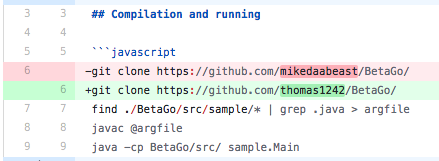
**Architecture Essay  
“BetaGo”**

**History and Context of the Project**

“BetaGo”[[1]](#footnote-1) is a small open source Java project based off the ancient Chinese board game – “Go”. It was created by GitHub users “thomas1242” and “crougraff” and is designed to be played via an interactive GUI, making it a very fluid and easy game to run and play. BetaGo is able to be played as a 2 player game, or also as a human vs computer experience, and has a very basic AI implementation where the computer simply chooses a position at random. Go known to be a particularly difficult game to create an accurate AI for - in fact, AlphaGo, a program developed by Google has only beaten a top-ranking Go player a handful of times[[2]](#footnote-2), so it is likely that the creators of BetaGo took on the project to test their game building skills, rather than to create a world-class AI.

BetaGo is a fairly recently created, and therefore does not have a large amount of history behind it. It seems to be a collaboration between two individuals who started the project for personal practice outside of work or study. The initial commit was on the 1st March 2018, and since then there have been only 34 other commits, appearing every so often, hinting that this is indeed a personal project, rather than it being work or study related - in which the commits would be much more regular.

Looking at all the commits of the GitHub project repository, it is apparent that this project is building off another project by another user, “mikedaabeast”. Their README.md has been edited delete mikedaabeast’s cloning instructions and has replaced them with their own (see opposite), as well as large dumps of tests and source code just appearing in the space of 1 day. The commit messages, assertions, and deletions make it clear that this new version of BetaGo aims to improve and add functionalities to the mikedaabeast’s project, rather than refactoring it to make it architecturally better.

Looking at all the commits of the GitHub project repository, there has been next to no large code redesign and all the work is centred around building the implementation of the project, rather than improving the design. It is clear that the programmers (both of the current version of BetaGo and the previous one) wanted to incorporate the Model View Control (MVC) design pattern from the start, and built their code around this model, rather than writing it, then remodelling it to fit the pattern later.

* 1 March 2018: Initial commit, cloned previous programmer’s BetaGo implementation
* 2 March 2018: Improved the visual design of the GUI, refactored the Board code (so much so, that it hints that the previous implementation by mikedaabeast did not work)
* 5 March 2018: Refactoring win screen
* 6-12 March 2018: Visual redesign of CSS
* 23 March 2018: Added code to calculate scoring
* 26 March – 1 April 2018: Added a difficulty slider to set the level of difficulty the player will be playing at
* 18 - 24 April 2018: Added a Monte Carlo Tree Search (MCTS) algorithm to calculate the best move to take by the AI, updated the GUI.

Since the goal of the project has always been to create a digital version of Go, the overall purpose of the project has not shifted from its original purpose and is unlikely to do so since the project is fairly un-extensible and isn’t meant to fulfil multiple purposes. However, a foreseeable future for the project is extending it into a project that allows the user to play a variety of traditional games such as Go, Chess, etc.

Platform:

Java IDE

Something that has a jvm installed

JavaFX

Junit5

1. https://github.com/thomas1242/BetaGo [↑](#footnote-ref-1)
2. http://fortune.com/2016/03/12/googles-go-computer-vs-human/ [↑](#footnote-ref-2)