SOFTWARE DESIGN SPECIFICATION 4PC PREDICTOR

Group 2

SWEPM GROUP-2 THE-3

1. Introduction

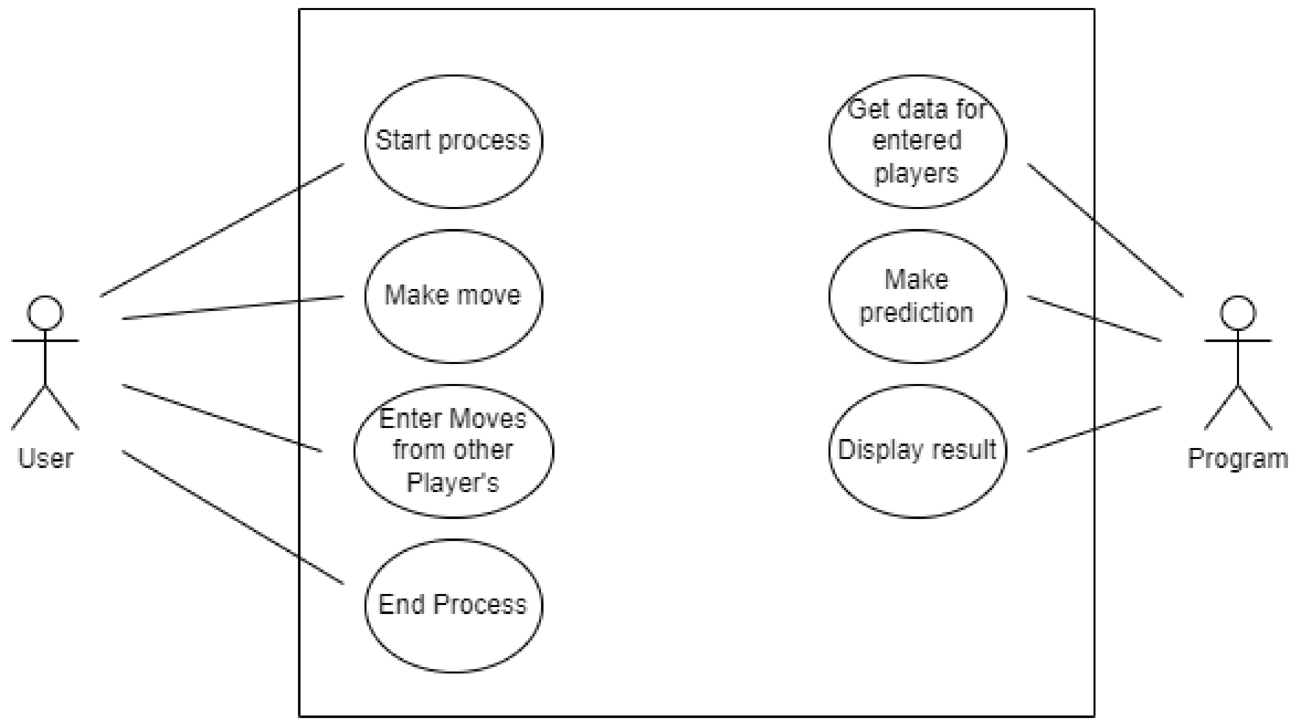
Four-player chess (also known as Four-handed, Four-man, or Four-way chess) is a family of chess variants typically played with four people. A special board made of standard 8×8 squares with an additional 3 rows of 8 cells extending from each side is common. Four sets of differently coloured pieces are needed to play these variants. Four-player chess generally follows the same basic rules followed on regular chess. Exceptions to these rules include nuances of when checkmate is delivered, depending on the variant, on what rank a pawn promotes, and the ability to capture a player's king, which takes priority over checkmate in the team's variant. There are many different rule variations, most variants, however, share the same board and similar piece setup.

We are building a local program that connects to a previously created database file in the same directory as the program from data given by Chess.com. The user is able to enter current made moves, the program than gives a percentile prediction, which colour which player could be. This will help the user understand the play, his opponents and their moves which can help him to improve his game and his understanding.

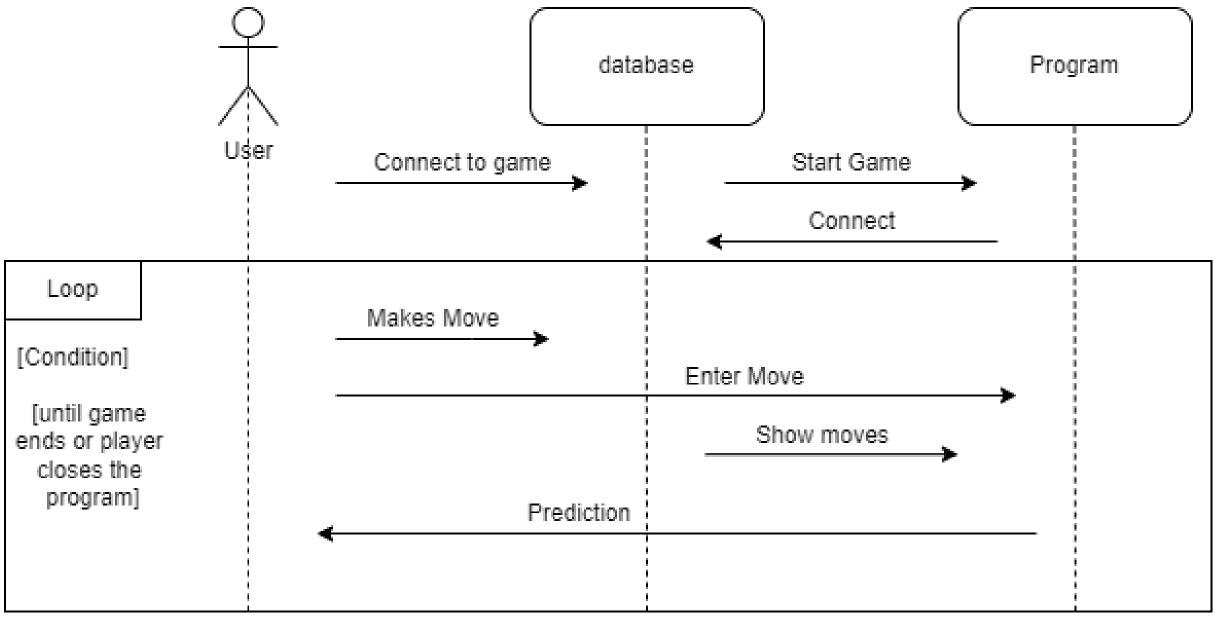
2. Architectural Design

External Perspective:

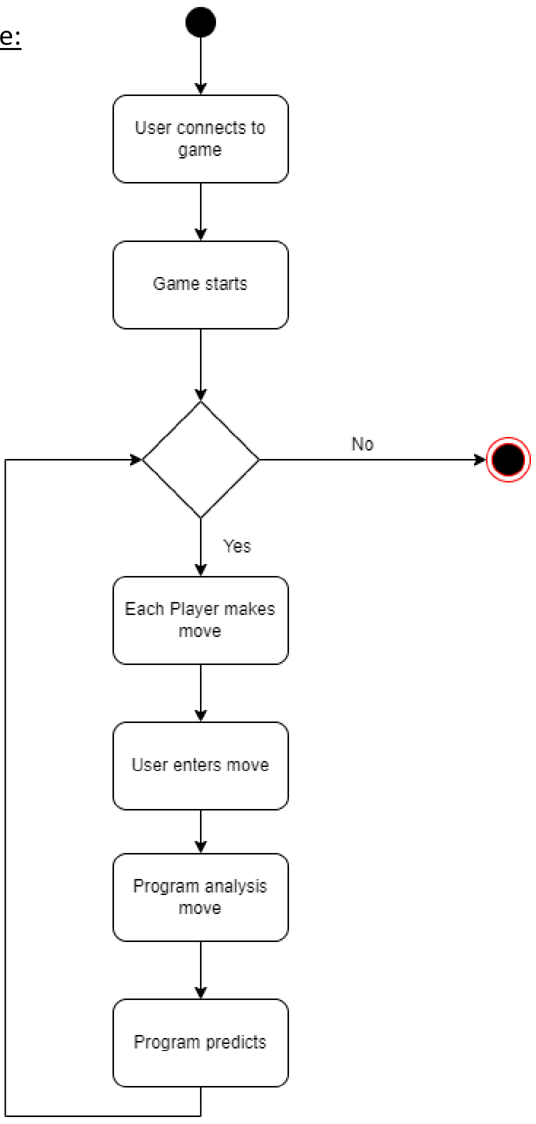
Container



Interaction Perspective:



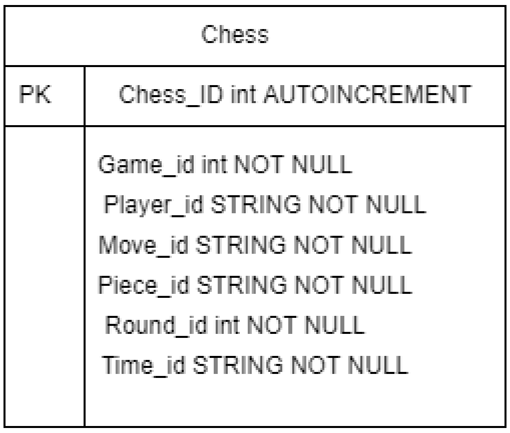
Behavioral Perspective:



3. <u>Database Design</u>

We will use SQL Lite as the Database library. There are many concepts to think through how the database relations are in the end build up for the prediction software. One possible approach is that the program observes every move from every player taken each round and loads it in the database. This pattern is than matched with patterns from players games which were already played and loaded in the database.
Also it is not yet clear how the machine model will look or if a machine model will be used. Nevertheless final design of database is a matter of change because programming such a prediction software will be a try and error race.

<u>Entity Relationship Diagram:</u>



Chess	
PK	Chess_ID int AUTOINCREMENT
	Game_id int NOT NULL Player_id STRING NOT NULL Move_id STRING NOT NULL Piece_id STRING NOT NULL Round_id int NOT NULL Time_id STRING NOT NULL

4. <u>Components selection, design and Interface design</u>

Chess.com database file:
A connection to a database file created with the python library “sqlite” from txt files, received from Chess.com is established. With “sqlite” querry’s data is sent to the Algorithm.

Algorithm (prediction):
Given the move the user enters, the algorithm compares the entered moves to previous games in the database and predicts to which percentile chance the given pattern fits to a specific player. The more moves are the same in the same round, the more per round that match, the higher the percentile prediction.
For example: 3 rounds in the current match have already passed, if the algorithm finds that move 1 and 3 are matching with one of the past games rounds 1, 2 and 3 there is a 75% prediction.

Interface:

ELEMENTS:

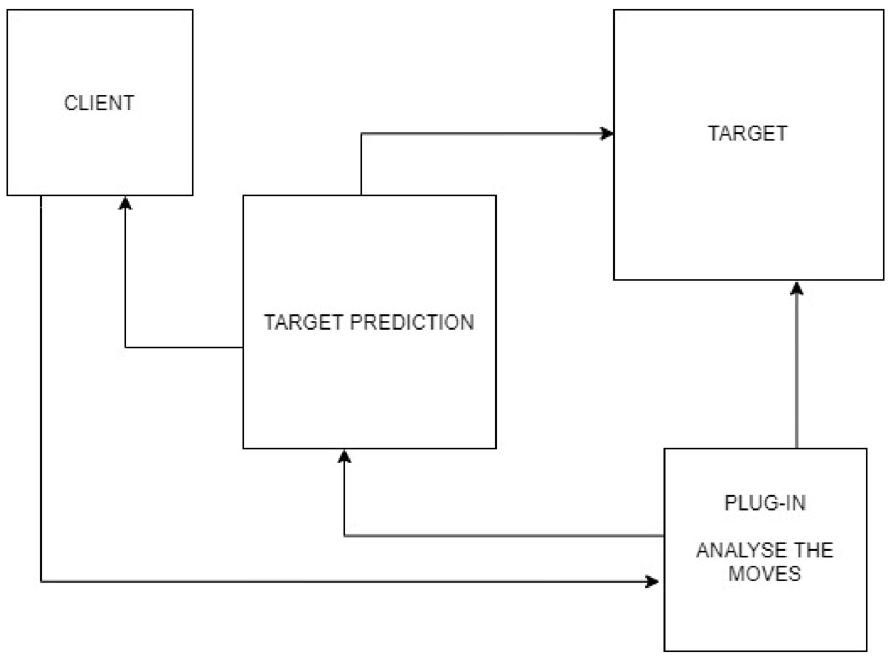
- Information input (moves from other players)
- Algorithm output
- Data from chess.com
- Modules

User enters moves from other players:
When user enters the move that was made by another player, the system provides this moves to the algorithm.
The algorithm requests a list of data from the database file containing the moves entered and compares to each player which pattern fits the most and returns a potential percentile prediction for each player fitting.
Prediction accuracy increases as players moves increases.

Interface consists of 2 parts: Field to enter player moves, list of player names with percentile prediction.

<u>Field to enter player moves:</u>
Here the user is able to enter player moves and submit them with submit button below.

<u>List of player names with percentile prediction:</u>
Output of the algorithm, a list with player names and their percentile prediction each is shown.



Prototype plan/outline:

Start
When the user opens the program and clicks the start button, an input field with a submit button on the left and a list (at the beginning empty) with future predictions and player names with colours on the right is presented.

Player prediction:
If a valid move is entered and submitted with the “submit” button, the algorithm starts to query move lists with player names from the database and returns the player names with the percentile prediction corresponding to it, the interface than displays the names and the predictions with corresponding colours in the right section of the Interface.

While running:
The player needs to enter every single move each round, the Algorithm automatically declares for each move a colour, the first move entered each round is always Red, second Blue, third Yellow, fourth Green.
The more rounds are played and therefore more moves are entered, the more precisely is the prediction.
Also, the user is able to read a Documentation about the program by clicking on the “Help” button.
Restart button to restart the program representing a new game is starting.
End Button to end the Program.