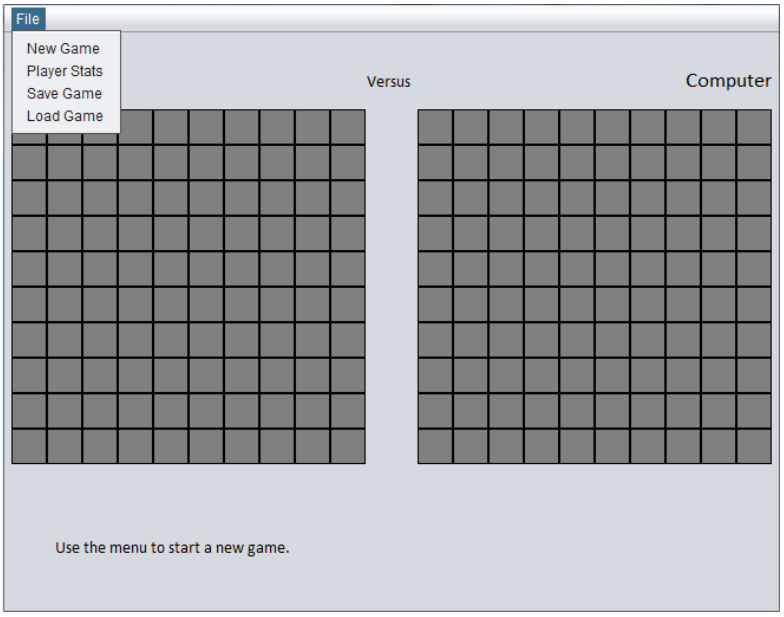
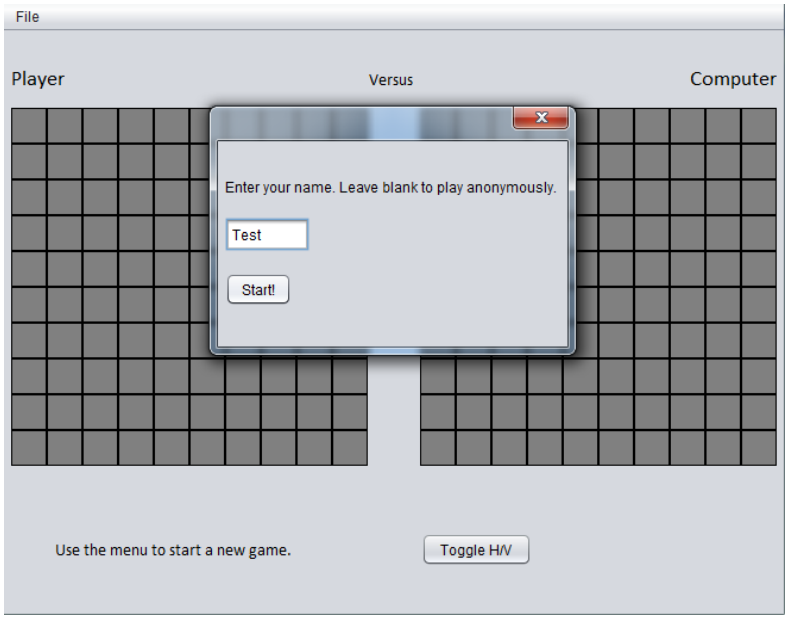
Battleship Gameplay

Pregame



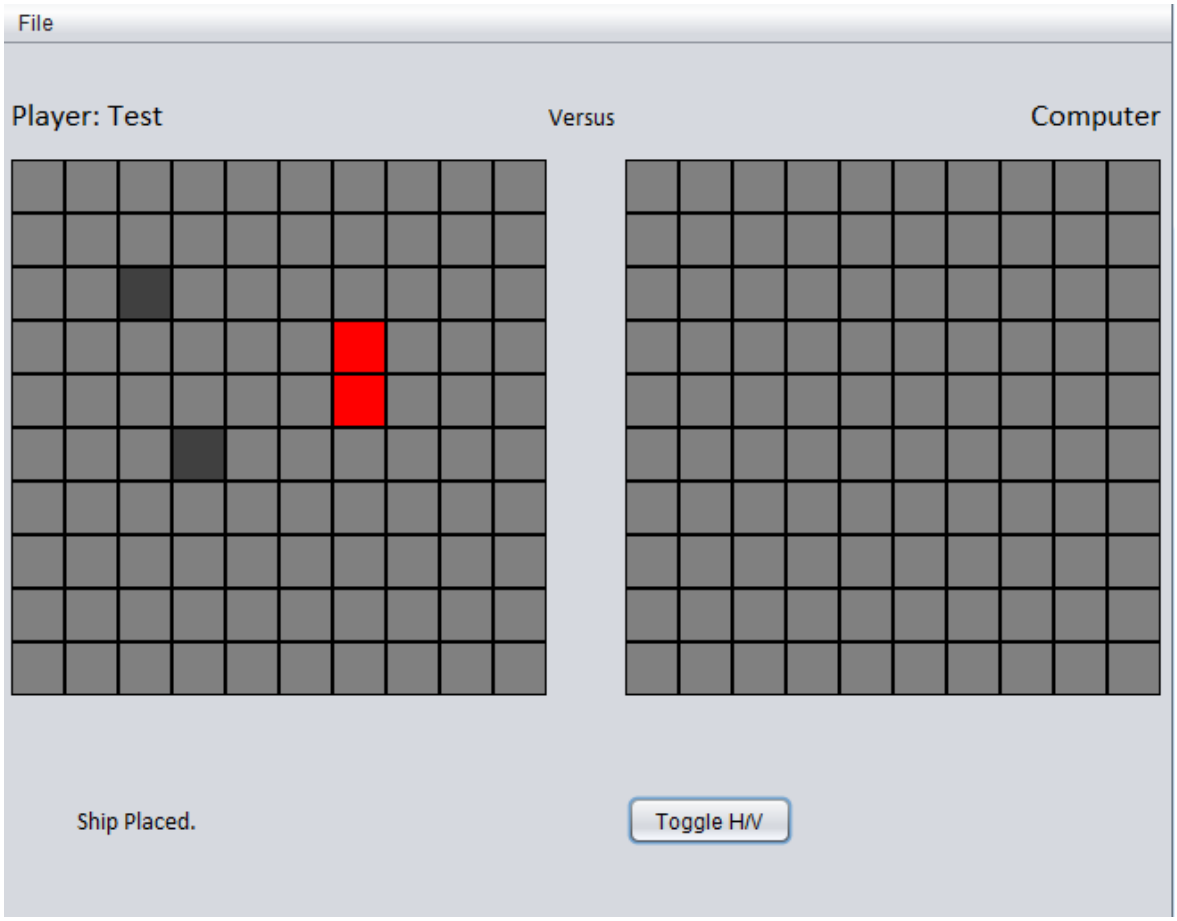
Select new game from the File menu.

New Game



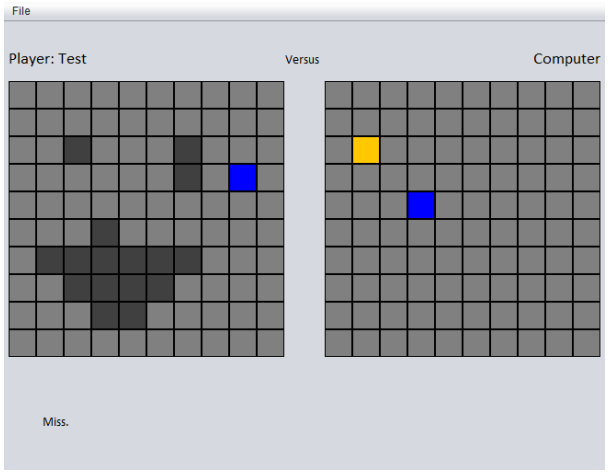
Enter a name, or leave blank to remain anonymous (no stats recorded).

Game Start

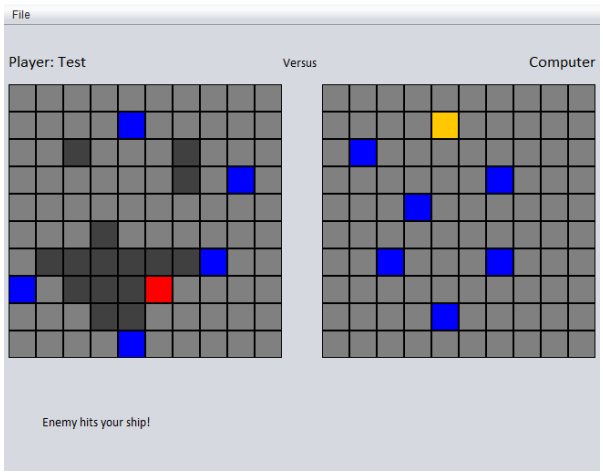


Click to place ships. Use Toggle H/V to change between placing vertically and horizontally. Cursor highlights where the ship will be placed before click.

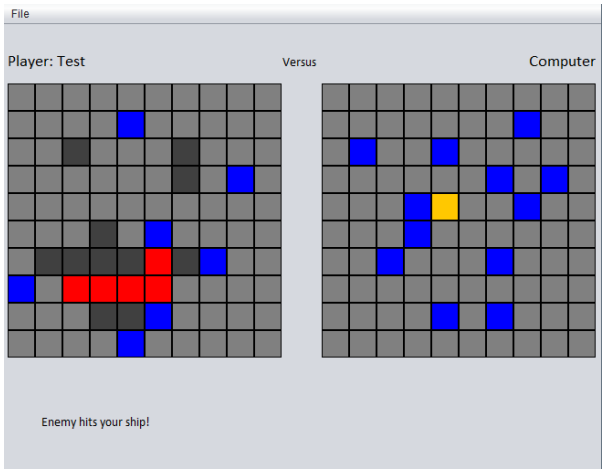
Attack Phase



After missing first shot.

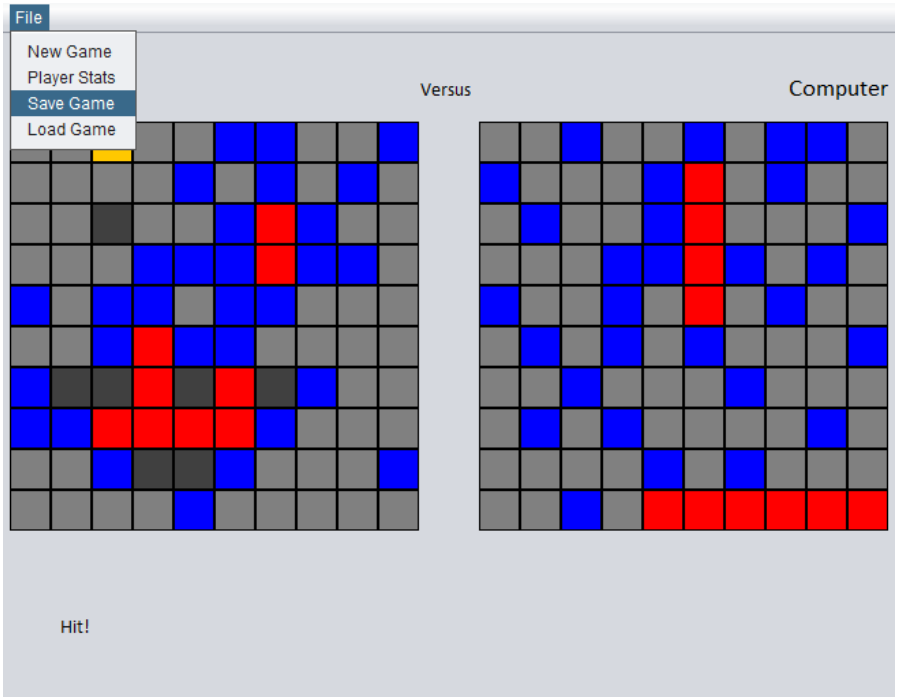


CPU hits player.



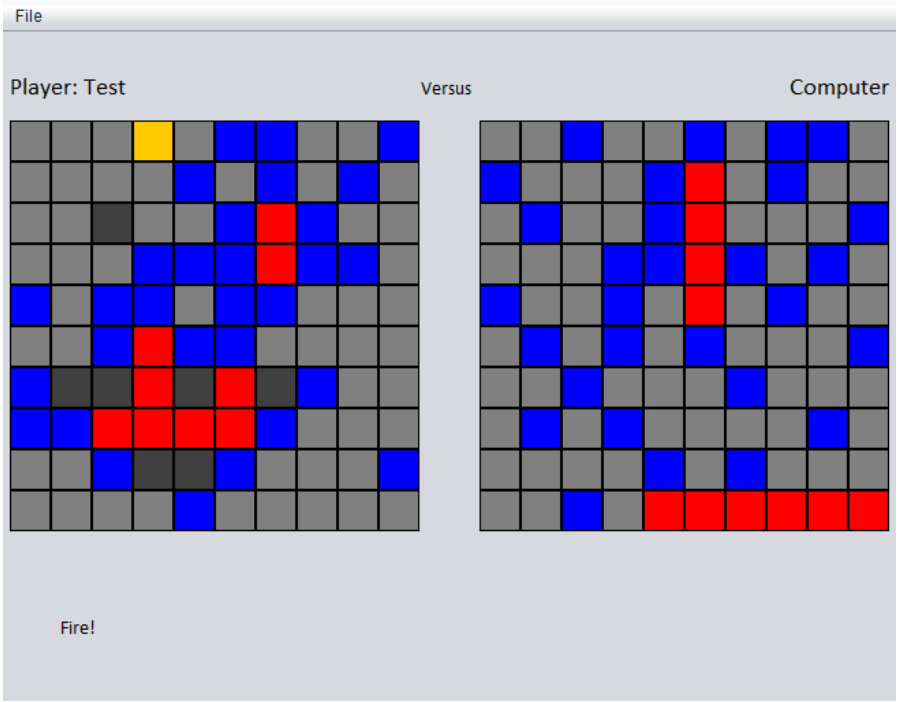
CPU successfully guesses subsequent hits.

Save



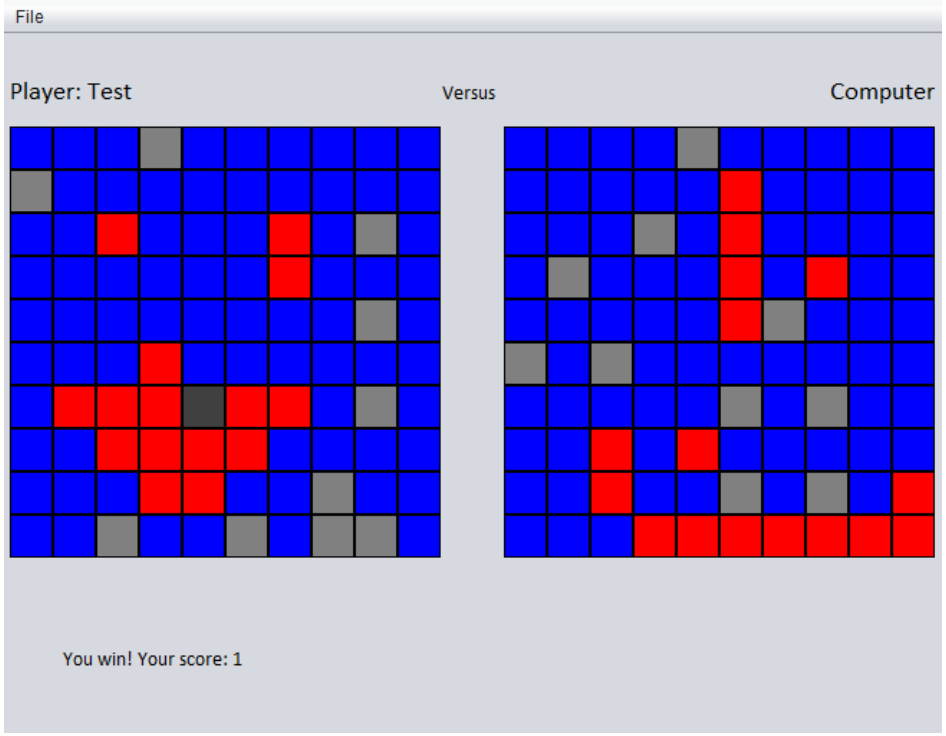
Saving mid-game.

Load



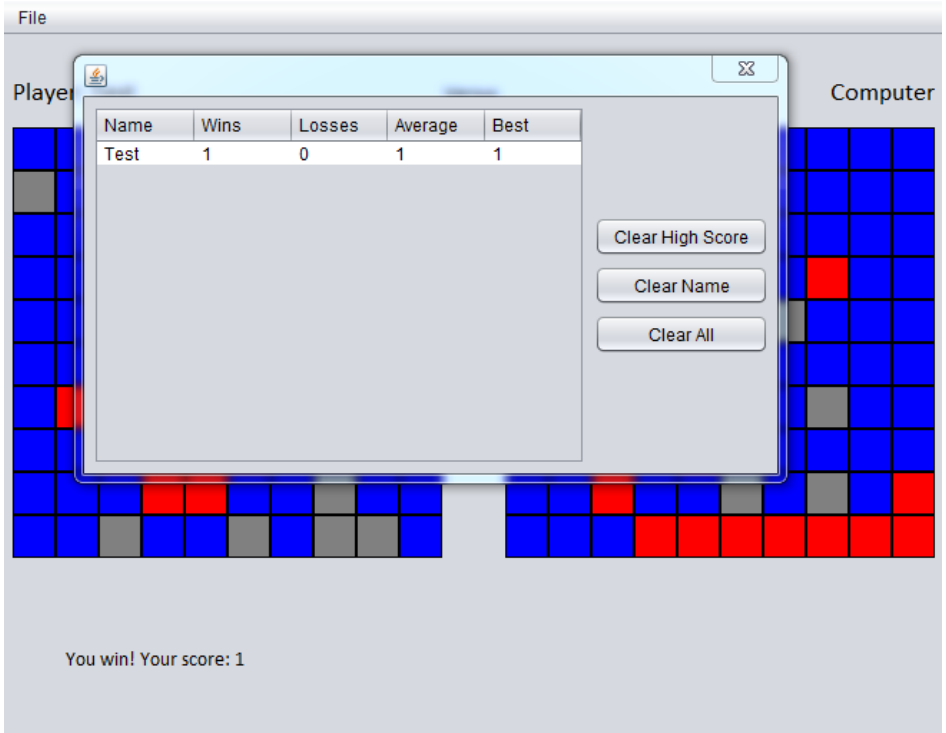
After closing game, then selecting Load from the File Menu.

Game Win



Win triggered by hitting the last square of the CPU's last ship.

Stats



Displaying stats (File menu) after the previous game.

Design Notes

Graphical User Interface

The GUI for the game was designed using the NetBeans IDE GUI builder. The reason for this is that constructing GUIs by hand is extremely tedious and can detract from other design elements. The NetBeans builder allows you to dynamically add, resize, and reposition components; the generated code is automatically updated. While the generated code can be messy and verbose, it is easy to maintain and remains distinct from the user-generated code.

AI

The AI proved to be more complex than initially expected. We approached its design by trying to describe the way we play the game in natural language. For example: Click a random square. If it hits, try its neighbouring squares until a ship is sank. We then transformed it into an algorithm that uses a random number generator, and static variables to “remember” its state for the next turn.