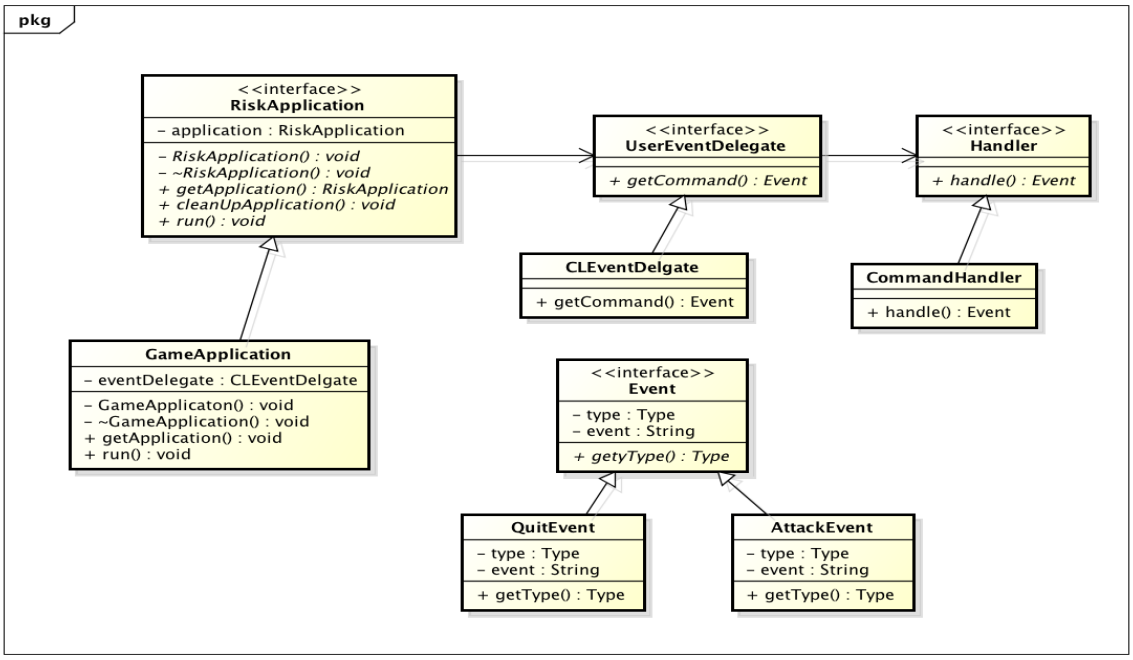
Josh Rahm has written the classes that abstract the ability to read and write data from a file descriptor in the form of generic types asynchronously to better abstract the IO between the client and the server.

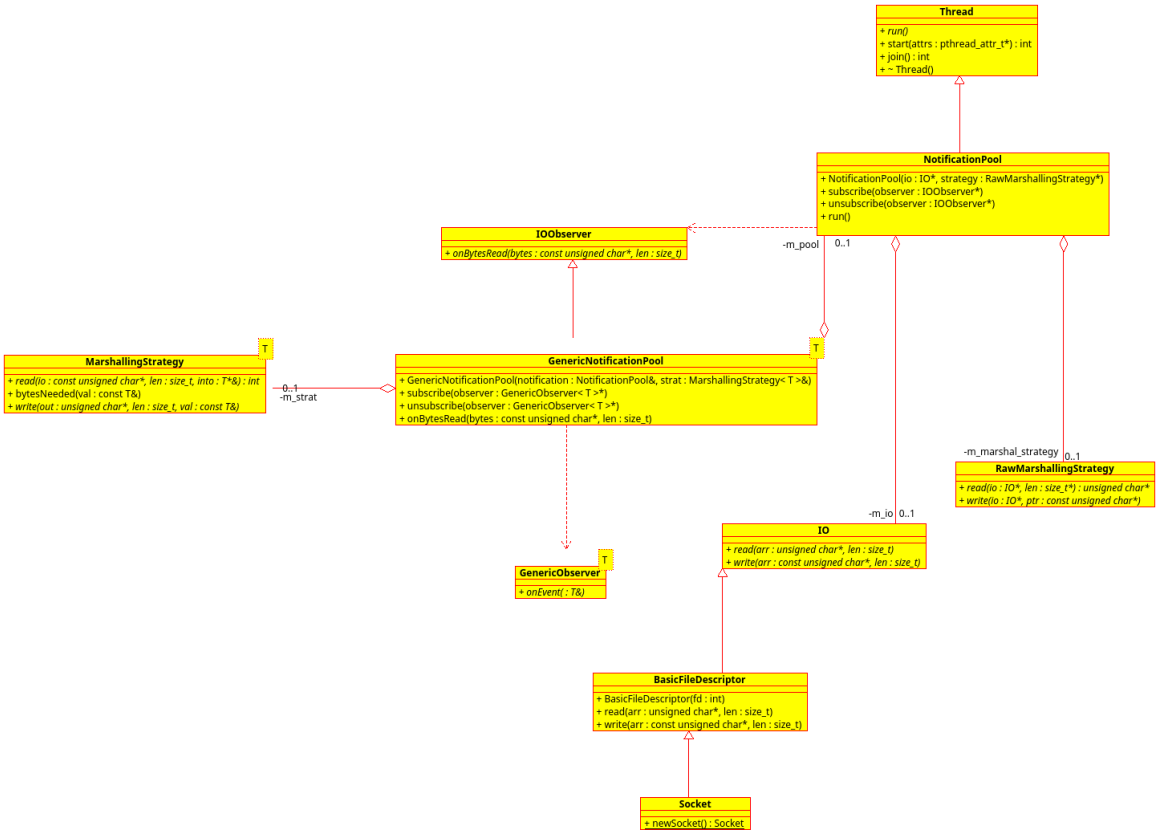
We still have quite a bit of work to do. We have to create the entire game state and make the server side communicate with the client side. Our design has changed a little bit because we realized what classes that we don't need and we realized other classes that we do need. We have also realized that we will not be able to get a full graphical interface up and running so we are going to stick with the command line interface risk game.

So far we have used the strategy design pattern, the observer pattern, and the singleton pattern. They are helping to make the code cleaner so we are able to develop our application efficiently. Using these patterns from the start has allowed us to create very concise and cohesive code that we won't need to refactor later.

Class Diagrams

Client Side:





Plans For Final Iteration:

For the next iteration we want a command line interface to be able to attack two countries to play risk. This includes being able to reliably communicate with the server to ensure the integrity between the game states.