COMP 4303 – Final Project

DUE: TBA (Around exams)

The final project for this course is to implement an agent for either a) StarCraft AI Bot, or b) GDMC project

Projects may be completed in groups of up to 2 students. Marking will be done as follows:

Video Presentation: (40/100 marks)

Students will make a video presentation about their final project. This presentation should give an overview of what was done in the project, as well as a working live demonstration of the system. You can use screen recording software such as OBS to record your gameplay for the video. I am not judging your ability to edit video for these marks, but instead how you were able to cover the functionality of your project. This video should be uploaded to YouTube and submitted as a link with the code.

YouTube Trailer: (10/100 marks)

Students will make a 2min - 3max minute video showing their final project results. This video is solely for your benefit to add to your portfolio to use later in your career.

Working Code: (50/100 marks)
Students will submit their project code, along with documentation on how to set up and install

Project Requirements

Starcraft: You must do one of the following

a) Implement a bot from scratch that is able to successfully build a base, make units, and perform a decent attack against the built-in AI. It does not have to be a world champion! If you take all of this code from UAlbertaBot then it doesn't count, since you're just submitting UAlbertaBot. You must have novel solutions to at least some of the problems.
b) Modify UAlbertaBot to perform a new strategy that it doesn't currently do. Such as make spellcasters, use late-game flying units, build a tower defense, cannon rush, etc etc.

Minecraft: You must implement an AI agent for the GDMC competition that will be judged on the 4 competition criteria: Adaptability, Functionality, Narrative and Aesthetics. Your project will not be fully marked based on the results of those criteria, but instead on how you tried to implement those criteria. So please be specific in your video about how you tackled each of those problems. Also, I want you to focus on / incorporate at least one technique taught in the course. For example: path-finding, cellular automata, etc. It can't just be semi-randomly placing houses that you hard-coded.

For EITHER project, please choose a Game AI topic to 'focus' on and specialize for your project.
For example, path-finding, build-order planning, attacking, terrain generation, interiors, etc.

Submission Instructions:

You may submit your project to me as a single .zip file on D2L, one submission per group. Or you may submit via GitHub by adding user davechurchill (me) as a collaborator to your **private** repo. Your submission must include all source code for your project so I can compile and run it. Ideally, your project will include a compiled .exe (Starcraft) or a single python command that I can easily run.

COMP 4303 – Video Presentation Specification

For your final project, you must submit a video presentation, uploaded to YouTube. This video must be between **15-20 minutes long**. Please take care with this video, it is worth a lot of marks. Please do not try to make the entire video in one take, you are encouraged to edit together multiple recordings. The video should include the following sections, and must contain a voice-over explanation of all of the information here. Each group member should ideally do an equal amount of speaking.

1. **Introduction**
    a. A brief introduction to what the project is about
    b. Motivation for why this project is interesting and/or important
    c. What each group member worked on, specifically

2. **Problem Description**
    a. A high-level overview of the problem you are attempting to solve / goal you're trying to reach
    b. A list of all the sub-problems you may have had to solve in order to accomplish your goal (Example: You need to scout / find building locations to make a StarCraft bot)

3. **Methodology**
    a. An overview of the algorithms/methods or method used to solve your problem
    b. Why you chose those methods, citing the properties of the environment and why you believed this algorithm was the best choice for this problem
    c. Show Pseudocode of your methods, with English description of what it does
    d. Show where in your submitted code the actual algorithm is implemented (Especially important for UAlbertaBot users since the code is so large)
    e. A list of things that you may have tried to do but didn't end up working in time
    f. (Minecraft) How did you tackle each of the GDMC categories within your solution?

4. **Results & Discussion**
    a. (Starcraft)
        i. Show some clips of some of the more intelligent behavior of your bot playing games
        ii. Be sure to highlight the aspect of the game that you decided to specialize in
        iii. How did your agent perform against the built-in AI? Run 1000 games against random CPU opponents and record the win/loss ratio against each race
        iv. Discuss some observed strong and weak points of the bot. How would you fix issues?
    b. (Minecraft)
        i. Show example of generated settlements in at least 3 randomly generated areas, walk or fly around the settlements showing off the really cool parts
        ii. Be sure to highlight the aspect of the PCG that you decided to specialize in
        iii. Describe you feel your project did in each of the GDMC categories
        iv. How did you arrive at the final version of your code? Describe any evaluation that you performed to determine whether one solution was 'better' than another.

5. **Conclusion**
    a. A brief summary of what you did, and whether or not you believe you were successful
    b. What would you do in the future to make more improvements if you had more time

Assignments as Project Milestones

In order to give you as much time as possible to work on the project, Assignments 4 and 5 in the course will be project-related milestones which help you keep on track to meet the project deadline.

Assignment 4: Assignment 4 will be a project milestone to install the project APIs and get everything up and running with some simple example code.

- **Starcraft Projects**
    1) You must install Starcraft, BWAPI, and compile and run the StarterBot from UAlbertaBot.
    2) Using the example code in StarterBot, you must construct 8 Probes using the Protoss Race
    3) Once 8 probes have been constructed, you must tell them to stop gathering Minerals, and give them move commands so that they are positioned in a straight line near your Nexus
    4) You must also print the names of your group members to the screen using drawTextScreen
    5) Take a screenshot of the final result of the line of Probes and your names
    6) All of your code must be contained within the StarterBot.cpp file for this Assignment
    7) Submit your StarterBot.cpp file and your screenshot to D2L for marking

- **Minecraft GDMC Projects**
    1) You must install one of the 2 GDMC API options presented on the following page
    2) You must use the sample code to place blocks which spell out “CS4303” somewhere in the Minecraft which is visible above the ground level at position x=0, z=0 (y is height)
    3) Take a screenshot of your spectacular CS4303 Minecraft creation
    4) All of your code must be contained with a single python file
    5) Submit your python file and screenshot to D2L for marking

Starcraft

The Starcraft / BWAPI / UAlbertaBot setup process is very well documented.
I will be demonstrating it extensively in class, but you can also follow these written instructions:

https://github.com/davechurchill/ualbertabot/wiki/Installation-Instructions
https://github.com/bwapi/bwapi

GDMC

There are two main ways of setting up the GDMC framework.
All instructions can be found on the GDMC wiki: https://gendesignmc.wikidot.com/

1) Setting up GDMC with HTTP Server (Recommended)

Using this method, an HTTP server automatically starts when the game is run, and you get/set block types via HTTP GET and PUT requests. It allows you to edit blocks of a Minecraft world in real-time, which the other method does not allow you to do. This method uses an HTTP server connected to Minecraft using a Forge Mod. I set everything up and got it running in less than 15 minutes. You just have to install the current retail version of Minecraft, then install the Forge mod loader, then copy the http server jar to a specific folder, and run Minecraft. It comes with example python3 code which is extremely easy to use to get started, however any language can be used if you really want to. The problem with using another language is that you need to be able to send http requests, as well as decode the Minecraft chunk data, which is already done for you with the example python code.

https://gendesignmc.wikidot.com/wiki:submission-httpserver
https://files.minecraftforge.net/
https://github.com/nilsgawlik/gdmc_http_interface/releases/tag/v0.3.1
https://github.com/nilsgawlik/gdmc_http_client_python (sample python code)
https://github.com/yhirose/cpp-httplib (if you really want to try C++)

2) Setting up GDMC with MCEdit

This method is historically more popular, and was what people used for last year's projects. However, it is a little more cumbersome to set up. I have never personally used it, but it involves installing Minecraft, MCEdit, Anaconda, and ends up using older python 2.7 code, so I'm not a huge fan. Also, this method does not allow you to edit an actual running Minecraft game in real-time. You must run your entire algorithm to completion in order to see any changes. You can use this method if you wish, but I will not be able to personally offer any help with setting it up or running it.

https://gendesignmc.wikidot.com/wiki:submission-mcedit