School of Computing
FACULTY OF ENGINEERING AND
PHYSICAL SCIENCES

Final Report

Remote Chess Playing

Anthony Moran

Submitted in accordance with the requirements for the degree of
BSc Computer Science and Mathematics

2022/2023

COMP3931 Individual Project

The candidate confirms that the following have been submitted.

Items	Format	Recipient(s) and Date
Final Report	PDF file	Uploaded to Minerva (DD/MM/YY)
Link to Online Repository	URL	
Link to Video Demo	URL	

The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

(Signature of Student) _____

Summary

My main objective is to build a web service that allows users to play chess over a network. *My current solution is using the peer-to-peer architecture, I need to research whether it is possible to port my solution to users over a wide area network*.
Users will be able to register an account, and become friends with other users on the platform.
The server will be able to keep a record of users' games and the "elo"/skill of each user.
A user can be a part of multiple games with different people simultaneously.

Acknowledgements

<The page should contain any acknowledgements to those who have assisted with your work. Where you have worked as part of a team, you should, where appropriate, reference to any contribution made by other to the project.>

Note that it is not acceptable to solicit assistance on 'proof reading' which is defined as the "the systematic checking and identification of errors in spelling, punctuation, grammar and sentence construction, formatting and layout in the test"; see
`https://www.leeds.ac.uk/secretariat/documents/proof_reading_policy.pdf`

Contents

Chapter 1

Introduction and Background Research

1.1 Introduction

Chess is a board game that has existed since the 7th century [**?**]. The game is played by two people, each one in control of an army of equal strength; it is up to the player's logical reasoning and deduction in order to conquer the board. For a long time, chess could only be played in person, or perhaps through the post. Internet Chess Club was founded by Danny Sleator in 1992 [**?**], and he lead a small team of programmers to develop the first dedicated chess server. This was the introduction to playing chess over the internet and allowed people to play chess together, regardless of the distance between them. It wasn't until 1995, where the first web based chess server was launched by Caissa [**?**], which featured a graphical user interface. This most likely contributed to a higher adoption of playing chess online because it gave users a friendlier interface, which was more intuitive and approachable than what was previously available. Since then, many similar services have been created such as <list examples> and it goes to show that there is quite some variety in the way this service is implemented.

1.2 Literature review

1.3 Libraries and Tools

Chapter 2

Methods

All of the code developed for this project has been managed by git and pushed to my online github respository with can be accessed by the link:

https://github.com/Anthony-Moran/Final-Year-Project

The history will show the various stages of my project and how I started off with a blank workspace and ended up with this project now.

2.1 Back End

Although the front and back end of this project have been worked on simultaneously, I will begin by describing the process for how I designed the back end first. This is due to the fact that the front end depends on the features developed on the back end in order to function. I have developed the back end using Python 3.10, the Flask framework and the Flask-RESTful extension to allow for a RESTful implementation of my service. I have chosen this language for my backend because I have previous experience building web services with it, and the quality of the code should reflect this. Additionally, the framework abstracts a lot of the processing involved for developing a REST based service, which meant I could focus more on the business logic of my application.

2.1.1 Flask/Flask-RESTful

Before delving into any web service, there is some boilerplate that is required. This can be seen in one of the early commits to this project where I set up an initial connection to the local host and add my first resource, routed to the base URL. For those unfamiliar with flask-RESTful, the way that it functions is by assigning everthing as a resource *referencing may be necessary*. Each resoucrce can be assigned any of the 4 CRUD operations, i.e. Create, Read, Update or Delete, via a POST, GET, PUT and DELETE request respecitvely. The initial resource only features a get request and it returns the index.html file.

2.1.2 Chess Logic

Add some general text about the implementation of the chess logic here

2.2 Front End

Add some general text about the front end here

2.2.1 The Welcome Page

Add some general text about the welcome page here

2.2.2 The Main Page

Add some general text about the main page here

2.2.3 The Error Page

Add some general text about the error page here

Chapter 3

Results

Test everything works. Was the solution limited to peer-to-peer or was it able to expand beyond that?

3.1 Starting a Game

3.2 Managing Multiple Games

3.3 End Game

e.g. Discarding the game state from the server

Chapter 4

Discussion

4.1 Conclusions

4.2 Features to be added in the future

What I would add if I had more time/resources.

Appendix A

Self-appraisal

<This appendix should contain everything covered by the 'self-appraisal' criterion in the mark scheme. Although there is no length limit for this section, 2—4 pages will normally be sufficient. The format of this section is not prescribed, but you may like to organise your discussion into the following sections and subsections.>

A.1 Critical self-evaluation

A.2 Personal reflection and lessons learned

A.3 Legal, social, ethical and professional issues

<Refer to each of these issues in turn. If one or more is not relevant to your project, you should still explain *why* you think it was not relevant.>

A.3.1 Legal issues

A.3.2 Social issues

A.3.3 Ethical issues

A.3.4 Professional issues

Appendix B

External Material

<This appendix should provide a brief record of materials used in the solution that are not the student's own work. Such materials might be pieces of codes made available from a research group/company or from the internet, datasets prepared by external users or any preliminary materials/drafts/notes provided by a supervisor. It should be clear what was used as ready-made components and what was developed as part of the project. This appendix should be included even if no external materials were used, in which case a statement to that effect is all that is required.>