* **Activities**

1. WelcomeActivity

Displays the home page which includes a play button. When pressed the button directs to GameActivity.

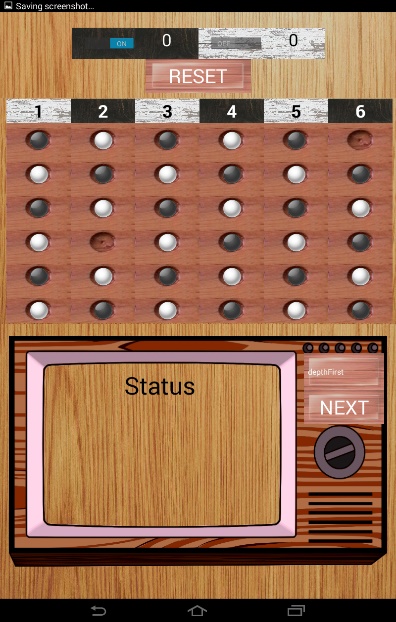
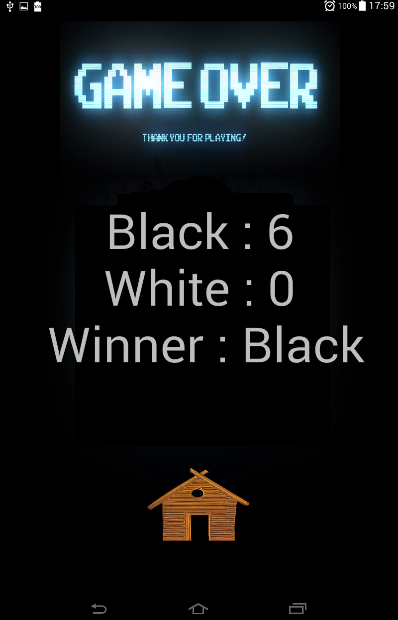
1. GameActivity

Displays the board, switch and score and updates them when changes are made.

When no more moves left and both player pass their turn activity ends and directs to ResultActivity while passing the latest score of black and white player.

1. ResultActivity

Displays the score and the new game button which directs to new Game Activity when pressed.



* **Model**

1. Player

Class for player object which holds score and type (either black or white).

1. Board

Class for board object which sets constants for black, white and blank stones. It also stores the size of the board and the array to hold all the stone types.

1. Game

Class for Game objects which uses player and board objects to define the movement of stones in the board. It also has variable p\_turn to store the turn of the player. It is responsible to validate the move and carry out the move.

All the AI algrothim used till date are implemented in the Game class.

1. Depth First Search

addValidDestTree() method uses Depth First Search algorithm to add valid moves as a node in a tree.

1. Breadth First Search

breadthFirstAlgorithm() method uses queue to implement the algorithm.

1. Best First Search

bestFirstAlgorithm() method uses priority queue to implement the algorithm.

1. Branch and Bound

branchBoundAlgorithm() method uses Best First Algorithm and uses score as constrain to cutoff the tree.

1. Min Max and Alpha Beta pruning

minMaxAlgorithmn() method uses recursion to find the best move for given ply. It uses Breadth First Algorithmn to get list of possible moves for given player in given ply of the recursion.

1. Tree

Class to store possible moves in a tree of linked list.

* **Bug Report**

**The app crashes when loading a game file which does not exist.**

* **Feature Report**

1. Player can select the size of the board between 6\*6, 8\*8 and 10\*10
2. Computer makes move automatically when pressed the Next Button
3. User can now save and load a game file.

* **Log**

1. 3/15/2018
   * Tried to solve the problems from Project 2
   * Solved the problem of not being able to load multiple files (1 hour)
   * Tried to solve the self-block problem, had issues in nextMoveAlgorithm() in Game.java. I decided to make changes in how the algorithm finds the possible moves. (2 hour)
2. 3/16/2018

* Solved the self-block problem with the solution that to get possible valid moves position of stones are actually changed in the board but restored again after finding the valid move. (3 hours)

1. 3/18/2018

* Made changes in LoadGame() to immplement loading files with user input file name. Used dialogs to ask user to input the file name.(3 hours)

1. 3/19/2018

* Used dialog to input the size of the board when NewGame buton pressed in WelcomeActivity.java (30 mins)
* Tried to finish serlization tasks and cleaning up the recently added code. (1 hour)

1. 3/21/2018

* Attempted to write most of the code for minMaxAlgorithm().
* Came up with different problems when using recursion. However solved most of it using global boolean values which were used as flags. (2 hours)

1. 3/24/2018

* Major issue when backing up the recentlt made changes I accidently replaced the main Konane folder with previous version. Had to redo the changes made during last 2 sessions. (3 hours)

1. 3/25/2018

* Immplemented the minMax algorithm (1 hour)
* Also immplemente alpha beta pruniung (1 hour)

1. 3/27/2018

* Documentation (30 min)