

Overview:

For Project 1, I am thinking about making a game of tic-tac-toe. It will be a 3x3 grid. The game is pretty simple and for now, the game won't be storing any player scores. That can be added as an incremental feature for future versions. The game will be multi-screen, the first screen prompting the player to choose between single player or multiplayer. The second screen will be where the game can be played. The UI mockups explain more.

The intended audience for this application is anyone. If you are bored, launch the app and kill your time.

Things I plan to learn doing this project:

This application will make use of the following topics:

- Adaptive layout.
- Buttons with background images.
- Control buttons on the screen using a state machine for the buttons (this is required so that one user can select 'X' and the other user can select 'O'). Giving a button state machine will also help to remember the turns for the players and the boxes that the players have already played in.
- Perhaps use things like segmented control to choose between single or multiple players.
- Min-max algorithm.

Resources:

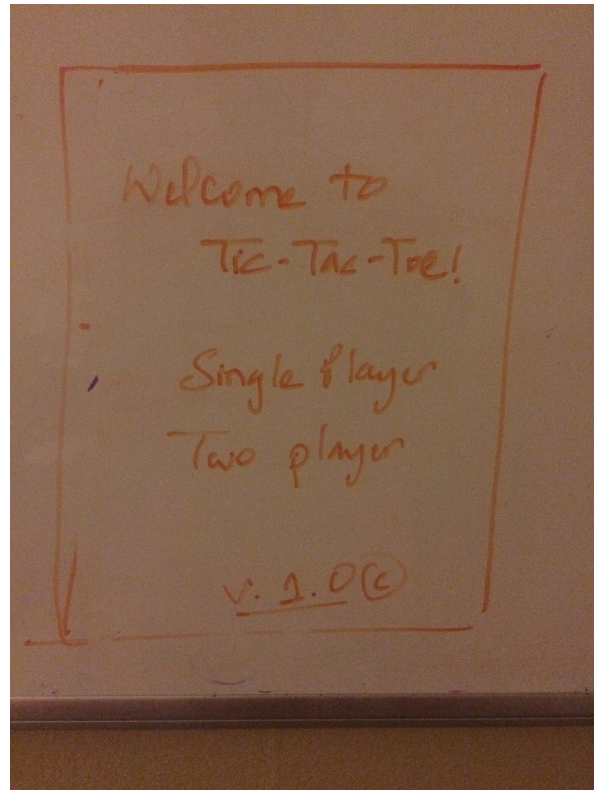
Min-max algorithm implementation in Java:

<http://www.codebytes.in/2014/08/minimax-algorithm-tic-tac-toe-ai-in.html>

Additional resource:

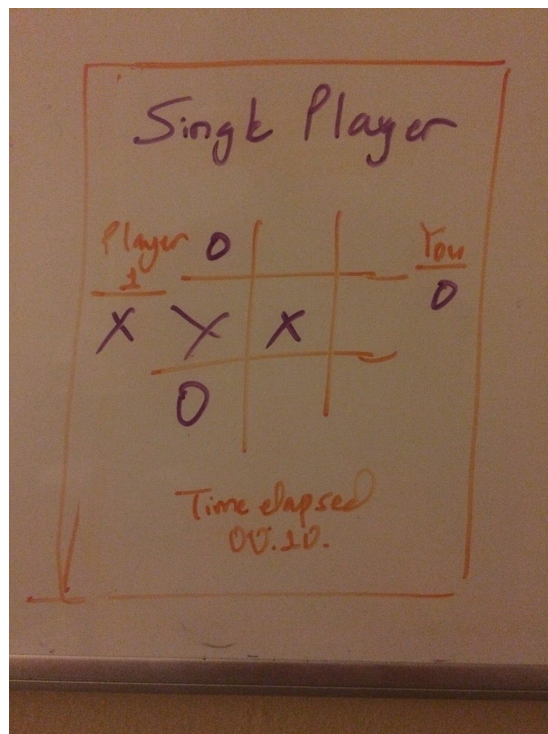
http://www.progtools.org/games/tutorials/ai_contest/minimax_contest.pdf

UI Mockup:



Screen 1 - Introduction screen:

This screen will allow the player the option to play against the computer or against a friend.



Screen 2 - Game screen:

This screen is the screen where the game will happen. The header in this screen will tell the user if they are playing single player or multiplayer.

Features of this screen:

- It will also show if 'X' is assigned to you or your opponent.
- It will also show the elapsed time.
- After the game is over, the player is diverted back to the first screen automatically.