# Tasks Completed So Far

## Game Model

I have written the game model (rules of the game, how players can move, etc) in python. The code can “play itself” making random moves.

## MCTS Code

I have written the MCTS code which takes from the AlphaGoZero approach which mainly boils down to – the NN will return (board quality , vector of move probabilities). The MCTS does *not* do a rollout when expanding a node. It just sets the reward == to the estimated board quality. Then, when calculating which child to explore, it multiplies the estimated reward of each child by *the moves’ NN probability score*. These changes made it impossible to just “use MCTS off the shelf”. I didn’t want to do that either anyway, wanted to understand what is going on.

## Training Framework

I have written about 80% of the training framework.

* An IdiotAgent, which mocks out a “real” agent, using random values
* saving / checkpointing / pickling an agent to disk once it has been improved so we can run and stop the learning at any point.
* training episodes / etc using MCTS to estimate board states, and then pick a move

# Challenges

## Model Development

Once you get into modelling the game, and do essentially “fuzz testing” on all the actions, you start to find bugs. I had to handle a few cases of stalemates (which never really happen in real life). I had to extend the model to handle some esoteric edge cases as well.

## MCTS

The tree simulation was difficult because I wanted to write it from scratch so I knew what was going on. Found a number of writeups on MCTS and some on AlphaGoZero MCTS. Spent a few evenings reading the papers, writing out the details, and then working out how to integrate the modifications that AGZ made to MCTS. I *think* I have a close replica of what they used now. It seems to be working.

# Timeline

* (1 week - Mar 29) Implement the head2head calculation at the end of an episode of training games
* (2 weeks - Apr 12) Implement the NN for an agent using Keras, training, etc.
* Rest of available time - Train - may be able to learn how to train on a GPU.
* Last week - write report / paper / etc.