

# CS 463 - Spring 2017 - Midterm Progress Report

## Project DevAI

Jacob Broderick, Kristen Patterson, Brandon Chatham

## I. RECAP

## II. WHAT WE HAVE

## III. WHAT IS LEFT

### A. *Stretch Goals*

1) *Base Expansion*: The only primary objective we have left in development that has not been completed and passed testing is the expansion but which will be touched out more late.

2) *Other Stretch Goals*: Fortunately, the other stretch goals we started have been finished. There may be one small addition made to add the ability for scouts to attack the opponent when it finds them, but other than that, stretch goals are being wrapped up and in some cases, being put through regression testing.

### B. *AI Module*

In order to show all of our development in action, or at least the interesting parts, we need to design our AI module to have a scripted strategy it takes. This will not be difficult because most of the code is already in the AI module only it is commented out for unit testing purposes. Once we build the script the way we like, we will film it and create our pitch for the product.

### C. *The Pitch*

Our pitch will largely be focused on how effective a resource this could be for students to learn about AI and machine learning algorithms. AI and machine learning overlap a bit conceptually but they are often thrown around as buzzwords in the industry. This project will give students the ability to see the steps behind scraping data, analyzing the data, and presenting it to data analysis algorithms for dynamic AI decisions and that is what our pitch will focus on.

## IV. PROBLEMS

### A. *Expansion Bug*

Our largest issue to-date is the expansion function bug. While we have the function written similarly to other functions that build structures, the primary difference is the location it chooses to build. The location of an expansion is very important because you want it as close to the resource deposits as possible. As we have mentioned in previous reports, our code finds the correct base location and is able to send units to that position, however, when we try to construct a base there, the function behaves not as expected. At first, we did not think this would be a persisting bug. It is very unfortunate that it has been stubborn as it was one of our bigger talking points for our library and pretty fun to watch visually. We will continue to try to fix this until expo and we have reached out to developers of the BWTA API for assistance on what we might be doing wrong. Hopefully they will be able to find a small hiccup in our code.

### B. *Deep Mind*

While this is not a problem in the general sense of the word, our project has been focused on abstracting the difficulties of using BWAPI to develop rudimentary AI modules. Deep Mind has announced it will be developing a Starcraft API that will likely be vastly better than BWAPI. Additionally, Deep Mind is partnering with Blizzard, the creators of Starcraft to

make this the industry standard of AI libraries for Starcraft and potentially all real time strategy games. So while this is not directly a problem for our project, it does significantly devalue it as Deep Mind resources will dwarf anything we have put together. Although, this project is not exactly cutting edge anyways.

### *C. New Starcraft Version*

Because of the preparation for Deep Mind development, Blizzard has released a new patch. This may cause direct problems for our code because we are not sure the new version will be backwards compatible with BWAPI. As of now, we have not tried it out because we have found resources online say it is difficult to return to the older version once you have updated. For now, we plan to stay on our current version until expo is completed and our project is done being evaluated. Then we can take the chance to see if BWAPI still works.