H446 Computer Science: Unit 3 Programming Project

Please complete the following form for your next Unit 3 lesson. Your teacher will need to approve your proposal before you proceed.

High-level project statement

What is the problem you are trying to solve? Is it a game? A revision tool? An ecommerce app? See the 'Project Advice' document or here for ideas: http://www.codeconquest.com/programming-projects/ideas-for-programming-projects/

This is a board game engine at its core. It will try to use symbolic reasoning to play and win board games against a human. It will start with a simpler game like tic tac toe or draughts and then hopefully progress to chess. The fact that there are so many board game engines out there in the form of chess bots shows that there is demand by users. The problem I intend to solve is entertainment. (Chess is an example hush level board game my project is not specific to chess)
e.g. here are 10 other solutions:
https://www.chessstrategyonline.com/play-chess-online

Technical Platform

Describe the hardware and software you will need. What programming language will you use? What development tools? How will you build the user interface? Factor in the time taken to learn new skills. Keep things manageable in scope: you won't be able to create Amazon or Call of Duty within the time we have.

This is a purely software based project. here I will describe the minimum skeletal version:

- It will use python for symbolic reasoning (the chess engine's brains).
- The interface will be a primitive web based GUI and will use javascript or typescript (but may later use VUE js)
- Basic validation is needed to check that the move is legal. I will do this by modelling the legal moves of each piece as a set of vectors on the chess board.
- The web server that host the website will be written in flask (a python library)
- The connection between the website interface and the chess bot will be done with either http post requests or websockets
- No persistent storage will be required on the backend

I am familiar with all these skills and languages and this should be (intentionally) a very achievable project. The idea is to have a wishlist of extra features and ideas of varying complexity and scope. Each subsequent prototype will add some of these features if I have already implemented user feedback and refined my first version. The refinements will likely focus on whatever the feedback is plus making the interface more user friendly and upgrading the board game engine (to play a more complex version e.g. move from drafts to chess or to use heuristics to improve its efficacy.

The game engine will work by doing tree searches to evaluate the best move by looking ahead. More advanced heuristics techniques will allow me to discard some of the resulting trees allowing for a move in depth search. This idea in its basic form seems achievable but this is one of the key elements that I will need to learn through (heuristics and symbolic reasoning).

Note typescript is a language that is not on OCRs list so I read that I may need permission to use it. It is a strongly typed superset of javascript that I am more familiar with.

The most crucial item to add later is the ability to loggin. This will allow me to show good validation and an impressive persistent storage solution in the form of a relational sql database. This will also enable me to add new features. As part of this I will likely use SQL alchemy and then marshmallow to create a restful api or aridane to create a graph ql interface. This will be in addition to the websockets for sending moves.

Additional features wishlist:
(to allow me to show my skills if I am not feeling overwhelmed)

- Allow logged in users to access different difficulty levels
    - This would be done by keeping less efficient version of my engine as I go along: e.g. one that looks less moves ahead
- Allow logged in users to add there games against the AI to the leaderboard:
    - This can be stored in a sql table, it will be displayed all players somewhere on the website
- Store match statistics like time taken and pieces advantage over time to be added to the users history of games and allow for headline statistics of how well or otherwise the ai is doing to be displayed on the website
- Store the match as a sequence of moves allowing it to be stores and then later 're watched'
- Allow puzzles by loading in a game state to play vs the computer in which the user is close to winning if they can see how

Here is the most ambitious one, I would like to implement the others but I am not expecting to get to this. It is just here to show the scope that is available if everything is going swimmingly (wich it never does in my experience) :

- Allow 2 different users who are on the find a match part of the website to be matched up and play chess against each other
- I have seen some examples of achieving this connection via websockets but this would definitely be outside my comfort zone.

Similar solutions

You must find at least three 'similar' systems and explore them for ideas, strengths, weaknesses. You will conduct more detailed investigations later. Include a web link and a sentence or two describing each solution.

My project is board games not specifically chess but I used chess websites to look at similar systems as chess is the most popular.

- https://gameknot.com/#csolurk This solution includes some good key features like playing against other people and the computer, it also allows users to login to access additional features like a leaderboard and puzzles. It's GUI look a little old and while the games are user friendly the menu pages give information overload
- https://www.itsyourturn.com/ This solution adds many other games that can be played and offers a version that is an executable and runs on a PC. It does however have a poor user interface for usability by modern standards and only allow users that are logged in to play (I think that while this is a valid design choice many potential users will be lost)
- https://chess24.com/en This is a more broad chess website so I will just be focusing on the play option. It allows logged in users to directly challenge friends in private games and access to the leaderboard. It offers multiple parameters such as time control and difficulty. Its user interface is modern and user friendly. Clearly more resources have been used in this website than others as the result is amazing.