AI For Games:
Being A Good Dad

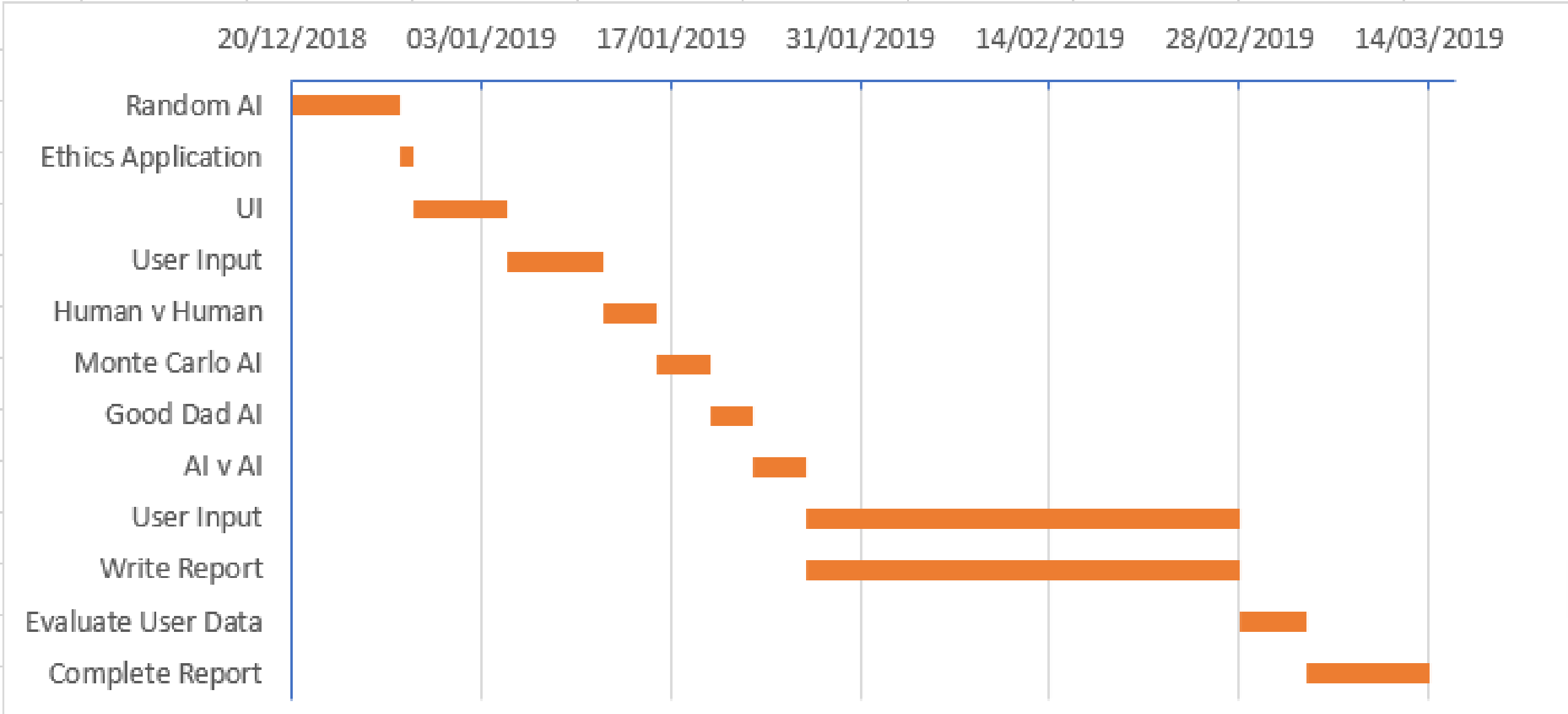
Problem Description

- AI are never challenging enough, either far too easy or depressingly strong
- Ultimate tic tac toe is a twist on the classic pen and paper game that adds an extra layer of complexity to this solved game
- Developing AI
  - Creating an evaluation function for this game is difficult so heuristic based AI (such as MiniMax) are not feasible
  - Instead monte carlo tree search will be implemented for asymmetric tree search
  - A brute force method may be possible but will not be developed

The Game: Ultimate Tic Tac Toe

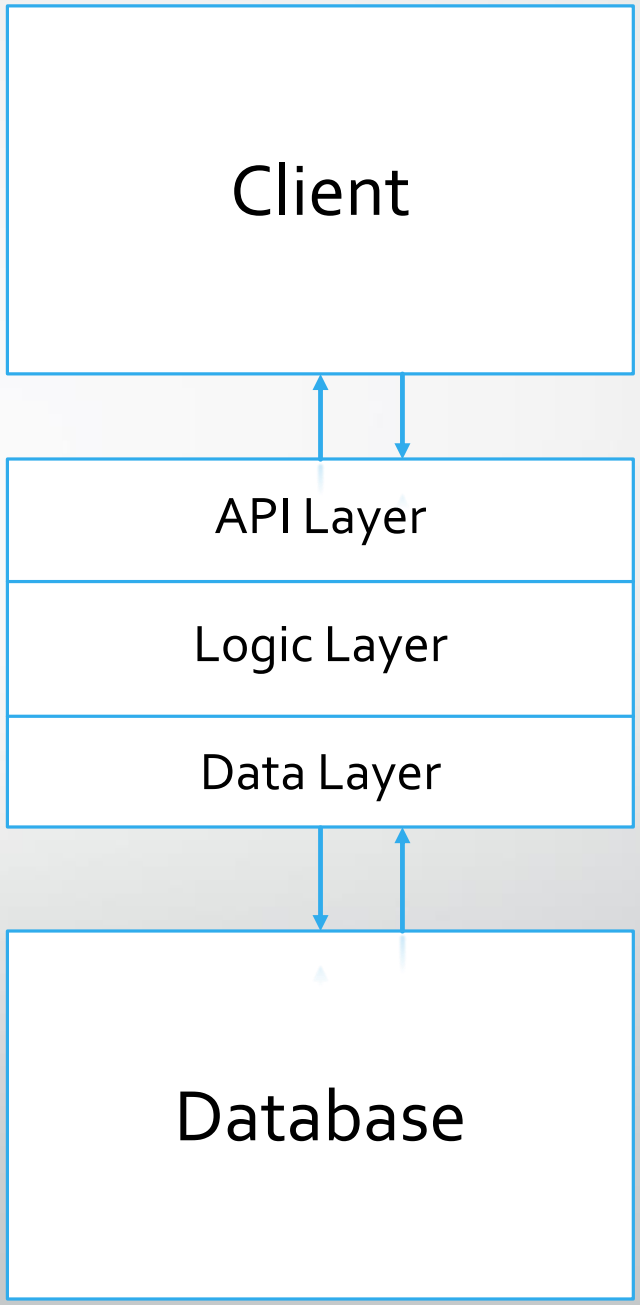
- Criteria
  - Game must be quick to play
  - Must be somewhat novel
  - Rules must be quick to learn
- Choice to implement
  - Allows complete control over code
  - Don’t have to deal with adapting API’s that are not designed for AI play

Project Plan

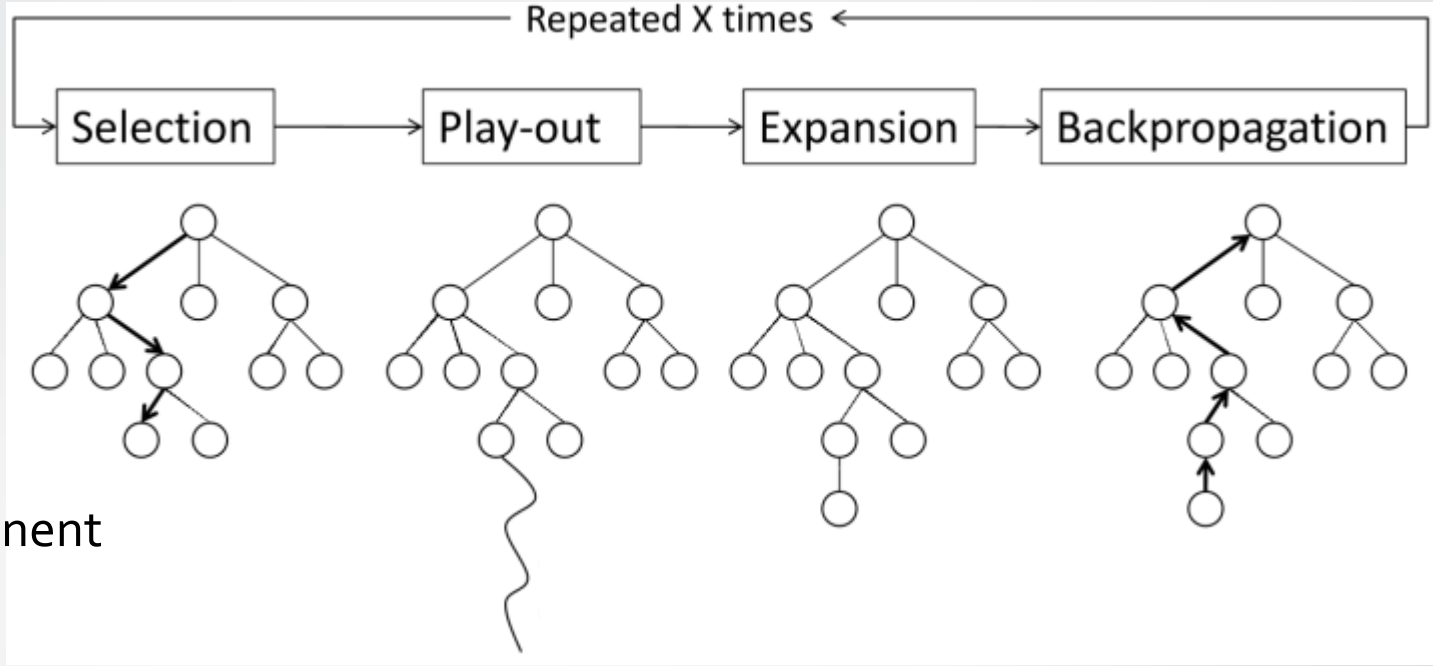


Design

- .Net Web app with angular front end
- Layered Architecture Server side
- MVVM Architecture Client Side



AI Design



- Random AI
  - Will pick moves at random
  - Intended to be easiest AI opponent
- Good Dad Ai
  - Will use modified Monte Carlo Tree Search (MCTS)
  - Will take into account the players skill level and move accordingly
- MCTS AI
  - Will use pure MCTS algorithm to attempt to find best possible move
- Rule Based AI
  - Will follow the strategy put forth in **Group Actions on Winning Games of Super Tic-Tac-Toe** (George and Janoski, 2019)
  - Will give user advice on how to best play based on it's rule set

Evaluating AI

- User Comments
  - Play through sessions
  - Or online questionnaire after games
- Experiments
  - Measure player performance (How often they pick the best move, second best etc) let them play a couple games against an AI then measure performance
  - Have Player play against an anonymous AI then a second AI and have them comment on how much they enjoyed one game compared to the other