Initial Plan - Strategic Reasoning with Artificial Intelligence

Jacob Berg (Student) Federico Cerutti (Supervisor)

CM3203 - One Semester Individual Project (40 Credits)

Project Description

The problem I am trying to solve is an issue of trust, more specifically, detecting whether or not your opponent is consorting with a supposedly independent referee to gain an advantage over you. For this, I will be using the game of Kriegspiel [1], which is a variation of chess in which you cannot see your opponents pieces and you must rely on an independant referee to give you information, such as when a piece is taken or if a move is legal. This partial information makes it very difficult to fully understand the current state of the game and relies on the players having to make guesses as to where the opponents pieces are. In the first few moves of a game, it can be easy for humans to detect if they are being cheated against if the opponent makes a move that is clearly illegal. However, past this point it is very difficult for us to know if they are being cheated against because the number of possible moves that an opponent could have made grows exponentially. I am aiming to use Artificial Intelligence to solve this problem.

My overall aims are to create an open-source implementation of the game Kriegspiel and to implement a system that analyses information given by the referee to determine whether or not they are cheating against you.

Project Aims & Objectives

My first aim will be to create an open-source implementation of Kriegspiel. This will allow me to fully understand how the implementation functions and will allow me to include any additional features that will be required for the rest of the project, such as obtaining the current gamestate from a player's perspective and keeping a history of the referee's statements.

I have done research into existing open-source implementations of Kriegspiel and I have decided that they would be unsuitable for use in this project. I was able to find one existing open-source implementation [2] of Kriegspiel, however it does not allow the opponent to use cheating moves and also includes many additional features such as a web-based interface and the ability for users to log-in which would add unnecessary complexity to the project. For these reasons, I have decided that it would be best to implement the game myself.

This objective can be broken into several smaller goals. I first plan to create a version of Kriegspiel in which you face an opponent which uses random moves, that is playable through the command line.

If I have additional time during the implementation period, I would like to add a graphical user interface to the game. However, this is not necessary for the core objectives of the project so this will be a very low priority for my implementation of Kriegspiel.

I am aiming to create a system that will extend my implementation of Kriegspiel to detect if the referee is working with your opponent to cheat. This system will only have knowledge of where one player's pieces are and a history of what the referee has said. I plan to use this information to detect if there is a conflict between what the referee has allowed and the current state of the game. By doing this, the system should be able to detect if the opponent is cheating. There are several ways that an opponent could cheat with the help of the referee, these include:

- Moving one of their pieces into a position that is not in it's allowed movespace (such as moving a rook diagonally)
- The referee not informing the player of a piece being taken
- The referee not allowing the player to make a legal move by informing that them that it is illegal

I plan to focus on detecting whether a piece has been moved to a location it should not be legally able to be. If I manage to implement this with enough

additional time, I will work on implementing detection of other methods of cheating.

Work Plan

I have decided on the following milestones.

- Initial plan submitted
- Command line implementation of kriegspiel is playable
- System detects a cheating move
- Final report submitted

29 Jan - 5 Feb Write initial plan Milestone - Submit initial Plan

6 Feb - 8 Feb Design game implementation

8 Feb - 27 Feb Implement the game

27 Feb Milestone - Game is playable Review meeting - Show the implementation and discuss cheat detection

26 Feb - 10 Mar Research methods of cheat detection

11 Mar - 15 Apr Implement cheat detection

15 Apr Milestone - System is able to detect a cheating move

16 Apr Review meeting - Discuss the effectiveness of the cheat detection system and the final report

16 Apr - 10 May Writing final report

10 May Milestone - Final report submitted

Deliverables

My deliverables will include a final report and an implementation of the game Kriegspeil with a cheat detection system.

The final report will show my approach for solving this problem and an evaluation of how effective it was. I will include the strengths and weaknesses of the system and any potential improvements that could be made. I will include results from testing the system, including cases where the system was able to detect cheating and potential cases that the system could not detect.

References

[1] Chess Club 2017, ICC Help: Kriegspiel, available at: https://www.chessclub.com/user/help/Kriegspiel, accessed: 5 February 2018

[2] Max Irwin 2014, Kriegspiel, available at: <u>https://github.com/binarymax/kriegspiel</u>, accessed: 5 February 2018