4-player Ultimate TicTacToe Coding Project Summary

Group 26- Cole Pearson, Nausherwan Tirmizi, Brian Kopec, Harshal Patel

4 player ultimate TicTacToe is a web based multiplayer game that is based off of the original tic tac toe game. This game allows players to play together in a 3x3 grid of TicTacToe games which makes the game a lot more challenging. Some of the more complex applications in 4 player ultimate TicTacToe are three levels of AI to suit different player difficulty levels, a database for leaderboards and player information, an implementation of a client-server model, and a chat to allow communication between players and a server.

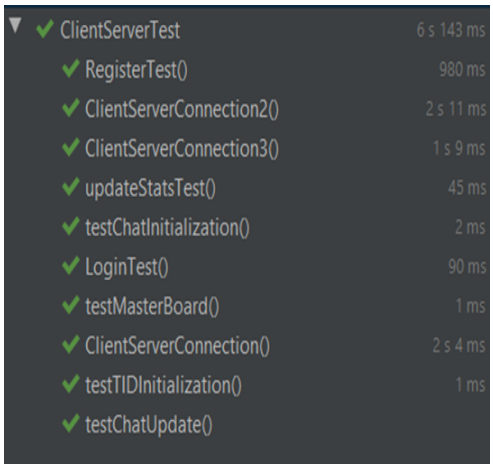
When the player starts the game, they will be asked to login or create an account. This is because player information, like wins/losses/points is stored in a SQL database which adds to the competitiveness of the game. After they log in, they will be able to join a server which is already running.

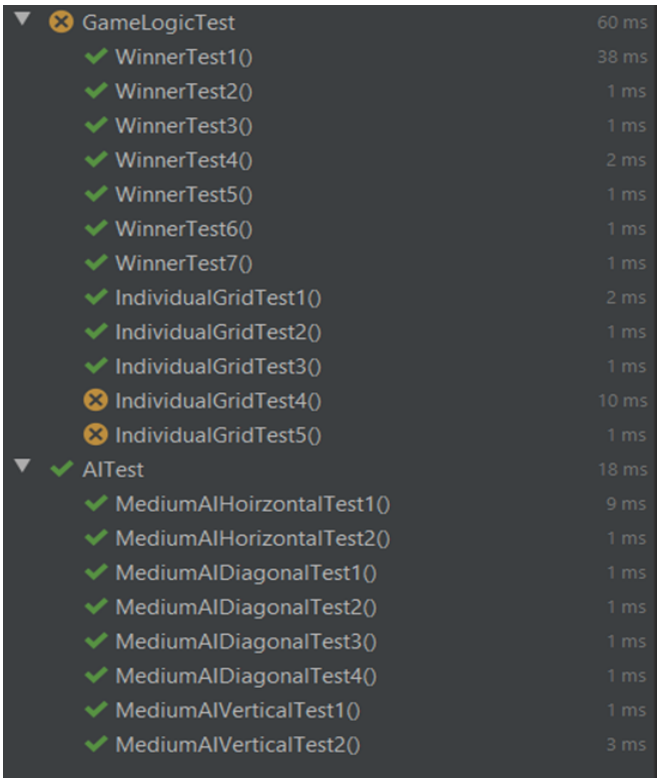
All of the logic is stored in a GameLogic class. When a player makes a move, this will check if a point has been scored and if there is a winner. If a point is scored, the UI is updated for each player with the symbol of the player who made the point. If a player has won the game, then the UI will be updated to a final scene that shows who won the game with an option to play again or exit.

Testing was done using the Junit testing framework for Java applications. One of the major issues running the Junit tests were that it does not support JavaFX and leads to many errors. To bypass this we have to create the test class as an Application and create an instance of a JavaFx Panel before the test functions. This allows us to include JavaFX elements into our test functions.

The first release focused on releasing a functional base game where players could connect and start a tic tac toe game. The second release built on quality of life improvements such as GUI improvements, chat. Alongside that AI functionality so people could play with an incomplete lobby.

The major functionalities that we tested were Client Server Connections, Database functionality, Chat messages, Turn Identification, Game Logic, and AI. All but two tests in the GameLogic passed. The Client and Server connections tested if data is sent back and forth correctly. The Chat and Turn Identification test whether they are being constructed correctly. The GameLogic tests check whether the isPointScored2 and isWinner2 function is functioning properly, and check for winners and points. The AI tests check whether the medium AI is correctly blocking a win from occurring.





Known and fixable issues have been inspected and resolved. These include AI bugs, GUI bugs, SQL database implementation and game logic functionality. Outstanding issues – including the lack of a hard AI, password encryption, and in-game username display – have been documented and their solutions have been drafted.

Overall the project was successful. Team members completed their individual tasks and everything was done efficiently and in a timely manner.