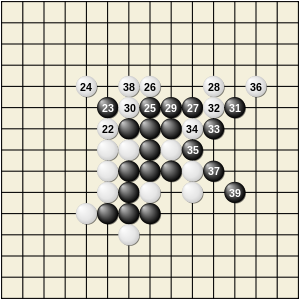
**Final Project: Gomoku**

Diego D’Gyves and Matthew Theisen

**Overview**

Gomoku is a traditional Japanese board game, considered by many to be one of the world’s greatest strategy games. Played on a 15x15 grid, two players (typically black and white) alternate placing stones on intersections of the grid lines. The game is won by the first player to place five stones of their color in a row horizontally, vertically, or diagonally.

The app will allow for selection of either local two-person multiplayer, or a single-player game against an AI. Throughout the game the user will have the option to save the game state (and given the option to either begin a new game, or resume a saved one). Once the game is completed, the option will be given to share a screenshot of the final result with one of the user’s contacts.



(Gomoku game in progress)

**User Interface**

**Technical Details** (some details about how you plan to go about implementing the application, and what API features you will use. This section should include your plans for how you will incorporate connecting to the internet, implicit intent(s), content providers, etc.)

For our Gomoku app we plan on using several activities and fragments to keep track of user data, such as a record of the games played, marble colors, and player names. We plan on creating a single player mode by implementing an AI that reads marble placements and places its own accordingly. The AI will scan the board both for sequences of opponent tokens as well as its own color tokens, and play either defensively or offensively as appropriate. For games with multiple human players, each will alternate turns. When placing a piece, the user will touch & drag to the desired location. Because of the imprecise nature of such a small touchscreen relative to the board & piece size, a preview of the user’s pending move will be shown and the user will click a button to confirm their piece placement.

We will create a class to represent the game board as a whole, as well as a class to represent each individual token that will store metadata such as piece color, position, and neighboring pieces of the same color in a row (progress toward win condition). We will create a main menu activity and game activity, as well as a settings fragment, help fragment, and an end game fragment. The end game fragment will contain an implicit intent to share the game results as an image with one of the user’s contacts, which will be accessed using a content provider.

The app will use web requests to download large image resources from the internet to use on the main menu, since different images will be displayed depending on whether the user is using a phone or a tablet. For the help fragment, we will include a large text box that will write out the game’s rules, how to play it, and how the score is kept, as well as developer credits at the bottom. We plan to use animations to move the tokens/marbles, animating between activities and fragments, and a special animation upon a game being won.

We will give the user the ability to save or load a game file that will be saved in local storage. Our app implements a thread through an AsyncTask that will create and operate a Media Player that is called within the end game fragment. A sound will be played depending on whether the player wins/loses against AI or in the case of multiplayer, whenever the game is over.

**Timeline** (plan to approach the implementation. In this section, you should include a series of milestones you’d like to achieve, and when you’d like to accomplish each one by. You should also include an estimate of hours of work it will take for each portion.)

Milestone 1 -- Nov 25) Graphical Elements - Board and pieces

* Approximately 16 hours between both of us to create classes, activities, and fragments as well as their UI which includes buttons, images, and the game grid itself. Also taking into account that UI will be created for phone and tablet.

Milestone 2 -- Dec 1) Game logic

* Approximately 16 hours on single player AI, Gomoku game logic, and Asynctask for sound.

Milestone 3 -- Dec 8) Full functionality - saving/loading, sharing screenshot

* Approximately 16 hours on being able to save/load game files to local storage and implicit intent for sharing game results.