CS 463 - Spring 2017 - Final Report

Project DevAI

Jacob Broderick, Kristen Patterson, Brandon Chatham

Abstract

This document details the progress made by the Starcraft AI Project. It includes a brief recap of the project, where the project currently is, and what stretch goals remain unfinished. It also talks about any problems that have occurred. This document was made with a video that details similar information, but demonstrates what the working project looks like.

Contents

I. Introduction

This project was requested by our client Dr. Fern. Dr. Fern's primary intent for this project was for us to reduce the learning curve of artificial intelligence and machine learning implementations for an AI club that will soon be created at OSU. Learning AI can be difficult but the basics can be picked up quickly. Some of the trouble spots for new developers who are not well-versed in the process of starteing projects might get hung up on configuring API's and setting up their development environment. Our project handles the configuration step for them. The environment is ready to use as soon as they open the Visual Studios project. Furthermore, if anything gets changed in a developer's settings, we have thorough documentation explaining how to fix it. Beyond this, we have included tools to help more skilled developers start data mining for machine learning algorithms with a python script we wrote and a replay analyzer we researched and chose from an open source developer. There are three team members: Brandon Chatham, Kristen Patterson, and Jacob Broderick. Brandon primarily focused on code design, implementations, and overall direction of the project to best fit the desires of the client. Kristen focused on documentation. Nearly all of the documentation, even a small bit of the comments you will find in the code, are written by Kristen. Kristen worked closely with Brandon to keep up-to-date on what the newest implementations were in order to keep documentation relevant to the current state of the project. Jacob focused mostly on choosing our development tools in Visual Studios and BWAPI as well as researched other tools that may be valuable to us. BWAPI is the standard API used for Starcraft AI and was an easy choice. Determining what other tools were well-suited to the specifications of the project were slightly more difficult but fairly simple decisions. Our client for the most part gave us autonomy on the project. While we did meet with him for required check-in's for the project, he allowed us to make our own decisions whether this was for better or for worse. We did consult him on a few occasions but usually made decisions primarily based on what the team felt was best based on what we knew Dr. Fern was most interested in.

II. Original Requirements Document

III. Changes from Original Requirements Document

A. Final Gantt Chart

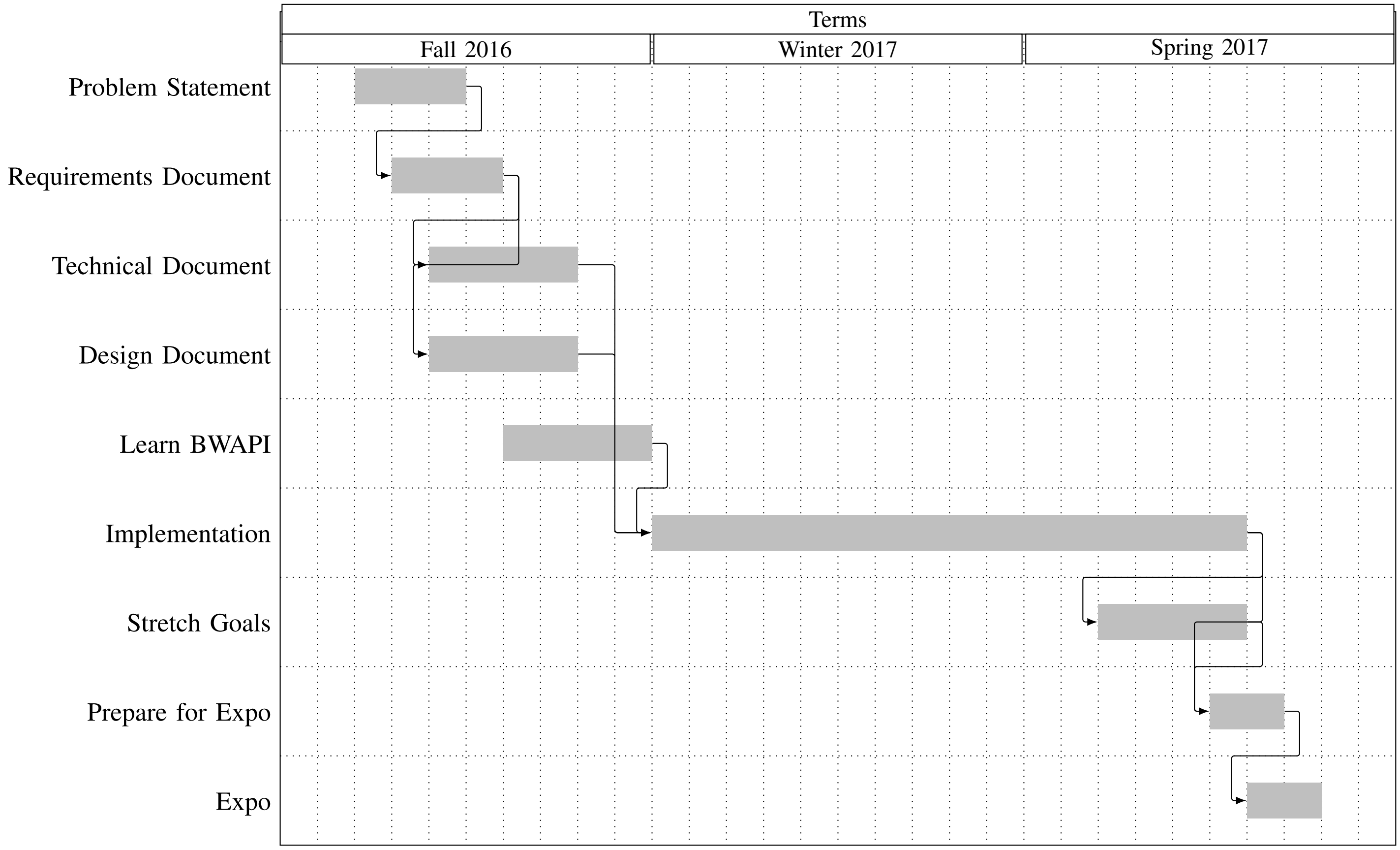

Fig. 1. Final Gantt Chart

IV. Design Document

V. Tech Review

VI. Weekly Blog Posts

A. *Kristen Patterson*

1) Fall 3: Last week I had clarified the exact details of the project with my group mates and also contributed to the problem statement by revising the abstract, writing the problem definition, and writing both the introduction and some of the conclusion.
Planning with teammates about the requirements document and discussing problem statement with client.
Had some clarification issues with some details in the problem statement. Client was also on vacation so could not verify whether the document fit his criteria early enough.

2) Fall 4: Last week our group had met with our client and had him look over our problem statement and provide feedback. I also corrected some grammatical and formatting errors with my section of the document. I also corrected the document based on the client's feedback.
Next week we need to get our client to sign our final draft of the problem statement. We also are going to have a rough draft of a requirements document by Wednesday to have ample time to set up a meeting to discuss with the client.
We had a problem when scheduling our meeting with the client as not all the group members were able to attend. I also had trouble figuring out why apostrophes would not show up properly on our tex document.

3) Fall 5: Last week I had completed the introduction, formatting, gantt chart, and abstract of the requirements document.
Next week we will meet with our client and discuss the requirements document as well as revise any issues we have with our document.
We had trouble meeting with our client to get a signature as he was very busy. I also had some time management problems as it was a very busy week.

4) Fall 6: Reviewed SRS document with client and made changes based on input.
Discuss with team about following documents and clarify materials needed with client.
Our team was fairly busy this week so there was a decent amount of time management problems.

5) Fall 7: Got the final draft of the SRS document signed and submitted as well as clarified the technology review with our teachers and TA. Also broke our project down into 9 parts and split up the project.
Finish my portion of the tech review then make correction and meet with client to make sure it is acceptable.
I had time management problems as I had other assignments due. There was also some clarification issues on what our 9 parts of the tech review should be.

6) Fall 8: Finished our Tech Review and submitted it. Discussed course of action for design document and progress reports.
Discuss format of final progress report and make any revisions to the tech review.
Mainly just time management problems again as this has been a busy term. Also need to clarify some questions for upcoming documents.

7) Fall 9: Received revisions for Tech Review and discussed changes to them as well as assigned roles for the design document.

Complete my section of design document as well as my progress report and presentation.

No problems this week.

8) Fall 10: Completed my section for the Design Document and Progress Report.

N/A

I have encountered some family issues that produced some problems in my schedule.

9) Winter 1: Set up the environment to work on the project.

Start gathering modules for and practice using BWAPI.

Trouble getting a new schedule to meet with our team.

10) Winter 2: Discussed timeline for project and started economy portion of code.

Continue working on economy portion.

Had a lot of other assignments that made it difficult to have more time for this project.

11) Winter 3: Economy and Construction portion of code is finished.

Work on unit production and control.

More time issues and sickness as well.

12) Winter 4: Started functions for training and selecting marines.

Start function for moving and attacking.

Just more time issues.

13) Winter 5: Finished my portion of revisions and progress report and started code for alpha video.

Finish alpha video code and presentation.

I had 3 midterms so I had less time to focus on the project than I wanted.

14) Winter 6: Finished all the revisions and progress report

Participate in elevator speech practice and start documentation of code

Time management

15) Winter 7: Practiced elevator pitch for expo.

Start Documentation

Didn’t get anything done as I was busy on other assignments.

16) Spring 1: Wrote up documentation for scraper and scouting class.
Come up with visuals for the second draft of our poster.
Set up of new schedule with team.

17) Spring 2: Discussed poster draft 2 with Kirsten for feedback and corrections and made revisions. Also found good visuals to use in poster.
Meet with group for group photo and meet and discuss poster with our client.
Was unsure of the best type of visuals to use for the poster.

18) Spring 3: Completed poster draft 2 and discussed meeting time with client. Also completed flowchart for visual on poster.
Meet with client to discuss final poster and make corrections.
Had trouble trying to come up with visuals for the poster and the general structure but I ended up going to Kirsten and she helped.

19) Spring 4: Discussed poster with client and made final corrections to code as well as discussed deliverables with our TA.
Make final corrections to the poster and submit it before the deadline.
We had some trouble trying to set up a time to meet with our client but because we had contacted him early we had plenty of leeway before the deadline.

20) Spring 5: Completed Wired interview.
Prepare for Expo.
Had trouble trying to set up an appointment for the interview.

21) Spring 6: Completed What we have so far portion of the Progress Report and the project details portion of the presentation.
Prepare for Expo.
Had trouble splitting up the work for the group.

22) Spring 7: Completed attacking function in program and prepared video for expo as well as participated in Expo.
Class sweep up.
Tried to finish up everything quickly for expo.

23) Final Post: If you were to redo the project from Fall term, what would you tell yourself?
I believe one of the biggest issues my team faced was that we got caught up in some of the more non-essential parts of the project and we didn't focus on completing the project before working on stretch goals and interesting mechanics.

What’s the biggest skill you’ve learned?

The biggest skill I have learned during this process is how to be able to talk about a project on multiple different levels of complexity.

What skills do you see yourself using in the future?

The skills I have learned on how to explain my project and how to properly work in a team and manage my time appropriately.

What did you like about the project, and what did you not?

I liked the concept and the coding of the project. It just felt like the schedule was catered more towards larger and more complex projects and I would have like to have more time focusing on development rather than extraneous planning.

What did you learn from your teammates?

My teammates taught me a lot of technical skills that they were used to using and I believe I also helped them with it as well.

If you were the client for this project, would you be satisfied with the work done?

I believe we provided a good starting point for students to use to supplement their research and learning.

If your project were to be continued next year, what do you think needs to be working on?

The actual artificial intelligence portion of the code would be the focus as our project’s main purpose was to supplement the development of that process.

Speak a little about your expo experience.

I spoke with many individuals ranging from industry reps to people who did not really know what a video game was.

VII. Final Poster

VIII. Project Documentation

A. Project Walkthrough

The project is sectioned off into many classes for a library described below as well as an AI module. This AI module is the main location where the interaction with the game occurs. It has a startup phase where it converts the play map into data for the AI to use. After the initial startup phase the AI starts a nested for loop where it goes through each individual unit and gives it command. The worker is in charge of gathering materials, constructing buildings, and scouting. The Command Center is in charge of training workers. The barracks are in charge of training military. The marines are in charge of moving to a choke point and attacking the enemy. It does this loop each frame until a victor is decided. A victor is decided when on of the players buildings and enemies are all destroyed.

B. Installation

In order to install this project all that is needed are the supplementary programs described below and cloning from our repo.

C. Running Project

After everything is downloaded and code has been implemented it is required to compile and import the data into the game. In order to compile the game you need to go into ExampleAIModule in the Project folder. Then you need to run ExampleAIModule.vcxproj. When the project loads up change the solution configuration to Release not Debug In the Solution Explorer window, right click ExampleAIModule and select Build and check to see it was built correctly. Go back to the Project folder and go into the Release folder. Here you will find ExampleAIModule.dll, copy it. Go to where StarCraft Brood War is installed and find or create a folder called bwapi-data and inside it find or create a folder called AI. Inside the AI folder paste the copied file ExampleAIModule.dll and start up ChaosLauncher found in BWAPI. Make sure to run ChaosLauncher as administrator. In ChaosLauncher make sure BWAPI Injector(1.16.1)RELEASE is checked and highlighted then select config. In the config menu make sure the path for ai goes to the path where you pasted the .dll file. Now go back to the ChaosLauncher and click Start then create a custom game and it should be set.

D. Requirements

In order to install the project and correctly use it it is necessary to download other supporting software as well. It is also preferable to be working on a device running at least Windows 7. Below is a list of the required programs to be installed:

StarCraft Brood War
Must be updated to version 1.16.1
Visual Studio Community 2013
Can also be Visual Studio Community 2015 with 2013 compiler tools
BWAPI 4.1.2
BWTA 1.7.1

Important! Blizzard has taken down their installer for the original game without any patches and replaced it with version

1.18 included which does NOT support this API. As of late, there is no official installer that can be used to get version 1.16.1.

E. API

1) ResourceGathering: Purpose:
The purpose of the ResourceGathering class is to start the game and have functions pertaining to the economy of the game. The class contains basic functions and variables to keep the gathering of minerals and gasses efficient.

Functions:
buildWorker
Input
Command Center
Processes
If the Command Center is idle and fails to construct a worker more supply will be built. Also, if supply is within 4 of the maximum supply, the AI has enough minerals, and it is more than 3 minutes into the game, then more supply will be built.
Output
Either a worker is made, a supply depot is made, or nothing happens.

workerGather
Input
Command Center
Processes
Uses type of Command Center to determine race. Selects a worker to perform construction of supply structure.
Output
Worker is selected to be able to perform processes.

gatherGas
Input
Worker
Processes
Commands the worker to gather gas.
Output
Returns true if the worker is correctly directed to gather gas and false otherwise.

gatherMinerals
Input
Worker
Processes

Commands the worker to gather minerals.

Output

Returns true if the worker is correctly directed to gather minerals and false otherwise.

getMineralCount

Input

None

Processes

Gets in-game amount of minerals.

Output

Count of minerals.

getGasCount

Input

None

Processes

Get in-game amount of gas.

Output

Count of gas.

Variables:

The only variables for the class is optimum gatherers for both minerals and gas and the current gatherers for minerals and gas.

2) Building Construction: Purpose:

The purpose of this class is to build any buildings necessary to play StarCraft. Any other buildings can be included in this class following this same pattern.

Functions:

buildCenter

Input

Command Center, Building location(optional), player flag

Processes

Uses type of Command Center to determine race. Selects a worker to perform construction of an expansion.

Output

Command Center is built.

buildSupply

Input

Command Center, player flag

Processes

Uses type of Command Center to determine race. Selects a worker to perform construction of supply structure.

Output

Supply depot is built

buildGas

Input

Command Center, player flag

Processes

Uses type of Command Center to determine race. Selects a worker to perform construction of gas structure.

Output

Gas structure is built.

buildBarracks

Input

Command Center, player flag

Processes

Build Terran Barracks if the input is of Terran type.

Output

Barracks is built.

buildGateway

Input

Command Center, player flag

Processes

Build Protoss Gateway if the input is of Protoss type.

Output

Gateway is built.

buildSpawningPool

Input

Command Center, player flag

Processes

Build Zerg Spawning Pool if the input is of Zerg type.

Output

Spawning Pool is built.

checkConstructionStarted

Input

Player flag

Processes

Check construction flags and switch them off accordingly.

Output

Construction status is checked.

Variables:

None

3) UnitAction: Purpose:

The purpose of this class is to be able to control units individually to perform actions. This includes buildings as well as other movable units.

Functions:

checkUnitState

Input

Unit whose state is being validated.

Processes

Checks the state of the input unit.

Output

True if unit is not dead, being constructed, disabled

trainMarines

Input

Barracks to train marines, player flag

Processes

Starts process to train marines.

Output

Marines are trained.

selectArmy

Input

None

Processes

Selects all marines.

Output

Marines are store in a queue.

Variables:

None

4) PlayerInfo: Purpose:

The purpose of this class is to store variables to keep track of the game state that would otherwise be a global variable.

Functions:

adjustMineralOffset

Input

player flag

Processes

Adjusts the mineral offset to store the process of building until it is actually built.

Output

Minerals are adjusted.

adjustGasOffset

Input

player flag

Processes

Adjusts the gas offset to store the process of building until it is actually built.

Output

Gas is adjusted.

resetMinerals

Input

player flag

Processes

Reset the mineral offset to zero.

Output

Offset is reset.

resetGas

Input

player flag

Processes

Reset the mineral offset to zero.

Output

Offset is reset.

Variables:

Flags to store what buildings are being built, counts of our current barracks and expansions, and offsets in the resource counts for planned buildings.

5) MapTools: Purpose:

The purpose of this class is to convert the map being played into data for the AI to interpret and use.

Functions:

getNextExpansion

Input

None

Processes

Finds next resource location to build an expansion.

Output

The tile position to build the base.

getAbsoluteTileDistance

Input

Tiles at starting and ending locations.

Processes

Calculates absolute distance not considering terrain.

Output

Distance

Variables:

None

6) ScoutManager: Purpose:

The purpose of this class is to send out units to scout and be have units attack enemies that are seen.

Functions:

scoutStartLocations

Input

Unit that will be sent to scout.

Processes

Checks all possible base starting locations starting with the furthest away from your starting base.

Output

Reference to the scout.

attackTarget

Input

Unit that was sent to scout and the target to attack

Processes

Attacks the target with the scouting unit.

Output

Unit attacks target.

Variables:

Flag to see if the unit is scouting and the unit that is scouting.

Below is a website providing a tutorial and documentation for BWAPI:

http://www.teamliquid.net/blogs/485544-intro-to-scbw-ai-development

https://bwapi.github.io/annotated.html

IX. New Technology

This project helped us familiarize ourselves with new API's and the process of developing a library for other developers. Usually in an academic setting, no one else is using your code. This gave us a new perspective. We were not writing code to just solve a problem, but instead helping solve a problem in an easy to understand way as well as design it such that it may be useful to solve even greater problems. We had to think of what a developer might want to do with our API before we even considered how we would write wrappers for the other API's. The primary resource used was: http://www.starcraftai.com/wiki/Main_Page. This web page encapsulated nearly everything the team needed with links to all of the tools we used. Additionally, when choosing our development tools and language, we consulted this page because it provides information on different technologies to use when interacting with Starcraft AI. From here, the team was able to determine what best fit the needs of the project and made it easy to know what tools were available that should be considered as the project progressed. The team did not consult many books on this topic. For the most part, the team used a small amount of previous knowledge from AI coursework and leveraged online sources. The same goes for on-campus resources. Only one or two on-campus resources were used sparingly if at all.

X. What was learned

A. Brandon Chatham

I mostly familiarized myself with machine learning, data mining, and the process of developing a library. There was a great deal of trial and error getting the tools configured and ready to use. This caused me to reach out to developers of both API's we used more than once. I gained a better idea of what it actually means to create industry level code as I was comparing my implementations and ideas to those of sophisticated AI modules that were developed by professional software developers for a competitive setting. As for non-technical experience gleaned from this project, it helped me learn how to effectively talk about my skills as a developer and best explain my previous work. I have found that I do a much better job of conveying my technical ideas in interviews and I think that is largely due to the confidence this project has given me. While it was not a huge up-taking, I was constantly trying to fix something or familiarize myself with something new and this now gives me the confidence that if I need to pick something up for a job, it will be a piece of cake. Project work is difficult. I found myself going through periods of not being able to make much progress and other times making leaps in implementations. It was extremely tedious trying to configure the API tools at times but it was very important that I figured those issues out in a timely fashion considering the team needs the code to work properly if progress is going to be made. It did not teach me much about project work that I hadn't learned from playing soccer (your team relies on you, encourage them, etc.) but it was more practice to prepare me for working in the real world comparable to what my internships had given me. Project management requires you to wear many hats. You need to plan for the future while ensuring the current tasks are being taken care of. Furthermore, you need to understand the situations of your teammates who are working on the project and evaluate what is actually realistic considering their workload, personal life, etc.. Working in teams is nothing new for me. Playing 20 years of organized sports has given me one of my best skills: seamlessly coming into a team and doing my best to find my fit while applying my skill set as effectively as I can without intruding on others. My first intention is to work hard and communicate effectively. If a leadership role arises and it fits me like it did on this project, I'm happy to take it. This was another opportunity to develop my skills in this area. If I could do it all over, I would spend more time trying to figure out how to create a basic neural network for our library. That would be have been an outstanding tool for new developers to familiarize themselves with and would have been awesome to explain to employers how we did it.

B. Kristen Patterson

The technical information I learned is developing a library for a large project, how to properly organize and document a large project, and the intricacies of machine learning. It also put a great importance on time management as even setting up the environment to program was such a lengthy task it could much more time than I originally thought. Since we were building a library, I also had to frequently search through documentation and try and decipher what each function use was and check to see if there was a more efficient alternative. The practice in machine learning also helped me to realize the numerous amount of conditionals that could be found when trying to replace a human with an agent. The non-technical information I learned was the experience i have taken away for a project as big as this. I have not really worked on projects as large as this and a lot of it was easily translated into other fields. Since I had worked on this project for so long and talked about it continually for months on end I was able to describe it in any level of complexity. This translates well in being able to talk about any project or have confidence in any of my methods. In order to be able to answer quickly and show that I understand my field clearly I need to just spend time to voice my projects and actions. Group work and project work require a large amount of time and dedication. When working on projects and in a group it is important to spend time organizing and try to make a schedule and keep it. It is also important to be able to designate time and resources appropriately before any work even begins. I have also realized that projects require a level of compromise and being able to give features and functions a priority based on how necessary they are to a projects success. Managing a team and project can be fairly difficult. Not only do problems show up in the form of bugs or realizing functions will not be able to be delivered, but they can also be in the form of lack of time to complete certain functions and not being able to accurately distribute resources to parts of a project. It also requires being able to understand the group's strengths and where they are best utilized. Interaction with teammates can be slightly hectic. As an individual apart of a team it is important to be able to show empathy and understanding for any misgivings that may occur in a teammates life. It is also important to not try to burden teammates with large workloads due to issues on your end. While rough, it is also important to try and keep the group on task and motivate them or provide help with their work. If I could redo this whole experience I would have mainly tried to focus more on the core requirements of the project before focusing on the stretch goals. I would have also liked to give myself more free time to work on the project in the winter by taking a lighter class load. If I had more time I would have been able to start on the actual artificial intelligence portion of the code as that would have made our project more attractive.