Final Year Project Plan

Chess with Artificial Intelligence

Author:   
Alex Mason

Moodle Name:   
ZEAC061

Student Number:  
100873598

Email Address:   
[ZEAC061@live.rhul.ac.uk](mailto:ZEAC061@live.rhul.ac.uk)

# Abstract

In recent years, Artificial Intelligence has been growing exponentially in the industry, especially in games. Google is a prime example of a company who has implemented Artificial Intelligence in the game Chess, which has gone onto beating some of the world’s best human chess players. This is very intriguing and interesting to me, which is why I want to research Artificial Intelligence and create a game that incorporates a basic AI system.

Therefore, for my final year project, I am going to create the well-known game Chess using the Java programming language. I am going to implement a basic form of Artificial Intelligence into the game so that single players can face the computer as the opponent, as well as having a traditional two player mode which involves two physical players.

The motivation for this project is to gain further experience with Java which would help me in my future career, as I am planning to be a software developer who hopefully specialises in Java and other object-oriented programming languages.

Adding Artificial Intelligence to the game in a single player mode adds a challenge to the project as this is an area I am not particularly strong in, despite being strongly interested in computer systems being able to “think” for themselves. This means my second motivation for this project is being able to research and implement Artificial Intelligence in games (in this project’s case chess) to also gain experience in this field whilst honing my knowledge in Java.

# Tools and Methodologies

***What tools and methodologies will I be using for my project?***

* I will use the Eclipse IDE to program the game
* I plan on using Scene Builder to create a basic GUI for the chessboard
* I will extensively document and comment my code using JavaDoc
* I will actively use Checkstyle for formatting my code
* I am going to implement design patterns wherever possible in my game, an example would be using the Singleton design pattern for my GUI
* Use JUnit Test Cases (Test Driven Development) to test the methods I am going to create in my classes

# Risk Assessment

* I think the biggest risk I could face during my project is not having enough prior experience or knowledge to fully complete a working game of chess which implements Artificial Intelligence. To mitigate this risk, I will be doing everything in stages of complexity. For example, I will focus on getting a basic game where I can move pieces, then focus on validation of pieces (can this piece be moved to that square etc), then a two player mode, then focus on creating a basic AI system which I will scale up from there. This means that if I can’t get a fairly smart AI system working, then at least I will still have a working game of chess for two players or better yet, a basic working AI system.
* Another potential risk is not doing enough research for Artificial Intelligence in games to confidently implement AI into my base game. One way to mitigate this risk is to give myself time for research in my project planning so that I can spend time researching when feasible to do so, without impacting the amount of hours/time spent towards the project. I will also be doing a report on Artificial Intelligence in Games at the beginning of term 2 so this will be useful to me when it comes to coding an AI system for my game.

* A final risk is that my timeline in my planning could be over ambitious in places with timings when it comes to doing one of my tasks. For example, it could take longer than anticipated to complete a task if I underestimate the complexity of the task. This would then delay the start of the next tasks which could potentially mean things won’t get finished on time towards the end of the project. To mitigate this risk, I will try to stick to my timings in my Gantt chart as much as possible or I could dedicate more hours to a task if its looking unlikely that I will complete it in time so that it doesn’t affect how much time I have for future tasks.

# Bibliography

***Designing an Object-Oriented Chess Engine in Java***

<https://www.e4developer.com/2018/08/16/designing-an-object-oriented-chess-engine-in-java/>

This resource has been very useful in making me think about what I could do in order to make a chess engine and the components that go into the creation of a chess engine. It has also given a few examples of possible data structures I could use as well as the kind of methods I would need for the validation of a player’s move to achieve a working game.

***Search Techniques – Computer Chess***

<https://en.wikipedia.org/wiki/Computer_chess>

This resource has helped make me think about possible approaches I can take to implement Artificial Intelligence into my chess game. Minimax is an example of a search technique which I can use for the AI, so that the AI can pick the move with the best possible payoff against the player/opponent.

***Killer Game Programming in Java***

By Andrew Davison – Published 30th May 2005 – ISBN: 0596007302

This book has so far provided me with a few useful coding techniques when it comes to programming games in Java. The book also covers 2D Java APIs, which will teach me about the fundamentals of 2D graphics and which APIs will enable me to manipulate shapes to aid the creation of my chessboard and chess pieces in my 2D GUI, if I decided to use an API to do so.

***Game Programming Patterns***

By Robert Nystrom – Published 2nd November 2014 – ISBN: 0990582906

This is a book that I am going to read for my project, based on a recommendation of the book from my advisor. It would be useful for my project because it will teach me about the implementation of different design patterns in my game, and which design pattern would be most suitable depending on what my class or program does. Furthermore, the book covers optimisation and how to write a robust game loop, which is definitely going to be useful for writing my chess game.

# Timeline of Tasks (First to Last)

1. **Initialisation/Pre-Conditions (7th October – 20th October)**

One of the first tasks I am going to do when programming my game is setting up a class which will be used to initialise the game. It will store the pre-conditions needed to start a game of chess (such as starting positions of each piece, setting captured pieces to 0 etc).

1. **Report on the History of Chess (21st October – 27th October)**

This report will coincide with the coding of my chess piece properties and move validation because it is important to research and understand the history and rules of the game if I am going to code it. Therefore, I will write my findings in the report.

1. **Properties/Validation (21st October – 10th November)**

One of the second tasks I will be doing is creating a class for each type of piece you have in a game of chess. These classes will likely contain the properties for each piece, such as the paths and number of squares they can move in one turn.

1. **Square Occupation (4th November – 24th November)**

The third task I will do after determining the properties of each chess piece is writing a class which handles the checking of whether a square is occupied or not, and if it is, by which player. If the square is occupied with a piece of the same player, the new chess piece cannot move there unless the player is castling their King with a Rook. However, if the square is occupied with a piece from the opponent, the player’s chess piece can move there and capture a piece.

1. **Report on the Coding Progression of my Project (22nd November – 29th November)**

This report will be focused on how the coding within my project has progressed now that most of the base game (chess without any Artificial Intelligence or Two Player mode) will be finished. It will include code snippets and explanations of some of the main components within the project.

1. **GUI (25th November – 8th December)**

After completing the previous three tasks first I will then be looking at creating a GUI to start testing the pieces and seeing if they will move with correspondence to their properties in order to make a valid move. There will be some sort indication to let a player know if a move is valid or invalid, perhaps an on-screen message or illuminating possible tiles the player can place their selected piece on. I will most likely go for a message letting the player know if they try to make an invalid move as I don’t want to potentially help the player by letting them know all the possible tiles they can move their piece.

1. **Interim Report (29th November – 6th December)**

The interim report will cover the work and findings I have done so far during term 1. It will refer back to some of the research I have undertaken and explain the code I have written so far.

1. **Two Players (9th December – 26th January)**

The next stage would be creating an actual playable mode. This will be the point in the game’s development where the game starts to properly take shape. I will program player turns and a checking system to see if a player has won or not.

1. **Presentation (9th December – 13th December)**

I will be presenting my project to other students and a member of the project committee. I will be demonstrating my game whilst mentioning my technical achievements, showing the code in the programs I have written and referring to important theory and algorithms.

1. **Report on Artificial Intelligence in Games (20th January – 26th January)**

This report is going to contain findings from my research of the usage of Artificial Intelligence in games. It will also contain information about a couple of AI algorithms which could prove useful for the development of my AI system for the remainder of term 2.

1. **Basic AI (27th January – 16th February)**

Once a playable game of chess has been created, I will start to think about making a basic AI system in order to have a single player mode. I will use my Artificial Intelligence in games research and research on creating a basic AI engine to help achieve this.

1. **Report on Artificial Intelligence Coding Progression (10th February – 16th February)**

This report will focus on the coding progression of my Artificial Intelligence system that I am implementing in my chess game. It will talk about what I have learnt so far and what the system is able to do so far after implementing basic AI.

1. **Final Report Draft (17th February – 28th February)**

I will create a draft for my final report, which will include overall research and findings over the course of the project. It will also include pieces from previous reports that I have written.

1. **Advanced AI (17th February – 22nd March)**

This is time-permitting and dependent on having basic AI already implemented, but I intend on creating an AI system which can play offensively (playing to capture pieces) or defensively (playing to defend pieces from being captured), or perhaps alternating between offensive and defensive plays depending on the situation or potential payoff within a game.

1. **Final Report (13th March – 27th March)**

This will be the final version of my final report. My final report will summarise my code and findings over the course of the project. It will include my smaller written reports, which were part of my research and software development.

# Gantt Chart (For Term 1)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Week 1  (07/10 – 13/10) | Week 2  (14/10 – 20/10) | Week 3  (21/10 – 27/10) | Week 4  (28/10 – 03/11) | Week 5  (04/11 – 10/11) | Week 6  (11/11 – 17/11) | Week 7  (18/11 – 24/11) | Week 8  (25/11 – 01/12) | Week 9  (02/12 – 08/12) | Week 10  (09/12 –  15/12) |
| Initialisation/Pre-Conditions |  |  |  |  |  |  |  |  |  |  |
| Report on History and Rules of Chess |
| Properties and Validation |
| Square Occupation |
| Report on Coding Progression |
| GUI Design |
| Two Player Mode |
| Interim Report |
| Presentation |

# Gantt Chart (For Term 2)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Week 1  (13/01 – 19/01) | Week 2  (20/01 – 26/01) | Week 3  (27/01 – 02/02) | Week 4  (03/02 – 09/02) | Week 5  (10/02 – 16/02) | Week 6  (17/02 – 23/02) | Week 7  (24/02 – 01/03) | Week 8  (02/03 – 08/03) | Week 9  (09/03 – 15/03) | Week 10  (16/03 –  22/03) | Week 11  (23/03 – 27/03) |
| Two Player Mode |  |  |  |  |  |  |  |  |  |  |  |
| Report on Artificial Intelligence in Games |
| Basic AI |
| Report on Artificial Intelligence Coding Progression |
| Final Report Draft |
| Advanced AI |
| Final Report |
| Presentation |