School of Computer Science and Informatics

Coursework Submission Cover Sheet

Please use Adobe Reader to complete this form. Other applications may cause incompatibility issues.

Student Number	C1521227
Module Code	CM2303
Submission date	04/02/2019
Hours spent on this exercise	5
Special Provision	

(Please place an x is the box above if you have provided appropriate evidence of need to the Disability & Dyslexia Service and have requested this adjustment).

Group Submission

For group submissions, *each member of the group must submit a copy of the coversheet.* Please include the student number of the group member tasked with submitting the assignment.

Student number of submitting group member	

By submitting this cover sheet you are confirming that the submission has been checked, and that the submitted files are final and complete.

Declaration

By submitting this cover sheet you are accepting the terms of the following declaration.

I hereby declare that the attached submission (or my contribution to it in the case of group submissions) is all my own work, that it has not previously been submitted for assessment and that I have not knowingly allowed it to be copied by another student. I understand that deceiving or attempting to deceive examiners by passing off the work of another writer, as one's own is plagiarism. I also understand that plagiarising another's work or knowingly allowing another student to plagiarise from my work is against the University regulations and that doing so will result in loss of marks and possible disciplinary proceedings.

Initial Report

Man vs Machine

Author: Jack Corscadden

Supervisor: Frank C Langbein

Moderator: Matthew J. W. Morgan

Module: CM2303 – One Semester Individual Project – 40 Credits

Project Description

The fear of computers usurping us humans has long been a staple in the science fiction genre. Films such as The Terminator, 2001: A Space Odyssey and Blade Runner have plots that involve the malfunctioning of, or gaining of sentience by Artificial Intelligence.

This is the inspiration for my project. **I want to build an AI that can beat me at my own game**.

There will be two main portions of this project:

Firstly I have to **build the game**, which I will call **Man vs Machine**. It will be the battleground that I will compete with the computer player on. Following on from the aforementioned paranoia around AI, the game will be a **strategy game** set in the distant future. On a war-torn planet, tribes of humans and tribes of machines have banded together in an attempt to survive against the odds.

The game will revolve around battles between these tribes. **Two opponents** can compete at any one time and the game will be playable locally by two human players, a human player and a computer player or two computer players. The game board will be divided into **tiles**, some of which are traversable and others which act as barriers. Each player will begin with a **number of units** which can move around the game board and fire at members of the opposing team. The player that **wipes out** the other players' tribe first will be the victor.

Secondly I will **create computer players** that can also play the game. A computer player needs to be a good challenge for human players. It should carry out moves that make sense and there should be an optional scalability to the difficulty (a more experienced human player may wish to play against a harder AI). I will attempt to create at least two computer players, one that uses the **Minimax algorithm** and another that uses **Monte Carlo Tree Search**. I will be able to make these two different computer players compete against one another in an attempt to find out which performs better statistically.

Part of creating a good computer player will involve designing a heuristic which can determine if a given state of play is good or bad. I will need to balance the sense of risk vs reward that the computer player uses as a metric to decide on certain moves, for example: is it worth sacrificing this unit to kill an opponent's unit?

The game **must be fun** to play but also a challenge, a computer player that **thinks ahead** and **tries to make clever** moves is essential to achieving this.

Project Aims and Objectives

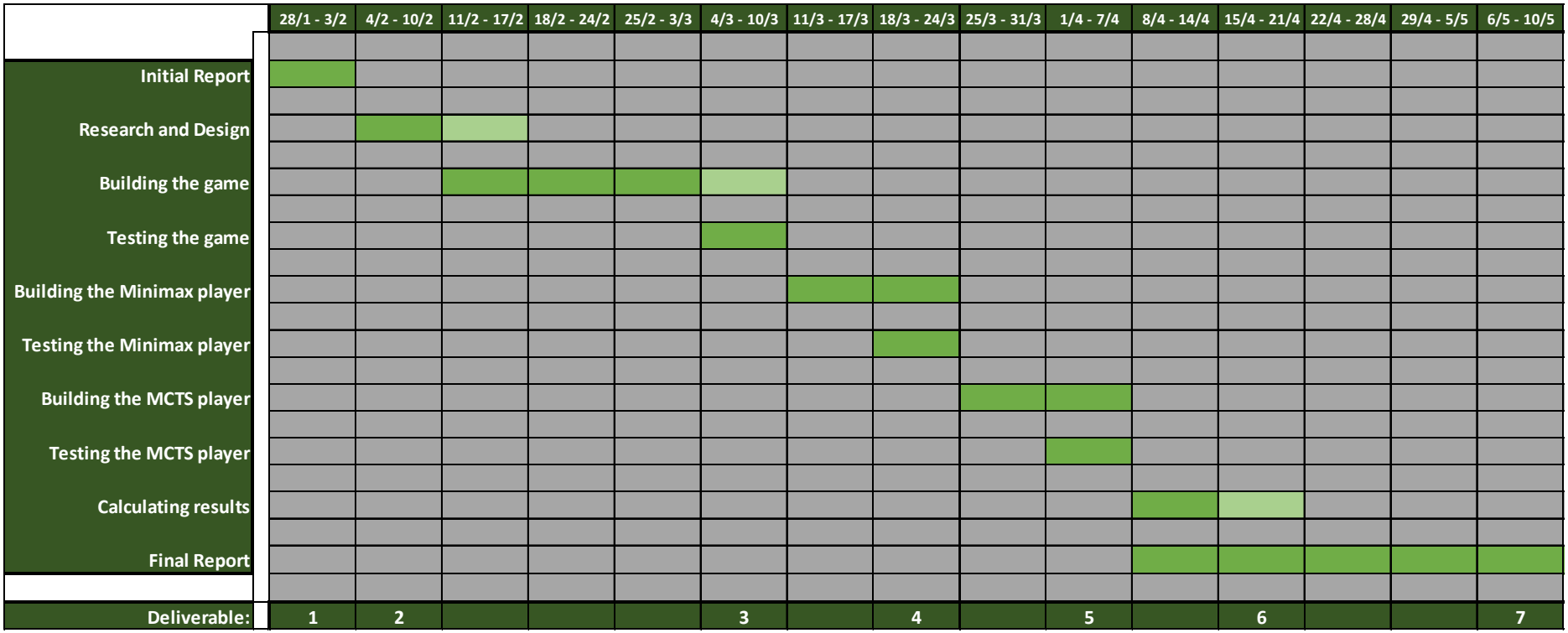
- create a **playable** game
    - the game should have a title screen which allows the user to set-up a match
    - the game should be a 2-player, 2D strategy game where each player controls a team of units and they take turns moving these units around the game board
    - each player will control several, varied units
    - the different unit types should have different properties (e.g weapon range and range of movement)
    - the winner of the game should be the player who wipes out the opposing players units first
    - the game should be playable by two human players
    - the game should be playable by two computer players
    - the game should be playable by a human player and a computer player

- the game should be **visually accessible**
    - different unit types should be visually distinct
    - when a unit is selected, its range of movement should be displayed on the game board
    - when a unit is ready to fire, its firing range should be displayed on the game board

- the game should have a **Minimax player**
    - this should be a computer player that can play the game in the same way a human player can
    - it should use the Minimax algorithm and a specific heuristic to determine which moves it should make and which it should avoid
    - it should employ the use of Alpha-Beta pruning to speed up this process
    - the Minimax player should not take too long to decide which move it is going to make
    - the moves made by the Minimax player should make sense
    - the Minimax player should have scalable difficulty
    - the Minimax player should be able to beat a human player

- the game should have a **Monte Carlo Tree Search player**
    - this player is also a computer player that can play the game
    - it should use the Monte Carlo Tree Search algorithm and a specific heuristic to determine which moves it should make
    - it should not take too long to decide upon a move
    - the moves it makes should make sense
    - the player should be able to beat a human player
    - the MCTS player should be able to function without a heuristic so that it will continue to work in different game modes (advanced feature) without additional code

Work Plan



Key:

To be completed

Potential overhang

Work Plan Steps

Initial Report: is the report you are currently reading. It includes a list of objectives for the project and a plan for achieving those objectives.

Research and Design: is an extremely important stage for the project. It is during this stage that I will decide upon a game engine, programming language, come up with feasible heuristics and design the GUI and gameplay of my game in greater detail. It will result in a large amount of documentation, some of which will appear in the final report.

I have given myself a week to carry out this stage since I already have a good idea for some of the design and I have also narrowed down the list of game frameworks I may use.

Building the game: Here I will produce the game in such a way that it will be playable between two human players. It will not have a computer player yet, but good design is essential for ensuring that an AI player can be plugged in and easily play the game.

Most of the graphical programming will be handled by the framework that I choose, allowing me to focus on the rules and logic of the game itself

At three weeks, this is the meatiest chunk of my work plan. It is quite a large task and it is important that I use good coding practices to ensure the computer players can easily plug into the game when they are created. I have been generous with the time allocated for this section as I feel it may take me a while to get used to working within my chosen framework.

Testing the game: I will need to test the game so I can iron out any potential issues which may arise. This is my first time building a game and I imagine there will be a few flaws the first time around. A period of testing now is important as discovering bugs at a later stage may cause **huge** issues.

Testing will involve unit testing and repeatedly playing the game. I will also get other people to play it to see if they can discover any issues that I may overlook. A week is an appropriate amount of time to dedicate to testing as I will obviously be testing in the previous stage as I work, this week should just be for discovering and eradicating smaller bugs.

I will also use this stage to **balance the game**, I may need to tweak the stats of different units if some of them aren't very useful or if they are too overpowered.

Building the Minimax player: When I have a playable game I will need to create a computer player for me to face off against. The first computer player I will build will use the Minimax algorithm.

Testing the Minimax player: The Minimax player will require testing also. I need to make sure it is making the correct moves and that it can play the game well (it should be able to beat me).

Testing will involve me playing against the AI and also watching the AI play itself. It is important that I understand why the AI is making each move it makes so I will have to slow the game down and focus on the decision making of the bot at every possible turn. I will need to ensure it is evaluating potential game states correctly, and I need to ensure it is choosing the correct option based on these evaluations.

Building the MCTS player: When the Minimax player is finished, I will build a computer player that uses Monte Carlo Tree Search to play the game.

Testing the MCTS player: As with the Minimax player, the MCTS player must perform moves that make sense and it should be able to beat me at the game.

Testing at this stage will be similar to the testing of the Minimax player.

Calculating results: The version of the game with both computer players available for selection is the final version. Now I will carry out numerous experiments to produce various results. The two computer players will play one another multiple times to see if there are any patterns (which search algorithm is better) and I will pit different human players up against the computer players, to see how they perform. The results accumulated from this stage will be saved and referenced in the final report.

It is important to perform a wide variety of experiments at this stage. There may be factors that affect the results (which player moves first may bias the results) so it is important to achieve a balance (average results where the human player makes the first move with results where the computer player makes the first move).

Final report: The last stage of my project is compiling everything into one document to explain what I have done, produced and discovered. I will outline the design for my project including the thought process behind each element of design. I will show the game and the computer players in action to display that I have addressed the problem adequately and finally I will address and discuss my results. Maybe there will be things I won't have time to implement or maybe there will be additional things I wish I had considered, they will all be written in the final report.

Supervisor Meetings: I have arranged to meet with my supervisor every Thursday at 13:30 in order to catch up and clear up any issues that have arisen. I will supplement these meetings with additional meetings if more help is required during the course of my project.

Work Plan Deliverables

1) Initial Report: will be produced in the first week. It is the report you are currently reading.

2) Design Documentation: will be a collection of design documents for use in the development of the game. It will include everything from sprites, rule-sets, and potential AI heuristics to pseudocode.

3) Functional Game: after coding and testing the game, I should have a product that I can play and distribute to my peers for them to also play.

4) Game with Minimax player: after coding and testing the Minimax player, it should be possible for me to play and distribute this version of the game (including the Minimax player) to my peers.

5) Game with both computer players: the final version of the game, containing both computer players which I can play and distribute to my peers.

6) Result Set: after carrying out experiments I should have a collection of results which I can reference in my final report.

7) Final Report: the final deliverable for my project. A written report describing and documenting every step I have gone through and analysing my results.