**Notes form research for the design stage**

Notes from MIT lecture: <https://youtu.be/STjW3eH0Cik>

* For a long time chess was considered to complex of a problem for a computer to solve as it required some aspects of intelligence and strategy that was thought to be exclusive to humans
* For example philosopher Hubert Dreyfus said that computers would never be able to play chess in a paper he wrote in 1963. He subsequently was invited to play an early chess computer to which he lost. This showed that using artificial intelligence techniques, computers can exhibit the traits of intelligence needed to play chess well.
* It became clear in 1997 that computers could play chess better that people when a chess engine called Deep Blue beat the world champion. This showed that if they were able to understand how to play, their superior computational ability would allow them to win.

Possible methods for creating a chess engine:

Approach 1:  
The computer would analyse and describe the board in order to use strategy and tactics in order to decide the next best move.

This is the way that humans play chess and there is no obvious way to use computation to allow a computer to take this approach

Approach 2:

The use of if-then rules that correspond to every possible move made at any given board state in order to determine the appropriate move. For example is board is as such and the user moves a bishop from F8 to G9 then the computer should move a rook from A1 to B3.

In order to achieve this the computer must be able to recognise all possible legal moves available to it and then score them in order to select the best possible one. There is no obvious way to score a move other than to see if any of the moves take a high value piece form the opponent. Without looking ahead some number of moves the computer would be unable to see complex play patterns and would be unable to correctly select the move that doesn’t immediately take a piece but is better as it allow the computer to take a move valuable piece next turn.

Approach 3:

This is where the computer creates a tree that represents all the possible directions that the current chess game can take. The root node is the current game states. Arcs coming of it correspond to all possible legal moves and the nodes that they connect to represent the resulting game state after this move. This branching continues until the tree has grown very large and all of the leaf nodes are game states that represent a finished game with no further moves where the computer has either won or lost. The computer then needs to select moves that lead to the highest chance of it winning. For example the move with the most corresponding leaf nodes that show the computer winning.