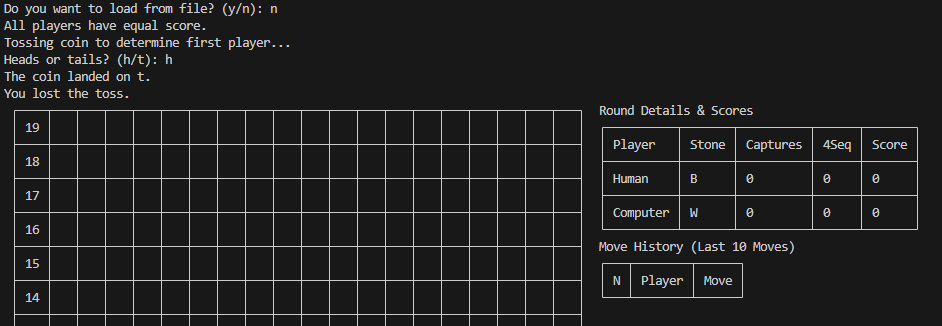
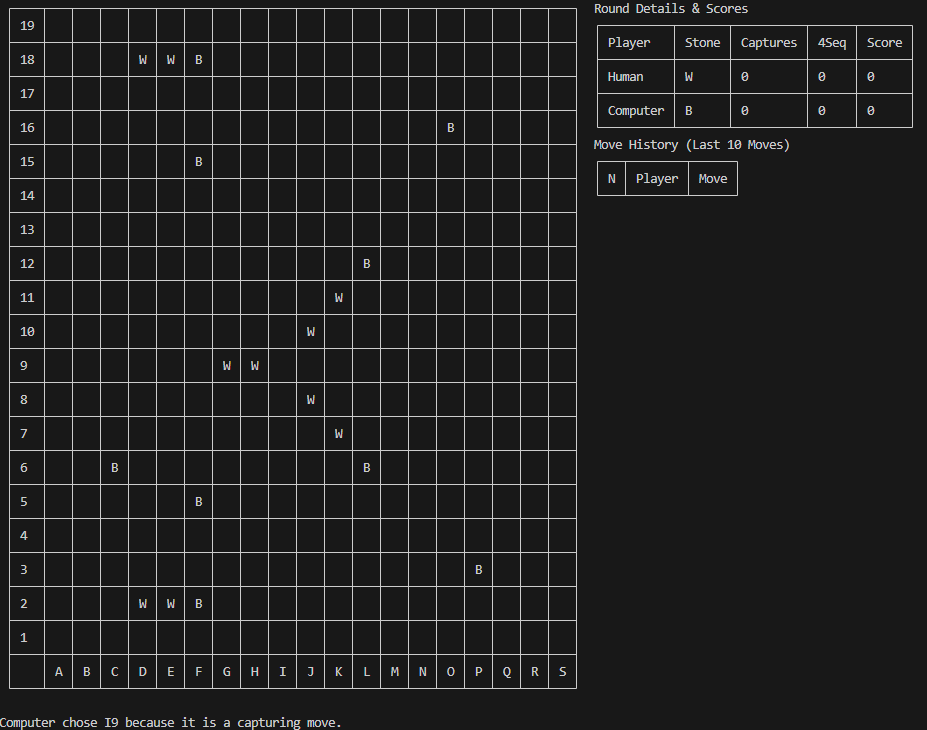
Submission Manual

# Screenshots

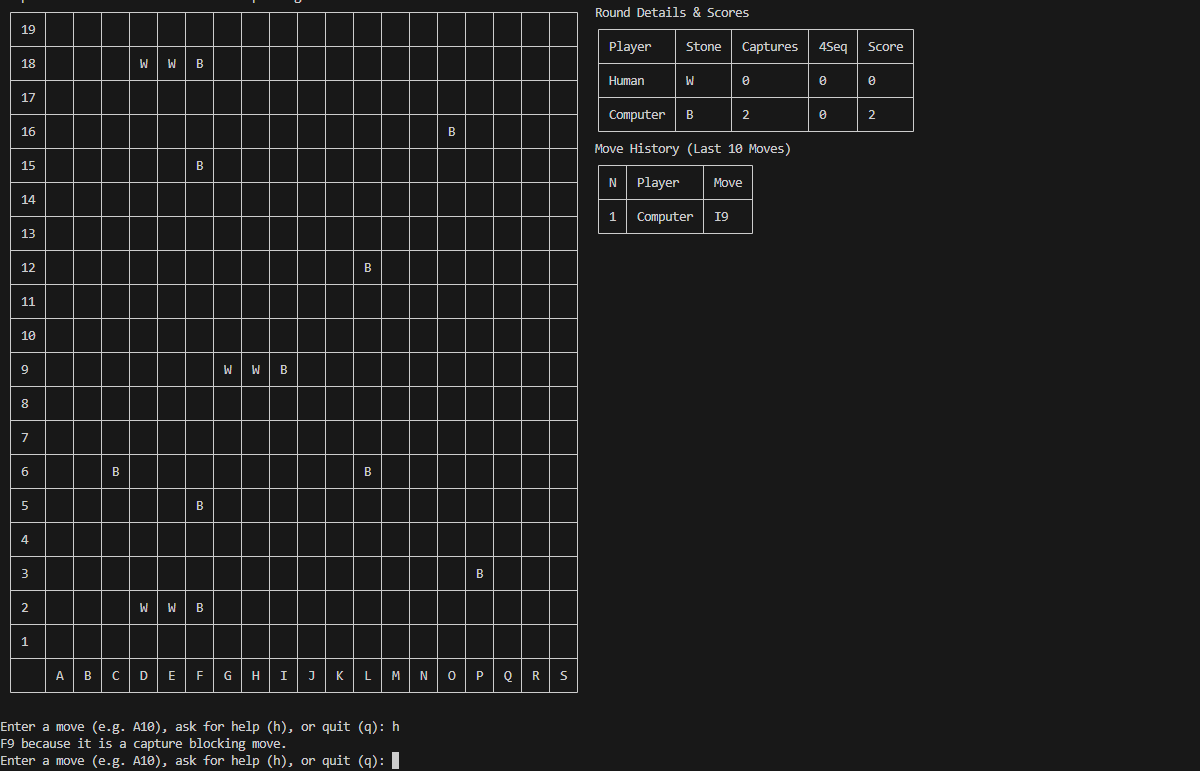
## First player of the round being determined



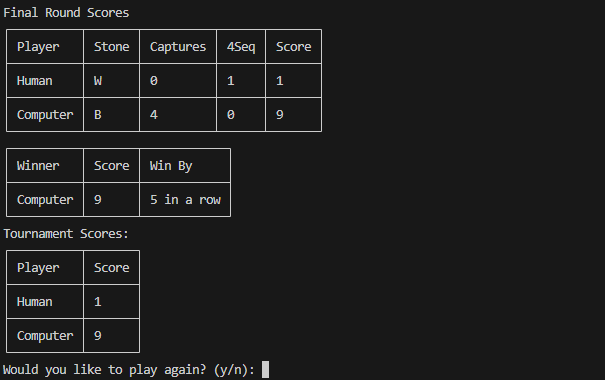
## Computer’s move being explained



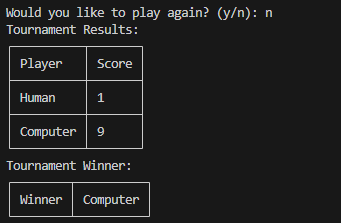
## Computer providing help



## Winner of the round being announced



## Winner of the tournament being announced



# Bug Report

* The program only accepts valid serial files. For example, if there are 10 white stones and 15 black stones (like in case 4), the program will tell the user to input a valid serial file. I thought this was a feature because it validated the input file, but sadly, it is not.

# Bugs fixed from Demo

* In the demo, the second move of the game was three intersections away from the center. It should have been the second move of the first player. This has now been fixed.
* The scoring algorithm was never incorrect. It only showed the wrong score because I had modified the serial file with two more pairs to account for the missing stone in the board so that it would load a valid file.

# Description of Classes

| **Class** | **Description** |
| --- | --- |
| Board | The board used for the game |
| Experimental  Board | Inherits from Board -- Allows the user to make invalid moves; Used for anlayzing moves |
| BoardDisplay | Renders board in a tabular format with positions labelled |
| GameOver | Inherits from std::exception; Thrown to indicate that game is over |
| GameDrawn | Inherits from GameOver |
| GameWon | Inherits from GameOver |
| InvalidMove | Inherits from std::exception; Thrown to indicate that an invalid move is being made on the board |
| Quit | Inherits form std:;exception; Thrown to indicate that player had decided to quit |
| MoveAnalysis | Provides analysis for a move, for example, if the move is a winning move, capturing move, capture blocking move, win blocking move, etc |
| Player | Represent a player -- virtual |
| Computer | Inherits Player |
| Human | Inherits Player |
| Position | Represents a position in the board; Class used because it contains methods to generate string represantations and calculate distances |
| Roster | Stores the users and the scores in a tournament |
| Round | Represents a round in the game |
| RoundDisplay | Renders and display the board, details, and move history while a round is played |
| Serial | Parses and generates serial strings |
| Strategy | Decides on which move to play -- seperated into a class from player so that the same player can use multiple types of strategies |
| Tournament | Represents the Tournament |
| Tournament  Announcement | Renders and displays tournament results in a tabular format |

## Data Structures

| **Name/ Structure named defined using “using”** | **Description** |
| --- | --- |
| BoardSequence | Used to store full or partial parts of the board; evaluates to std::vector<char> |
| MoveHistory | Used to save the history of moves of a player; evaluates to std::vector<std::pair<Player\*, Position>> |
| analysis | A tuple to store the analysis of a move -- evaluates to  std::tuple<int, int, int, int, int, int, Score, Score, int, std::string>  Contains information about whether it is a winning move or not, whether the opponent can win, the differential in score after the move, the differential in score if opponent makes the move, the capture differential, opponent capture differential, and whether opponent can capture two plies ahead, the psedo score, the negative distance the center and the position  Initial plan was to create a class, but I ran out of time; A vector of this tuple is then sorted to find the most optimal move |
| priority queues | Usesd to store the move and their score for different types of moves; this was later discarded for one single tuple instead of having multiple queues and if blocks |
| StrategicMove | evaluates to a std::pair<Position, std::string>; the second string is the rationale for the move |
| map | Used to player to stones, stones to number of captures, and player to score |
| vector | Used for storing various other data structures -- no arrays used |
| set | Used to return list of moves -- for example, available moves, winning moves, capturing moves, etc |
| Table | Used to represent board, scores, and history for display -- from tabulate |
| graph | Not explicitly used in the code, but MoveAnalysis and Board use graph algorithms for various methods |

# Project Log

| **Date** | **Time** | **Messge** |
| --- | --- | --- |
| Sep 27 | 4:27 PM | Fix rule |
| Sep 27 | 3:35 AM | Read serial file in the order it is displayed |
| Sep 26 | 11:50 PM | Fix bug |
| Sep 26 | 11:43 PM | Fix bug |
| Sep 26 | 11:26 PM | Fix bug |
| Sep 26 | 5:43 PM | Refactor |
| Sep 26 | 5:27 PM | Handle multiple winning moves |
| Sep 26 | 12:05 PM | Refactor and update |
| Sep 26 | 11:57 AM | Fix bug |
| Sep 26 | 11:40 AM | Fix bug |
| Sep 26 | 11:00 AM | Use LF in serial files so that it works on linux |
| Sep 26 | 10:52 AM | Enable compilation in linux |
| Sep 26 | 10:32 AM | Check ahead two moves for capture safety |
| Sep 26 | 9:36 AM | Reformat computer's message |
| Sep 26 | 8:26 AM | Consider distance from center when deciding move |
| Sep 26 | 7:39 AM | Refactor |
| Sep 26 | 7:31 AM | Refactor to use a tuple instead of multiple priority queues |
| Sep 26 | 4:50 AM | Improve algorithm |
| Sep 26 | 4:40 AM | Refactor |
| Sep 26 | 4:30 AM | Show board while announcing Round winner |
| Sep 26 | 3:43 AM | Enable output serialization |
| Sep 26 | 1:55 AM | Add stones to fix error |
| Sep 26 | 1:50 AM | Fix number of stones in case 3 serial |
| Sep 26 | 1:50 AM | Refactor error message |
| Sep 26 | 1:41 AM | Raise exceptions when serializing |
| Sep 26 | 12:33 AM | Enable loading from file |
| Sep 26 | 12:11 AM | Add provided serials |
| Sep 25 | 10:17 PM | Add method to check if round is over |
| Sep 25 | 10:13 PM | Enable initializing round with a board |
| Sep 25 | 8:41 PM | Refactor |
| Sep 25 | 7:43 PM | Refactor and correct |
| Sep 25 | 6:47 PM | Check if opponent can capture |
| Sep 25 | 6:00 PM | Check if opponent wins/scores in a certain place |
| Sep 24 | 8:30 AM | Handle captures when setting stones in the ExperimentalBoard |
| Sep 24 | 8:06 AM | Make computer use strategic moves |
| Sep 24 | 7:38 AM | Refactor |
| Sep 24 | 7:32 AM | Refactor |
| Sep 24 | 7:22 AM | Test and implement is\_opponent\_winning\_move |
| Sep 24 | 6:32 AM | Refactor and update algorithm |
| Sep 24 | 5:51 AM | Make rationale clearer |
| Sep 24 | 5:40 AM | Implement announcing tournament scores after round end |
| Sep 24 | 5:17 AM | Implement round score announcement at round end |
| Sep 24 | 4:29 AM | Implement getting help |
| Sep 24 | 4:14 AM | Fix bug |
| Sep 24 | 4:12 AM | Fix bug |
| Sep 24 | 3:49 AM | Enable human vs human to test strategies |
| Sep 24 | 3:48 AM | Refactor |
| Sep 24 | 3:23 AM | Start implementing strategy |
| Sep 24 | 2:45 AM | Add win\_delta |
| Sep 24 | 2:42 AM | Refactor |
| Sep 24 | 2:37 AM | Add is\_losing method |
| Sep 24 | 1:21 AM | Refactor |
| Sep 24 | 1:17 AM | Refactor |
| Sep 24 | 1:16 AM | Implement history table display |
| Sep 24 | 12:38 AM | Implement MoveHistory for Round |
| Sep 24 | 12:34 AM | Implement Scores table |
| Sep 23 | 11:58 PM | Fix RoundDisplay |
| Sep 23 | 11:37 PM | Format first and last columns |
| Sep 23 | 11:30 PM | Format bottom and top rows |
| Sep 23 | 10:31 PM | Refactor |
| Sep 23 | 10:23 PM | Use fmt/format instead of format for cross compatibility |
| Sep 23 | 8:52 PM | Start implementing RoundDisplay |
| Sep 23 | 6:13 PM | Refactor |
| Sep 23 | 5:31 PM | Display tournament results after end |
| Sep 23 | 4:31 PM | Test algorithm to get pseudo score that's used for making moves |
| Sep 22 | 8:39 PM | Refactor and fix algorithm to get all diagonals |
| Sep 22 | 5:17 PM | Refactor |
| Sep 22 | 4:23 PM | Refactor |
| Sep 22 | 3:36 AM | Remove display to reduce test duration |
| Sep 22 | 3:33 AM | Test and implement awarding points for captures |
| Sep 22 | 3:26 AM | Award 5 points for sequences of at least 5 in a row |
| Sep 22 | 3:19 AM | Test getting score for 5 consecutive stones |
| Sep 22 | 3:04 AM | Test and implement getting score for 4 consecutive stones |
| Sep 22 | 2:36 AM | Fix typo |
| Sep 22 | 2:23 AM | Refactor |
| Sep 22 | 2:21 AM | Refactor |
| Sep 22 | 2:20 AM | Rename StoneSequence to BoardSequence |
| Sep 22 | 2:16 AM | Refactor literals to symbolic constants |
| Sep 22 | 2:02 AM | Refactor |
| Sep 22 | 1:53 AM | Refactor |
| Sep 22 | 1:34 AM | Test and implement getting stone sequences for a particular stone |
| Sep 22 | 12:15 AM | Test and implement getting stone sequences from a board sequence |
| Sep 21 | 11:18 PM | Refactor comments |
| Sep 20 | 6:21 PM | Refactor |
| Sep 20 | 5:59 PM | Refactor helpers header and cpp so that only .h can be included |
| Sep 20 | 5:46 PM | Make computer play random moves |
| Sep 20 | 5:15 PM | Add helper function to select random element from a container |
| Sep 20 | 5:36 AM | Get and validate move from human |
| Sep 20 | 4:51 AM | Implement Round |
| Sep 20 | 3:36 AM | Prepare executable |
| Sep 19 | 12:06 AM | Refactor |
| Sep 18 | 11:53 PM | Test and implement checking if move is a capturing move |
| Sep 18 | 9:50 PM | Prevent test from taking too long to run |
| Sep 18 | 9:44 PM | Test and implement checking if the next move is a win blocking move |
| Sep 18 | 9:23 PM | Try implementing win blocking move |
| Sep 18 | 7:18 PM | Update const qualifications |
| Sep 18 | 7:15 PM | Test and implement analysis of first and winning moves |
| Sep 18 | 6:45 PM | Remove unnecessary semi colons |
| Sep 16 | 4:22 AM | Update rubric with completions |
| Sep 16 | 3:42 AM | Add rubric template |
| Sep 16 | 3:39 AM | Test and implement winning by capturing 5 or more pairs |
| Sep 15 | 4:59 AM | Test and implement winning by 5 in anti diagonal |
| Sep 15 | 4:46 AM | Test and implement winning by 5 in main diagonal |
| Sep 15 | 3:57 AM | Test and implement winning by 5 in col |
| Sep 15 | 3:46 AM | Test and implement winning by row |
| Sep 15 | 2:49 AM | Remove display |
| Sep 15 | 2:44 AM | Test multiple captures |
| Sep 15 | 2:05 AM | Test and implement handling of capturing up and left in diagonals |
| Sep 15 | 1:58 AM | Test and implement handling of capturing down and left in diagonals |
| Sep 15 | 1:53 AM | Test and implement handling of capturing up and right in diagonals |
| Sep 15 | 1:43 AM | Test and implement handling of capturing down and right in diagonals |
| Sep 15 | 12:22 AM | Refactor |
| Sep 15 | 12:13 AM | Test and implement handling of capture above |
| Sep 15 | 12:11 AM | Test and implement handling of capture below |
| Sep 14 | 6:09 AM | Test and implement handling of capture to the right |
| Sep 14 | 5:49 AM | Test and implement handling of capture to the left |
| Sep 14 | 5:33 AM | Update how board is displayed |
| Sep 14 | 4:14 AM | Implement display using the tabulate and fmt libraries |
| Sep 13 | 10:12 PM | Add fmt library for formatting strings |
| Sep 13 | 3:43 AM | Test and implement making moves on the board |
| Sep 13 | 2:11 AM | Test getting the third moves |
| Sep 12 | 2:47 AM | Test and implement getting available moves |
| Sep 12 | 1:02 AM | Ignore IDE files |
| Sep 12 | 12:20 AM | Test and implement getting players from roster |
| Sep 11 | 10:03 PM | Reformat |
| Sep 11 | 9:50 PM | Test and implement Roster |
| Sep 11 | 4:15 AM | Add header guard and reformat |
| Sep 11 | 3:07 AM | Test and implement getting the turn of the next player |
| Sep 11 | 2:45 AM | Test and implement getting total number of stone played on the board |
| Sep 11 | 2:09 AM | Refactor |
| Sep 11 | 2:06 AM | Test and implement get\_no\_stone\_on\_board |
| Sep 11 | 1:57 AM | Refactor |
| Sep 10 | 8:24 AM | Refactor |
| Sep 10 | 7:48 AM | Test and implement getting diagonals |
| Sep 10 | 6:56 AM | Test and implement get\_col method |
| Sep 10 | 6:52 AM | Test and implement get\_row method |
| Sep 10 | 6:24 AM | Rename |
| Sep 10 | 6:22 AM | Refactor |
| Sep 10 | 5:42 AM | Rename test suites and cases |
| Sep 10 | 5:38 AM | Rename |
| Sep 10 | 5:37 AM | Refactor to Position class |
| Sep 7 | 7:51 PM | Comment and refactor |
| Sep 5 | 8:25 PM | Test and implement captured pair data within Board |
| Sep 5 | 4:52 PM | Correct row/column switch |
| Sep 5 | 3:15 PM | Test and implement getting board from serial |
| Sep 5 | 3:11 PM | Fix error in provided serial file |
| Sep 5 | 2:25 PM | Refactor into fixtures |
| Sep 5 | 1:59 PM | Fix bug |
| Sep 5 | 7:50 AM | Test and implement getting scores and captured pairs from serial |
| Sep 5 | 6:41 AM | Compile library as static to prevent linker errors |
| Sep 5 | 4:01 AM | Start working on SerialFile class |
| Sep 5 | 3:27 AM | Add serial file from provided format |
| Sep 5 | 3:20 AM | Add guidelines.md |
| Sep 5 | 3:09 AM | Add description.md |
| Sep 5 | 3:05 AM | Add tests and src structure |
| Sep 2 | 12:05 AM | Update .gitignore |
| Sep 1 | 11:59 PM | Add CMakeLists and check if GoogleTest is working |
| Sep 1 | 11:51 PM | Add .clang-format |
| Sep 1 | 11:51 PM | Ignore build files |
| Sep 1 | 7:57 PM | Initial commit |