Christian Collier, Qingyue Li, Mark Ozdemir

COSC-150

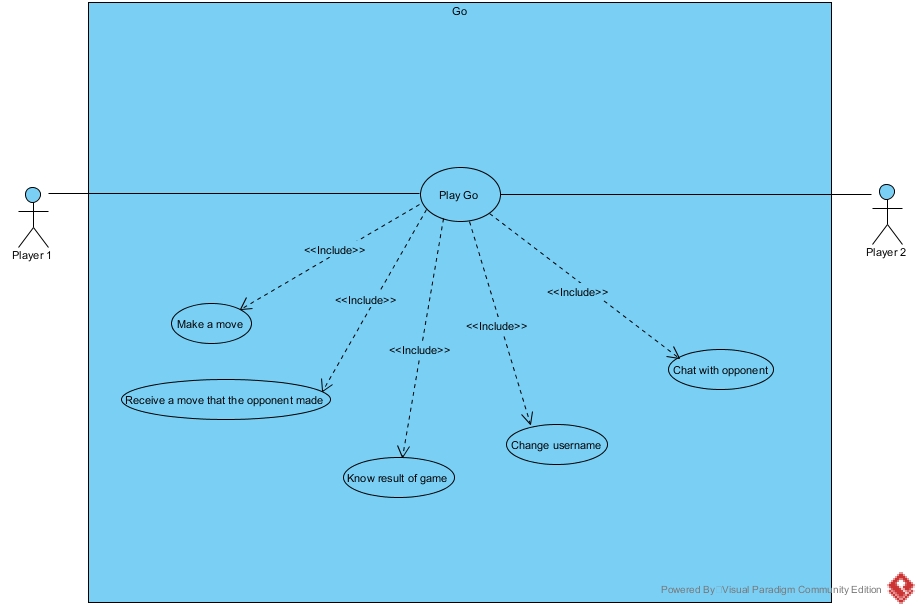
12/5/17

Go Team 1 User Report

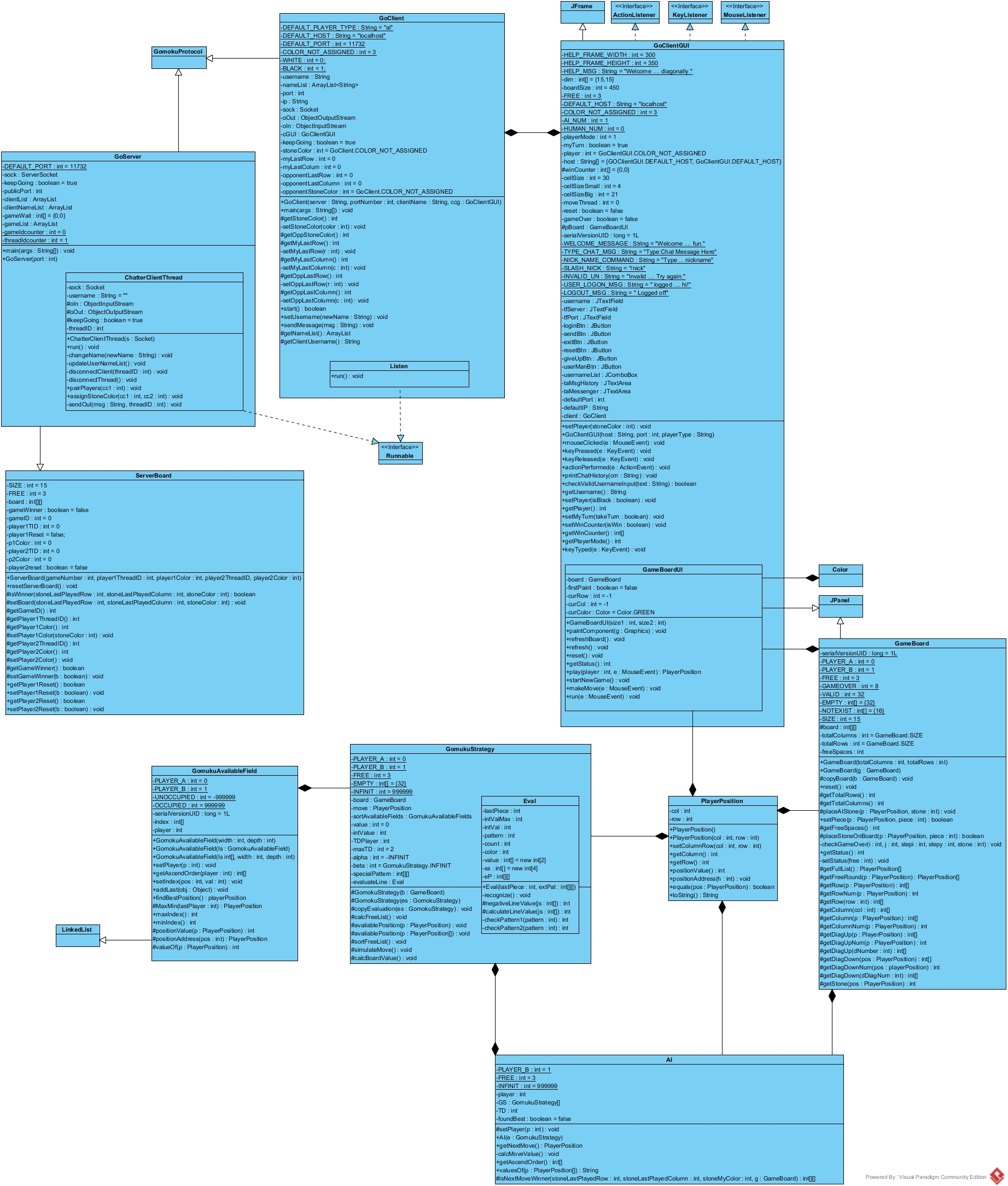
**Introduction:**

Welcome to the Go User Report for Team 1. In this report, we will show our UML use case diagram, class diagram and sequence diagrams for our program of Go. We will then discuss our project timeline and milestones as well as task allocations. Finally, we will discuss what we have learned through this project.

**UML Use Case Diagram:**

****

**UML Class Diagram:**

****

**UML Sequence Diagrams:**

**Project Timeline and Task Allocations:**

The following are dates that we planned to have certain milestones done by at the latest:

11/14/17:

* Have preliminary UML designs for meeting with Prof. Grace – Christian

11/21/17:

* Finalize UML diagrams for feedback – Mark, Kathryn, Christian
* Compile code from previous projects for use – Mark, Kathryn, Christian

11/28/17:

* Complete GUI design – Mark
* Work on server connections – Kathryn
* Compile research for AI – Christian

12/2/17:

* Preliminary PowerPoint – Christian
* Preliminary AI – Mark
* Fix server-client connections – Kathryn

12/4/17:

* Finalize PowerPoint – Christian
* Finalize user report – Christian
* Finalize user manual – Kathryn
* Finalize feedback from UML diagrams – Kathryn
* Streamline code format/structure – Mark, Christian
* Check and finalize connections with other computers – Mark

**What We Learned:**

Through the Go project, we learned three primary concepts that were essential to the completion of this program and gave us insight into developing a better team.

Principally, we learned how to properly reuse and modify code that was used on previous projects in this class. One of the more difficult aspects when first presented with this project was how many working parts there were that we had to implement. Instead of recoding our project from scratch, particularly with the server and client connections, we were able to reuse our Chatroom project code to build the server and client connections. Despite some differences between the two programs, we were able to modify our code effectively from that previous project so that it fit into the goals for this project. For example, we implemented a serializable class called ChatterMessage in our previous project, which sent messages in object streams instead of just through strings. Due to this project’s standard protocols, we adjusted our code such that we removed the ChatterMessage class and only sent strings as messages between users and from the server to its clients.

Secondly, we learned how artificial intelligence can be used for online games and what the basics are for human vs. AI or AI vs. AI programs. Obviously, our main concern for this project was developing the AI algorithm that effectively wins a game of Go as close to every time as possible. Through the class PowerPoints and some independent research on algorithms such as Minimax and alpha-beta pruning, we better understand how more complex games are able to use these algorithms to their advantage.

Finally, this project taught each of us how to effectively manage and allocate a large-scale programming project such as this one. With many different working parts, we were constantly told by Prof. Grace to start early and to plan accordingly. Fortunately, we had already had a beneficial and effective team strategy from the previous projects and it was clear what each of us would be able to specialize in. This project, however, reinforced those ideas and forced us to further understand and manage the various aspects of the program. Moreover, we had to balance our task allocation of the program with the other requirements for the project, such as the user manual, user report, etc. Each of the other components required a significant comprehension of the program we had written as well as how to effectively articulate how our code works. Therefore, this project also has taught us how to communicate our findings and express our ideas behind the lines of code that were written.