Discussion – Philip Mortimer

Project Description

- I intend to create a fully fledged chess application. The chess game will be targeted towards a general audience of consumers and as such will include a graphical user interface of high quality. To input a move, the user will click on a piece which will result in all direct legal moves being highlighted. The user then clicks on one of these highlighted squares to make a move.
- My project will support two players playing offline on the same computer, a player and a computer playing offline, a computer and a computer playing offline together and two players playing online against each other together. Additionally, each player will have the option to participate in offline and online tournaments. In offline tournaments, the user may customise the number of participants in a tournament as well as the difficulty of the computer.

Project Description

- My project will have a login system (with the option to login as a guest). Guests are only able to play offline. Players who create an account are able to play online and offline and can even play games against friends online. The application will keep track of the games an individual player has played, allowing them to see a “replay” of a match they have seen. Players will also be able to partake in multiple offline matches at a time by saving the details of each game they are playing. This record of their matches will be used to give them a basic rank for offline and online play and they may choose to share their online ranking to see themselves on an online leader board.
- The computer will have varying degrees of difficulty which will accommodate novices, intermediate players and highly skilled players.

Project Description

- For my computer, I plan to use a few different algorithms. For the lower difficulties, I plan to use a depth limited minimax algorithm that uses various established heuristics of board value (such as the comparison of the number of pieces each side has along with their weighted values). By altering the depth of the search, adding a variable that determines the probability of choosing the best option (i.e. determines how often the computer system chooses the worser option) and using heuristics of various sophistication, I can create varying levels of difficulty.

Project Description

- For higher difficulties, the aforementioned algorithm can be highly effective. However, for these difficulties, I plan on using a feedforward artificial neural network to determine the value of a specific board state. At the moment, I am exploring multiple methods of training the network including utilising a database of grandmaster moves (which is publicly available) , a larger database of chess games (which isn`t just limited to grandmasters and is also publicly available) or using the open source Stockfish engine as my training data. However, I haven`t yet decided which one of these I will use.

Project Description

- For online play, I plan to use a centralised java server that is running. When a player makes a move, they send an updated text file to the server. Thus, when the other play wishes to make a move, they connect to the server and download the file containing the details of the current game.

Language and IDE

- I plan to develop my project (which is a game) using the Java programming language and the NetBeans 8.2 IDE.
- I have chosen NetBeans as it provides an array of highly useful features, such as the highlighting of syntax in various colours (aiding readability). Furthermore, it highlights syntactically incorrect code, making it far easier to spot errors. NetBeans also offers comprehensive debugging features such as a variable watch and breakpoint. NetBeans also offers a GUI builder for java that is easy to use and I am comfortable with.
- I chose Java because the object oriented nature lends itself well to a game like a chess and reduces code duplication. Furthermore, the portability of code allows my program to be utilised by devices of different operating system which is useful considering that the software will be distributed across an audience.

Language and IDE

- Java also supports features such as exception handling which is useful for things such as type checks and file handling and also makes the language more versatile.

Client

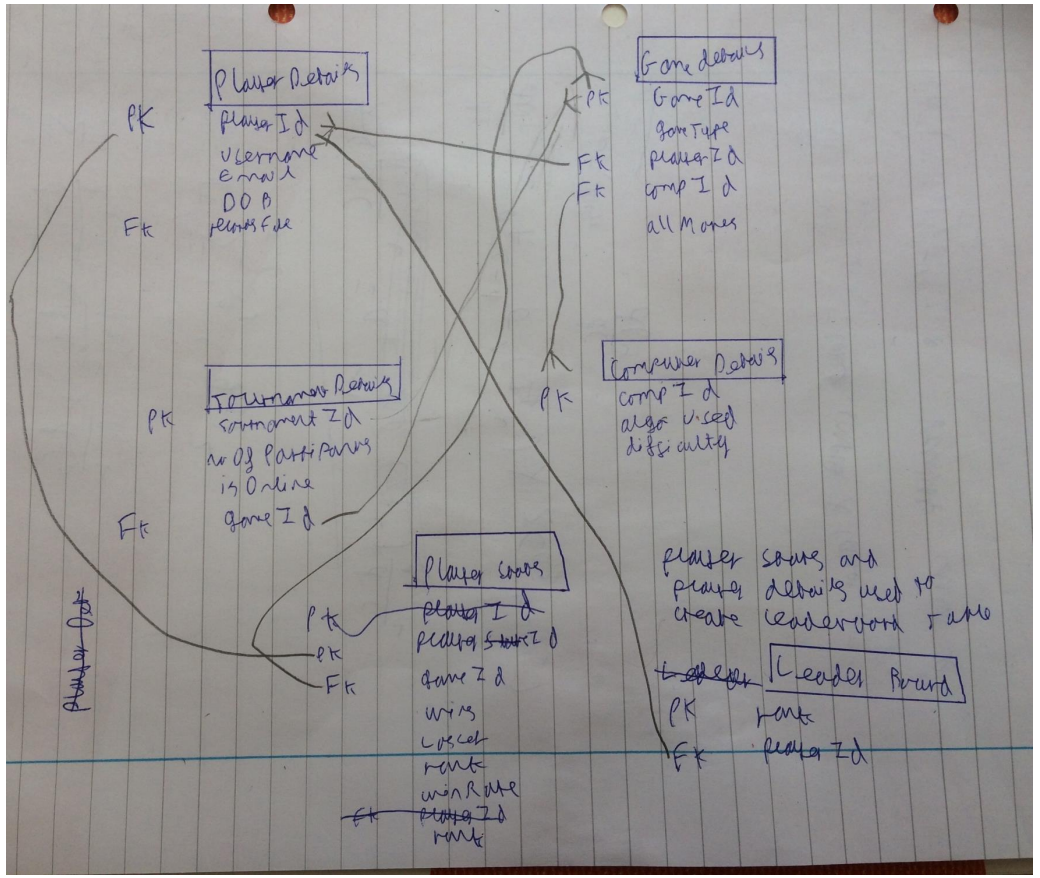
- My client is the West Cross Chess Club which would like to develop a system to facilitate their members to play locally and online together in order to improve their skill. This is particularly important to them as COVID-19 prevents them from physically meeting up at the moment. They would like the ability to play chess regardless of lockdown restrictions and whether it is the summer holiday or not.

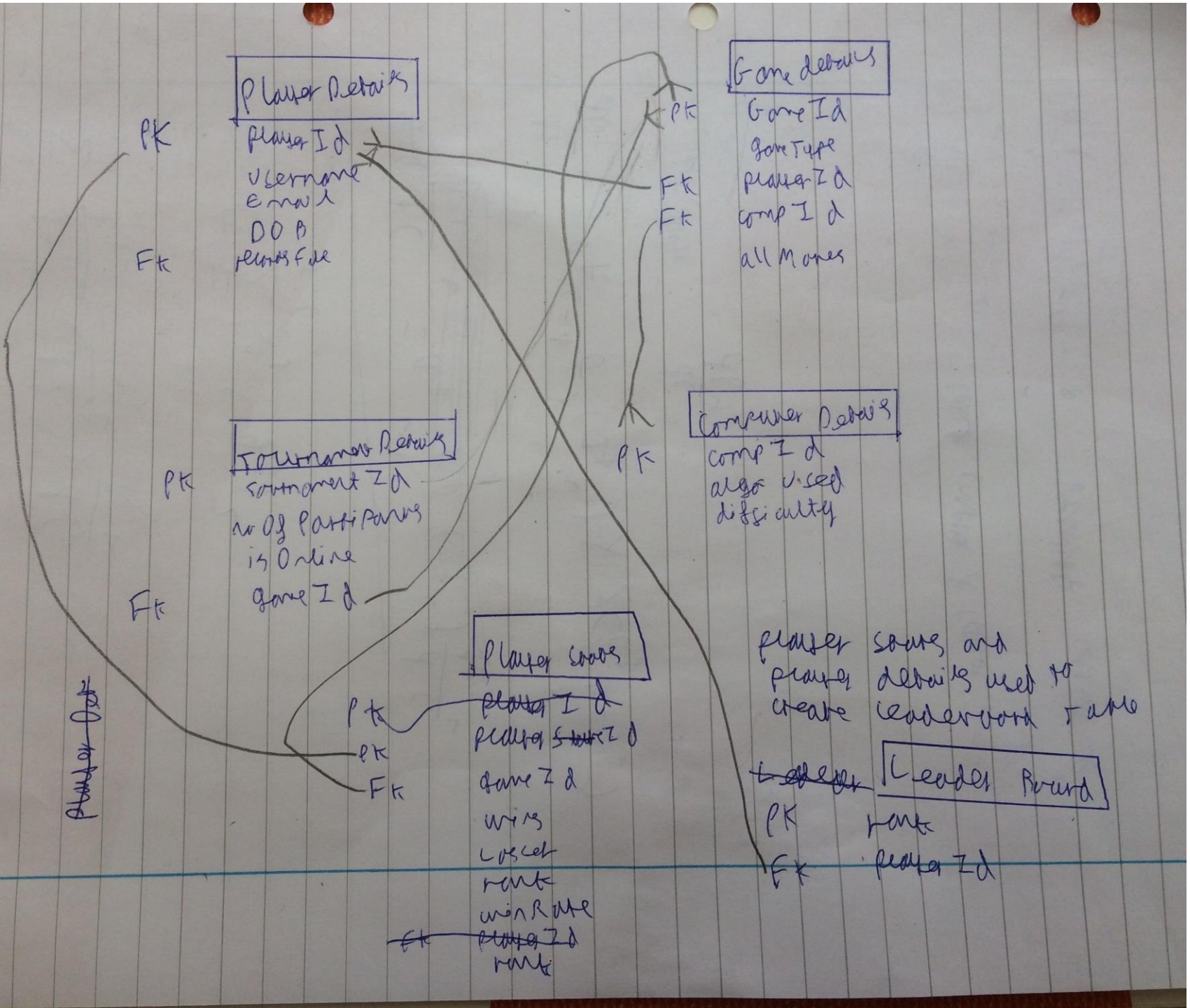
Files

- For my application, there will be four main entities. Player, tournament, game and computer type. Using these we can create tables that keep track of all the games a player has played along with win/rate etc. Thus we can achieve the files needed by storing, player details, tournament details, game details, player stats and player leader board. Additionally, as we are using a server, we will need a text file containing details of all ongoing games and the players participating in them. This server will also store the leader board and the login details of players.

Entity Relationship Diagrams

- These are my proposed ERD`s and they aren`t exhaustive or finalised and (like the rest of the discussion) are designed to give a rough idea of what the system will be like.





I would also strive to store data in the third normal form wherever possible to reduce data redundancy and duplication. This means that data stored in a table will be fully dependent on the primary key.
Additionally there will also be a table storing login details in the centralised server.

File Handling

- Each time a new user is added, a record will be added to the file that stores all the login details of users (which is stored in the centralised server). Furthermore, when a user finishes a game of chess, a record will need to be added to the game id table and the table that keeps track of a players stats will need to be updated (e.g. to show the new win/draw/loss ratio). To allow the player to play multiple games at once, these games need to be stored in text file. However, when one of these games is finished, the data needs to be moved to the file keeping track of all the games the user has ever played and the file that maybe named “game3” will need to be cleared.

File Handling

- When a new user is created, they will need to be added to the text file storing all login details. Each time a user plays a new game, the file storing their stats will need to be updated. Additionally, I will be handling records of multiple different types such as Boolean (is game finished), string (date of birth) and integer (number of wins). If a player wishes to delete their account, that record will be deleted from the login details file. The leader board records also would need to be edited to update the leader board.

Calculation

- In terms of calculations, there will be numerous complex calculations. For example, the calculation of the rank of a user. Calculating the value of a board position using various heuristics will be complex, particularly when combined with running these values up through a search tree. The neural network will also have many complex calculations to both run a value through a network and to calculate how the network should be altered using backpropagation and gradient descent. Determining what moves are legal from a given position will also be complex to determine as it will require many possible options to be considered.

Searching and Sorting

- Each time a player logs in, the text file storing all the user details will be stored will be searched for the relevant username and password. This will probably be done using a binary search. Furthermore, the computer playing chess will have to search for a given number of board states and will use technique known as depth first search to optimise the search tree (using alpha-beta pruning). The player details file will need to be sorted each time a users detail is added to it. The sort may simply be a form of insertion sort where the unsorted segment of the list is merely the new item or something more complex such as merge sort or quicksort. Furthermore, files such as game details and tournament details will need to be sorted in order of primary key to enable quick access of data.

Searching and Sorting

- The leader board of all the players will of course need to be ranked. When updating a players stats, the file logging all of a players games will need to be to searched to determine their win rate both online and offline

Validation and Verification

- When the user inputs creates an account, they will input numerous fields. All required fields (such as password and username will have presence checks). I will apply a format check to the date of birth field to ensure that it is in the form dd/mm/yyyy. Then I will use this to calculate their age and perform a range check (i.e. if their age is over 140 then the input won`t be accepted). I will also devise an algorithm to determine whether the date entered is a valid one. For password, I will employ a length check to ensure that it is greater than 4 characters to ensure some level of security. I will also have an optional field in which the user can input the number of years they have been playing chess for. A type check will be applied to this field to ensure that the input is an integer and this number can not be greater than the age of the user.

Validation and Verification

- For username, a look up check (of sorts) would be performed to see if the username is already taken. The same would be done with the email address which would have a format check to ensure that the “@” and “.” symbols were present in the address as well as no upper case letters.
- For the username , password and email fields I will utilise double entry verification. Once all details have been input and validated, I will ask the users to confirm the details using screen based verification.

Other features

- On top of all the aforementioned features, I will also have a clock that displays the time that each game is lasting for. This will be programmed through multi threading.

Possible Limitations

- My project will be limited by the fact that the software will be distributed to a wide audience and as such, the computer can`t take a ridiculous amount of time to make a move. The device that I will use to develop the neural network will also not be very powerful and hence the computer won`t play as well as it would if it were trained on a supercomputer. Again, due to my hardware, reinforcement learning methods such as Q learning are infeasible. And as I am using a supervised neural network, the maximum skill of my bot will be limited by the quality of the training data. Furthermore, my system will not be able to do every thing as well as “real” chess system such as support multiple different external engines

Possible Limitations

as I do not have a large team of developers working on the project with me. However, I will be able to develop a system that is on par with professionally developed chess software in many regards. Although, a minor limitation, the fact that my application is written in Java makes it slightly harder for users to access as they need to download the game locally and install Java when compared to a website. Conversely, it has the advantage of being accessible without an internet connection once it has been downloaded. As Java does not have matrix libraries that are as optimised as libraries in other language, or as comprehensive, I will write my own matrix library which will naturally be slightly less optimised than something like “numpy”. The neural network framework will also be slightly less optimised as I will not make use of existing frameworks which have been developed by dozen`s of people over a few years such as “TensorFlow”.

Feedback	Accepted?	Justification
A lot of features – consider whether all of them are necessary for a chess club system. E.g. playing against a neural network and tournament system over the top and not needed. Core of the club should be allowing members to play against each other / offline.	Yes	I agree, I think a neural network structure is excessive and broadly irrelevant to the task. A tournament structure is also not necessary given that my client is the West Cross Chess Club.
Neural network seems excessive and over the top.	Yes	A neural network is probably too computationally expensive for most members' computers.
Online play seems a difficult feature and something that could be replaced by offline play. The club could use the offline feature and a website like chess.com instead for online play.	No	I concede that online play is not an essential feature and I may well consider dropping it. I shall develop a prototype and then use this to conclude whether online play is a viable feature or whether it should be replaced with use of an already existing website.
Feature to play as a guest and access various features of the club should be implemented.	Yes	Having the ability to play without logging in is a useful feature for many who may not remember passwords / who may not have created an account.
The ability to customise who is what colour as well as rulesets should be added.	Yes	A customisable ruleset is important and a standard software feature and hence I shall implement it.