Slide 1:

Hi, I am Jude Haris, and my project is called Go-LD. Go-LD is a software tool for the board game called Go.

Slide 2 :

The main aim of the project was to create a tool for Go players to improve their skills by solving Life and Death Go Problems. This aim includes creating a "computer" player within the program which can solve Go problems. The "computer" player should be able to play against the user and against itself.

Slide 3:

This is how a Go board looks like in the middle of a game.

Slide 4:

Quick introduction to Go. Go is a two player turn-based board game. Unlike most western board games, the stones, i.e. the pieces are played on the intersections not in between them. Players take turns placing a stone of their colour. A stone cannot be moved but can be captured and removed. The objective of the game is to surround as much territory as possible by creating walls with your stones. The game ends when both players agree no more good moves can be played and hence both players will pass. After the game ends the player with the most territory wins.

Slide 5 : The empty intersections around a stone or a group of stones are called the liberties. If the opponent is able to play on all the liberties, then the stone/ group of stones will get captured. Here the entire black group is captured by the white stone being placed at the top left.

Slide 7 : There are two other important rules , one stops a player from making suicidal moves and the Ko rules which restricts players from repeating the same board position over and over

Slide 10: Back to the program. Go-LD contains three main components: Play Mode, Editor Mode and the computer. It also consists of 3 screens , the title screen, the play mode screen and the editor mode screen.

Slide 11: Here is the title screen, simple, nothing fancy.

Slide 12 : This is the play mode screen. On the left we have the Go board with a problem loaded and, on the right, there are some buttons which controls the “computer” and allows the user to interact with the board.

Slide 13: And now here is the editor mode screen. As you can see the layout remains the same with the board on the left and some buttons on the right.

Slide 14 : The computer within Go-LD uses a Tree search to find the best move on the board. Initially the minimax search algorithm was used, later on it was adapted into the Alpha-Beta Pruning search and even later, Iterative Deepening was augmented on to the Alpha-Beta search.

The search uses of 3 types of heuristics : Board evaluation , Move ordering and Move Generation.

Slide 15 : The board evaluation function is used to determine the value of the board when the depth limit is reached during the search.

The pattern-based heuristic looks for occurrences of pre-defined patterns on the board and adds predefined values depending on the patterns found to the total board score. One player’s patterns are added negatively, and other players patterns are added positively.

The Liberty Count heuristic simply counts up all the liberties for both players and the difference is multiplied by a weighting factor and added on to the board score.

Slide 16 - During the search if the better moves are searched first then searching the other moves after will produce more Alpha-Beta cut-offs which in basically means less search space.

The Killer Move heuristics stores good moves for each depth of the search and during the move ordering process if these “killer” good moves appears as a valid move then they are arranged to the top of the list to be searched first.

Principal Variation is the best sequence of moves found by the search, during each iteration of the search the previous iteration’s Principal Variation is used as the first sequence of moves to be searched.

Slide 17 – The move generator generates a subset of moves from all the valid moves to limit the breadth of the search.

A simple distance-based heuristic is used to priorities moves close to the keystones. A keystone a normal stone which acts like a mark to define the objective of the problem. The two diagrams here show how moves are prioritised. Higher number indicates higher priority.

Slide 18: A pattern-based heuristic is also used to give move priority to moves which play on vital points of a pattern found. Here we can see the combination of the pattern-based heuristic and the distance-based heuristic.

Slide 19 : The evaluation phase consisted of beta testing and performance evaluation.

Beta tester were given the program along with some solvable problems and were asked to fill out a google form. Beta Tester found functionalities all to be working as expected. But some issues were raised about the user interface.

During the performance evaluation I gathered a collection of problems from an online library which the computer could solve. And then gathered all the relevant data such as time to solve and difficulty of the problem to analyse the computer. The results showed that Iterative Deepening was found to work better than fixed depth search on problem which took longer than 10 seconds and also the best depth was found to be 6 where difficult problems can be solved in a reasonable amount of time.

Slide 20 – Finally here is an example of problem solved by the computer. Here we can see group of black stones about to be captured by white. Black is able to live here by playing at a or b , any other move will allow white to capture black. Let’s see where computer plays as black.

Slide 21: Okay so the computer plays at A and note the red circle that appears indicating that white can’t play there because it would be suicide. The human player might respond like so to put the pressure back on black to play the correct move.

Slide 22 : …

Slide 23: And finally, the computer plays here and in doing so creates what is called 2 eyes which means the black group is safe from all threats and hence black wins this situation.

Slide 24: I was able to achieve my main goal and create a tool for Go players, there are many ways to develop it further , but I am very happy with the result.

Slide 25: Alright that is all, thank you for listening , any questions?

Slide 8: Life and death problems are a specific type of Go problem, which are solely concerned with the outcome of a certain group of stones. Whether they live or die. Getting better at life and death is, one of the best ways to get better at Go in general.

Slide 9: Here are 2 examples of how life and death problems look like.

Slide 24: Here the top variation shows the one I just described and the bottom one is another possible move the human could have played and how the computer would have responded.

Move Gen: If any are found, then the predefined value of the pattern is added on to the total board score. One player’s patterns are added negatively, and other players patterns are added positively.