**CST3190 – Coursework 1 Project Proposal**

**Multiplayer connect four game website Implementing A.I algorithms**

# **Project Introduction**

I will be creating a game website of the popular game connect 4, I will be using Vue.js and implementing multiplayer into it to allow people to play the game from any given two devices. The idea for this project is to learn a A.I algorithm that would compete against other players by calculating the best move it can make with a given game board.

The connect 4’s game objective is for the player to align 4 chips in a row which can be arranged either diagonally, vertically or horizontally. There are several A.I algorithms that can be implemented to create a somewhat difficult opponent we can have a random A.I that places pieces anywhere it pleases, and we can a have an A.I with a more defensive approach that tries everything it can to defend the opposing player from making a 4 in any given position and lastly, we have an aggressive A.I which will push to make winning its number one priority.

This projects objective is to help me further strengthen the technologies I have learned through the years e.g. HTML, JavaScript and further expand my knowledge on app development frameworks such as Vue.js while learning new skills such as implementing an A.I into a simple game and learning about A.I algorithms and how they work.

For this project to work as intended I will be mainly focusing on creating the game and multiplayer features in a very strict time frame that will allow me enough time to be able to implement and start a research documentation on the A.I technologies and its implementations.

# **Roadmap**

**Objective 1**: Creating a website with the game fully implemented and running for user testing and analysis the website should be as clean as possible in terms of coding quality and Vue features implementations.

**Objective 1.5**: Test the website to ensure that most bugs are dealt with before further implementation and keep track of tests in a word or spreadsheet document for to keep as documentation for later.

**Objective 2**: Research and implement a multiplayer feature that will allow two users to play the game from separate machines. This will consist of learning the technologies that are used to implement this type of functionality to understand the code that I will be writing to avoid mistakes that could hinder me from progressing any further with my website.

**Objective 2.5**: Test multiplayer functionality and clean code to ensure readability and future proofing the code for future features that will be implemented. This will have to be done as the next stage will require me to add a lot of new functionality to the page which would be practically impossible if done in an unorganised manner.

**Objective 3**: Research into the best A.I algorithm to use to solve the game. Ensure that I understand the algorithm and the A.I that I will be using to ensure that I can explain the features that the algorithm provides and the drawbacks that may come with it. To ensure that this project is somewhat optimized as well I will be ensuring that the best algorithm is used to minimise lag and possible code crashing from to much processing. The algorithm that I will be starting with is the minimax algorithm which I will be discussing later in my report.

**Objective 4**: Implement the A.I into the game and test its functionality to ensure that its optimised properly and if possible, to optimise it further to deliver the best result in less time. Further research will be required into algorithms and optimisation and future proofing to ensure that I am using an optimal way to implement my A.I.

# **Technologies used**

## **Website**

The technologies that I will be using to implement the game will be Visual Studio Code using the Vue.js framework these will be at the core of my website to showcase the technologies that I am familiar with such as HTML, JavaScript and java/python if required at all for the server implementation.

## **Multiplayer**

The multiplayer implementation will be done using JavaScript and the overall structure for the server and client side technologies will consist of Express which is a web framework that will power my web server, socket.io is a WebSocket library that will allow communication between the browser and the server and finally Webpack which is a module bundler that I have done some research on that will allow me to fix any speed problems that I might be facing in the future from loading to many <script> files or any other problems such as scoping or dependencies problems.

## **A.I**

For my A.I implementation I will be looking into the minimax algorithm to start with and how I can implement that into my program to allow the A.I to calculate the best possible move it can make with any given board. If possible, in the near future I will try and experiment with different algorithms to see how they work and if I can implement faster algorithms that would not buffer/freeze before calculation the next possible move.

# **Gantt chart and project progression**

I have created a Gantt chart to help me visualise the progress I have to undertake to make sure I stay on schedule and meet all the requirements in the given Deadlines.

|  |  |
| --- | --- |
| Task | Deadline |
| Create a HTML Page for the game | 15/11/20 |
| Create the CSS styling for the objects that will make up the game pieces and board | 17/11/20 |
| Create game functionality e.g adding pieces on board, win/loss state. | 03/12/20 |
| Test implementations and document results | 05/12/20 |
| Research ways to implement multiplayer | 07/12/20 |
| Research server to client communication | 07/12/20 |
| Research server to client technologies | 07/12/20 |
| Background/Literature review and Initial Development | 11/12/2020 |
| Once implemented test the multiplayer communication between client and server and document testing | 15/12/20 |
| Optimize communication to ensure client and server communicate as efficiently as possible | 20/12/20 |
| Research into ai algorithms to implement | 07/12/20 |
| Create a research documentation paper that will study the optimisation of such A.I algorithms to help further optimise the A.I | Do alongside project research. |
| Research existing and known technologies that I can use to implement those algorithms | 07/12/20 |
| Once implemented optimise the A.I to decrease latency and improve response times | 10/01/21 |
| Research other algorithms to maximise performance | 25/01/21 |
| Test and document testing | 01/02/21 |
| Add extra features to the game such as Log in and leader boards | 25/02/21 |
| Finalise Report and submit final work | 12/03/21 |
| Prepare Demonstration video for submission | Same time as final report |

I have also used this website to better organise my project tasks to ensure that I can stay focused on the project and keep on track no matter what stage I’m at during the development. <https://trello.com/b/9SaxOr6Y/cst3990-project>

# **Other ideas to take into consideration and possible future projects**

To allow myself to improve a lot more in website design and create harder and more challenging tasks along the future of my projects and implementations I have also considered adding a Sudoku solver and a game of chess that studies other solving algorithms that would help me crack these games. The Sudoku game is not a 2 player or multiplayer game however I would be really interested in a algorithm that would help me make a solver for it and understand how it works.

Lastly, I would investigate a A.I algorithm that uses neural networks to play and win chess games, the project would have to be a really organised and well tested game as the neural network would have to have specific rules set into its system such as winning is good and loosing is bad also destroying other players pieces is good while loosing your own is bad. I believe that with the right algorithm and the right set of rules I can work into implementing a good A.I algorithm that will give even a moderate chess player some difficulty playing against this A.I.

# **Books and publications required**

## **Minimax algorithm research:**

**Article title:** (PDF) Rminimax: An optimally randomized MINIMAX algorithm

**Website title:** ResearchGate

**URL:**

<https://www.researchgate.net/publication/230671808_Rminimax_An_optimally_randomized_MINIMAX_algorithm>

**Article title:** (PDF) Research on Different Heuristics for Minimax Algorithm Insight from Connect-4 Game

**Website title:** ResearchGate

**URL:**

<https://www.researchgate.net/publication/331552609_Research_on_Different_Heuristics_for_Minimax_Algorithm_Insight_from_Connect-4_Game>

**Website title:** Arxiv.org

**URL:**

<https://arxiv.org/ftp/arxiv/papers/1505/1505.01603.pdf>

**Article title:** (PDF) Real-Time Connect 4 Game Using Artificial Intelligence

**Website title:** ResearchGate

**URL:**

<https://www.researchgate.net/publication/26623095_Real-Time_Connect_4_Game_Using_Artificial_Intelligence>

**Article title:** (PDF) Optimizing Multiplayer Gaming Protocols for Heterogeneous Network Environment

**Website title:** ResearchGate

**URL:** <https://www.researchgate.net/publication/224719118_Optimizing_Multiplayer_Gaming_Protocols_for_Heterogeneous_Network_Environment>

## **AI teaching itself to solve connect 4 (neural networks):**

**Article title:** AZFour: Connect Four Powered by the AlphaZero Algorithm

**Website title:** Medium

**URL:**

<https://medium.com/@sleepsonthefloor/azfour-a-connect-four-webapp-powered-by-the-alphazero-algorithm-d0c82d6f3ae9>

**Website title:** Cs229.stanford.edu

**URL:**

<http://cs229.stanford.edu/proj2019aut/data/assignment_308832_raw/26646701.pdf>

**Article title:** Learning to play Connect 4 with Deep Reinforcement Learning

**Website title:** Codebox Software

**URL:**

<https://codebox.net/pages/connect4>

## **Multiplayer optimisation and implementation:**

**Article title:** (PDF) Optimizing Multiplayer Gaming Protocols for Heterogeneous Network Environment

**Website title:** ResearchGate

**URL:**

<https://www.researchgate.net/publication/224719118_Optimizing_Multiplayer_Gaming_Protocols_for_Heterogeneous_Network_Environment>

**Website title:** D0.awsstatic.com

**URL:**

<https://d0.awsstatic.com/whitepapers/optimizing-multiplayer-game-server-performance-on-aws.pdf>

**Author** Multiplayer HTML5

**Article title:** Multiplayer Game Development with HTML5

**Website title:** Packt

**URL:**

<https://www.packtpub.com/product/multiplayer-game-development-with-html5/9781785283109>

# **Conclusion**

I believe that the difficulty of this program will help me better understand the technologies I have been using and help me improve them even more. The A.I aspect to this program will prove to be quite a challenge and help me better understand algorithms and how they calculate the best route without using to much processing power to do so.

I was also considering other games such as **Sudoku** or **Chess** that I could implement into this project however I have yet to test the limits of what I am capable of at the moment and would not wish to try out for a really ambitious Sudoku solver or A.I that has perfected chess yet. I believe the level of difficulty that I have set for this project is quite good as it will help me explore several new technologies and explore how they work.

My main focus for this project is to learn as much as possible while prioritising the deadlines set for all of my tasks, this project should be a fun and somewhat difficult piece to add to my resume once completed and polished.

I will also be making a research documentation on the algorithm that I am using and how I could further optimise its algorithms for better performance when faced with more difficult computational tasks such as a Chess or other real life processes that would otherwise take to much processing power or time if they would calculate the best solution for every move.