

Reversi Game Implementation

Kyle Alexander

registration: 100082709

1 Description

Reversi is a game played on an 8 x 8 Board by 2 people with 64 pieces that have both light and dark sides. Players alternate turns and in each turn a player places a single piece or "Disc" on to the board capturing all pieces between the placed piece and all pieces of the the players colour that are either horizontal, vertical or diagonal to the placed piece. Players win by having the most pieces flipped to their colour on the board at the point when neither player can legally play a piece. A move is only legal when it captures an opponents piece.

By the end of the project the aim is to have a GUI¹ with which a player can play a game of Reversi against a second player or a computer opponent. If time allows I plan to also implement varying difficulties of computer opponents as well as an option to play against an opponent over the internet using a system where one of the players hosts a game on their application and the other connects to their IP address. This will involve building an application capable of hosting a game of Reversi by listening on a port as well as presenting the game to the user and detecting player input. Of course opening to the game up to the internet entails many risks that will be detailed in Risks 2

The computer opponent will be the most complex part of the program as board game AI can range from very simple heuristic functions detailed here² all the way to large decision making trees that use alpha beta pruning to try and find the optimal moves to make while taking into account future possible moves³. If time allows then different difficulties could be implemented by performing searches on the game tree to different depths or a very simple AI may not use a game tree at all and simply perform an analysis on the existing board to make a decision.

2 Risks

There are little risks beyond the expected Project risks such as time mismanagement and inadequate testing. This changes if the online feature is added in which cases security measure must be added to verify connections and make players aware of the risks of exposing their IP address to people they cannot trust. In the proposal I foresee the following risks and possible counter measures for the final project.

¹A graphical user interface. In the context of Reversi this will be a representation of a board and display where pieces have been placed.

²Heuristic/Evaluation Function for Reversi/Othello <https://kartikkukreja.wordpress.com/2013/03/30/heuristic-function-for-reversiothello/>

³An Analysis of Heuristics in Othello https://courses.cs.washington.edu/courses/cse573/04au/Project/mini1/RUSSIA/Final_Paper.pdf

Time Mismanagement An obvious risk will be if I do not manage my time correctly and cannot meet the deadline for the final project. To avoid this I plan to write elements of the report as I build or research the associated parts of the program so that I can avoid having to go back through my work at the last moment to finish the report.

Inadequate Testing If not enough testing is done and a lot of bugs are still in the finished game then this would damage my marks, especially when the final code for this program should not number a huge amount. To stop this I plan to test as I build parts of the application as well as dedicate time just for testing at some point in the year.

Online Play Like mentioned before if an online feature is implemented there will inherently be security risks. Given that no secure data will be transmitted by the game the only risks I can foresee is guaranteeing that data transmitted is verified and from the computers that first started the game. There is also a risk in that users will have to communicate their IP addresses to each other. There is little the application can do in this regard other to inform the user of the risk and suggest they use a secure channel to exchange the data.

3 Proposed Solution

Given that Reversi is a turn based game there is little need for any real time processing so my decision on programming language is largely made on what I believe myself to be most proficient in and provides easy to use 2 dimensional graphical libraries. For these reasons I have decided to use the Java programming language along with its Swing widget Toolkit for window management⁴ and the Java AWT `BufferStrategy`⁵ as it provides 2D primitive drawing through a graphics object as well as double buffering for a smooth GUI.

During the first week of the year I started to prototype a very simple Reversi game and using a 2 dimensional integer array to represent the board as well as some predefined constants to represent the state of each board position I finished a working 2 player Reversi game. The game has had limited testing but due to the small amount of logic involved there appear to be no bugs. The logic to work out whether a move is legal as well as what pieces to capture is very poor and I imagine can be refined to include a lot more code reuse but it works outwards from the piece last placed checking in the

⁴JFrame Documentation <https://docs.oracle.com/javase/7/docs/api/javax/swing/JFrame.html>

⁵`BufferStrategy` Documentation <https://docs.oracle.com/javase/7/docs/api/java/awt/image/BufferStrategy.html>

cardinal and diagonal directions. As it works outwards it tests the board cells and if it finds any number of opponents pieces followed by a piece of the players own colour the move is legal.

4 Proposed Time Management

Since I have made a working 2 player prototype my focus on the first semester will be researching game trees and the various evaluation functions used in Reversi. During this time I also hope to add a menu to the game that will allow players to choose whether they want to play against another person or a computer as well as set difficulty. Depending on how fast the research progresses I plan to implement A very simple computer opponent that does not look at future possibilities but makes a move based on the current state of the board only.

Once the Christmas break starts I will hopefully be in a position to start implementing more advanced AI principles such as a game tree as well as traversal methods like alpha beta pruning. The second semester will be a continuation of this as well as looking into the option of adding the online play feature.

The report as mentioned before will be written in tandem with the above mentioned programming and research to avoid me having to read over my own code and re read any material. A timeline can be seen in the Gantt chart below¹

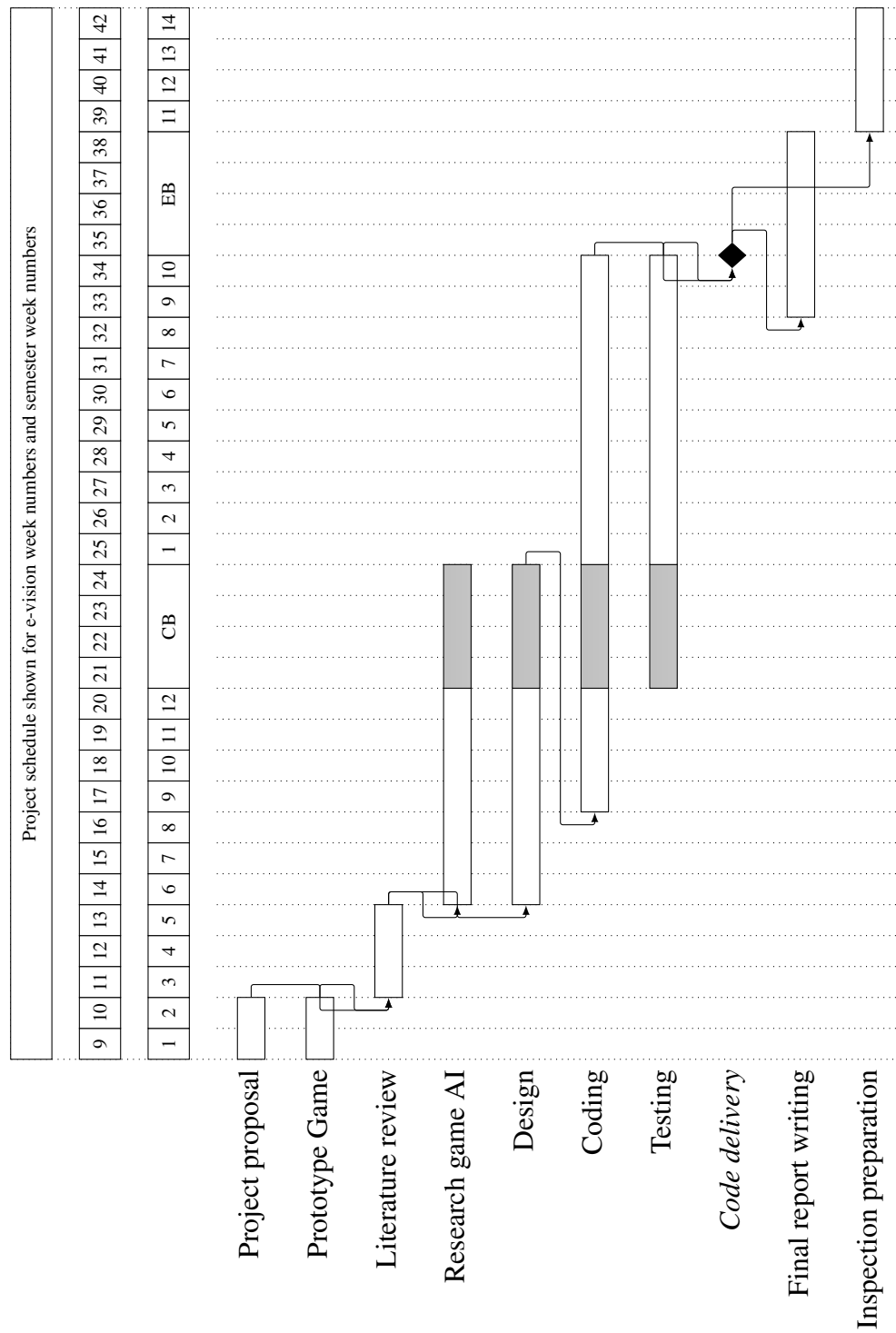


Figure 1: Project Gantt chart

Project proposal

Description of project: aims, motivation, understanding of issues, problems	First	2.1	2.2	3	Fail
Resources, references: evidence of preliminary work to identify key resources, initial reading	First	2.1	2.2	3	Fail
Proposed approaches: relevance, suitability, appropriateness	First	2.1	2.2	3	Fail
Risks: identification, suitable contingency planning	First	2.1	2.2	3	Fail

Quality of writing

Clarity, structure correctness of writing	First	2.1	2.2	3	Fail
Presentation conforms to style	First	2.1	2.2	3	Fail

Workplan

Measurable objectives : appropriate, realistic, timely	First	2.1	2.2	3	Fail
Gantt chart: legibility, clarity, feasibility of schedule	First	2.1	2.2	3	Fail

Comments

<div style="border: 1px solid black; height: 480px; width: 100%;"></div>
--

Supervisor: supervisor

Markers should circle the appropriate level of performance in each section. Report and evaluation sheet should be collected by the student from the supervisor.