# Kono Implementation in Java & Android

#### Philip Glazman

# Bug Report

This program does not have any known bugs.

# Feature Report

All the features listed in the rubric are implemented as described.

# Description of Classes

### Classes

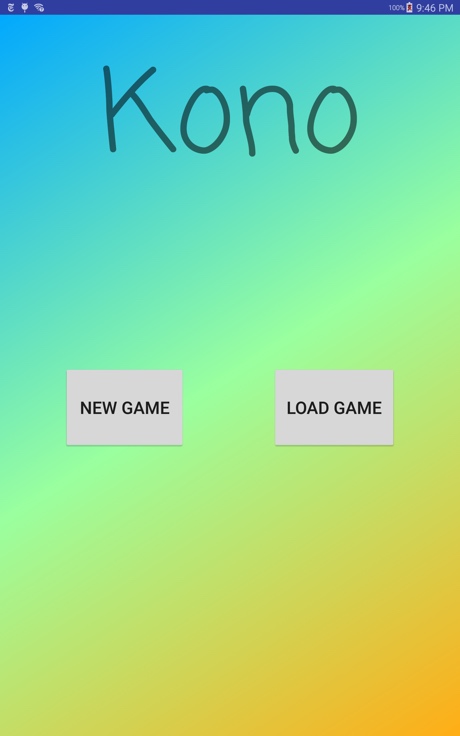
* Board
  + Maintains data model for board.
  + Responsible for checking validity of board moves.
  + Includes selectors for accessing the data model.
* Player
  + Provides basic functionality for getting row and column coordinates of player move.
* Computer
  + Extends the Player class.
  + Responsible for the decision-making of the computer and for help mode.
* Human
  + Extends the Player class.
  + Responsible for storing the initial and final coordinates for the human player’s turn.
* Round
  + Maintains the logic of the round.
  + Determines if the winning conditions of the round are met and calculates the final round scores.
  + Holds the player colors and scores.
* Tournament
  + Responsible for storing round number, computer tournament score, and human tournament score.
* GameConfiguration
  + Responsible for starting a new game.
  + Manages serialization for loading and saving game.
* TupleCoordinates
  + Tuple data structure used by computer for storing the coordinates, strategy, and direction of its move.
* StartActivity
  + GUI components for first launcher screen.
  + Allows user to start new game or load game.
* NewGameActivity
  + GUI components for starting a new game or loading a file.
* MainActivity
  + Responsible for displaying board and providing main game functionality.
* EndActivity
  + Shows the final scores of the game to the user.
  + Provides user feature to quit or start new round.

# Log

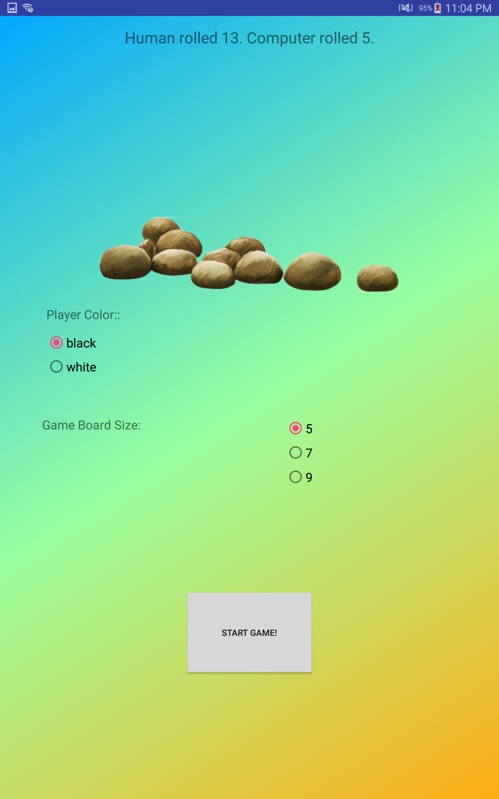
|  |  |  |
| --- | --- | --- |
| Description | Timestamp | Duration |
| Worked on the main board layout. | 3/20/2018 10:27pm | 3 hr |
| Added activity menu for starting game. Added dice roll feature, and user input for selecting board size and player color. Added radio buttons for selecting board size and player. Dice is rolled randomly. | 3/21/2018 10:25pm | 3 hr |
| Implemented board class responsible for data model of board. Created basic implementation of GUI for main board activity. | 3/22/2018 4:18pm | 2 hr |
| Board pieces correctly set up. Board GUI components correctly displays the board data model. Players alternate at each turn. Added input validation for player moves. Board refreshes after each turn. Human user can select pieces and move to valid locations. Added computer class and implemented basic outline of computer strategy. | 3/23/2018 11:30pm | 9 hr |
| Completed computer strategy. Added GameConfiguration class and implemented serialization for loading and saving files. Fixed issue where computer would not recognize all of its own pieces. Fixed issue where computer could not detect which piece to block. | 3/25/2018 5:48pm | 6 hr |
| Implemented tournament control. Fixed bugs with serialization where file would not correctly save. Fixed bugs with serialization where loaded file would not detect player colors. Implemented help mode. | 3/27/2018 7:45pm | 4 hr |
| Added improvements to the GUI and fixed constraint layout issues. | 3/28/2018 7:45pm | 1.5 hr |
| Fixed bugs where computer would not recognize its own pieces when playing as black. Fixed tournament control so that points awarded are the difference in player scores. Added numbered columns and rows to table layout. Fixed issue where computer would not select its random color after winning the previous round. Fixed issue where the number of pieces for each player was not correctly counted. Added feature for computer to quit the game under certain conditions. Refactored existing code. | 3/29/2018 3:49pm | 5 hr |

# How to Run the Program

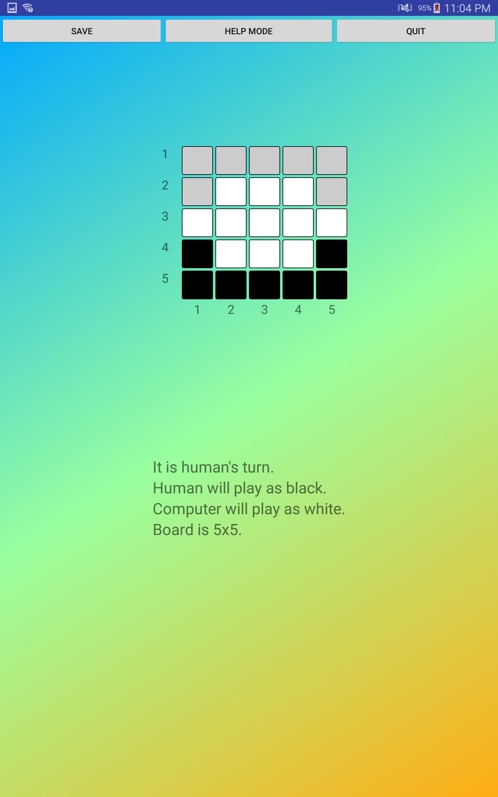
## Launcher Screen.

**

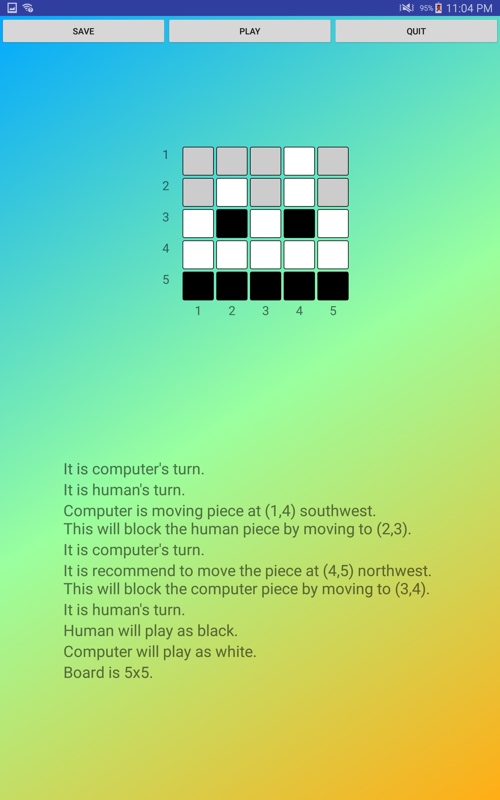
## Game Configuration Screen.



## New Game Board.



## Playing a Game.



## End Game.

