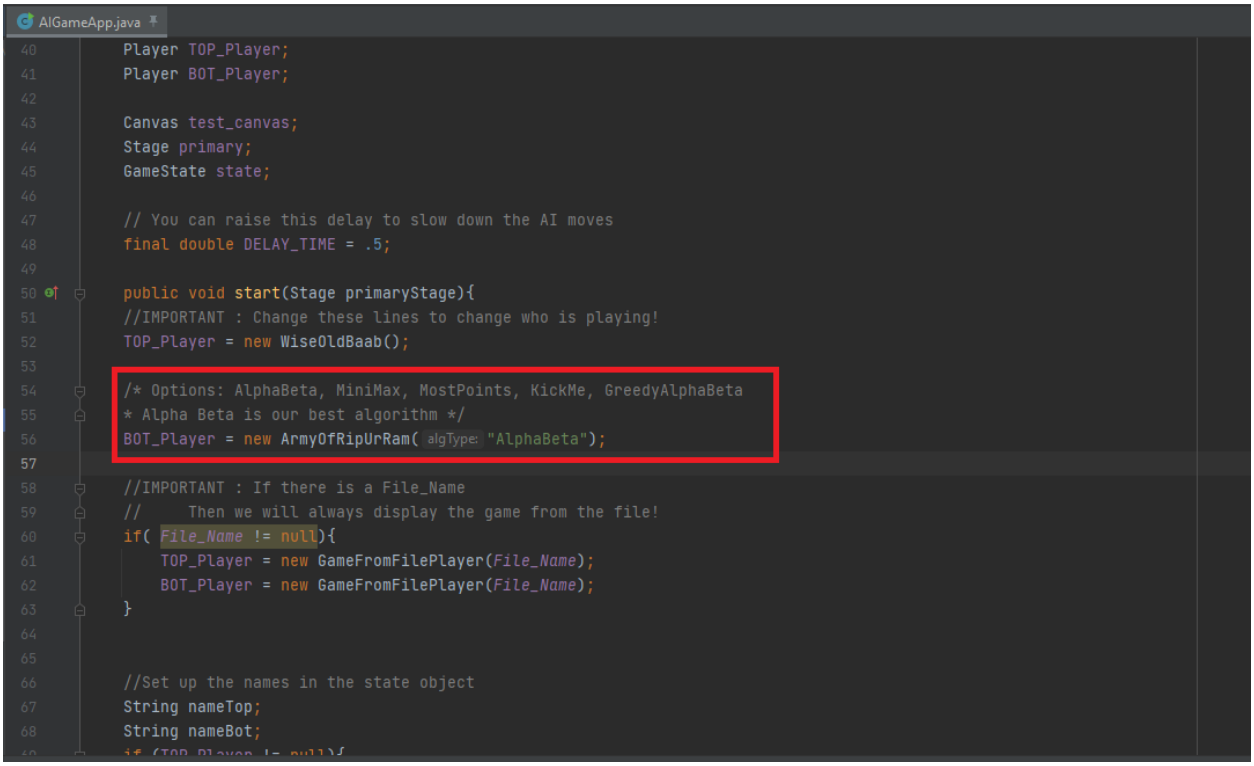

This picture just shows that to make one of our Ai's, you just have to pass in the string of whichever AI you would like to run.

All the Ai's we have

Alpha Beta

This is the best algorithm we have. It uses Alpha Beta pruning and constructs the tree as we explore the tree. That is to say that we do not construct the entire tree, only creating the branches that we explore while running the Alpha Beta pruning algorithm.

- Strengths
  - It beats the Baab army. It can search the farthest into the future compared to our other bots. Its utility function is one we have found to work the best against all the Baabs.
- Weaknesses
  - None it is godlike.
    - Only quote this if we win against all the Ai's.
- Syntax:
  - BOT_Player = new ArmyOfRipUrRam("AlphaBeta");
- Depth:
  - 12 by default

Greedy Alpha Beta

This is what we assumed would work the best. It is an exact copy of the class above, but it's utility function is different. It only evaluates the GameState based upon how many points it will have at the end of it all. Since the goal of the game is to win points, we assumed this would work fine since it can look so far into the future. We were wrong. It loses against BabyBaab.

- Strengths
  - None, it loses to everything. During the game we see it does not really do anything to keep any of it is pebbles safe from being captured. So, the enemy bubbles some stones on the left side, and then shifts one pile over and uses their next move to capture the stones they just spread across our side of the board.
- Weaknesses
  - Existing.
- Syntax:
  - BOT_Player = new ArmyOfRipUrRam("GreedyAlphaBeta");
- Depth:
  - 12 by default

Most Points

This does not use any type of tree or anything like that. It just selects whichever bin has the most stones and makes that it's move.

- Strengths
  - Very fast algorithm, and it did beat the random move and other very basic bots.
- Weaknesses
  - Moving the bin with the most stones means we let the enemy push over some stones onto our side of the board and then use their next move to capture those stones.
- Syntax:
  - BOT_Player = new ArmyOfRipUrRam("MostPoints");
- Depth:
  - 1 by definition

Kick Me

This just gets the utility function of the good AlphaBeta AI and inverts those values. So it thinks really bad moves are good moves, and vice versa.

- Strengths
  - Nothing. We thought if it made really bad moves it might mess up the enemy AI by making moves it assumed our AI would never make.
- Weaknesses
  - Any AI that has any IQ over -1 will beat this AI. Against BabyBaab we actually lost 0 – 48 most of the time.
- Syntax:
  - BOT_Player = new ArmyOfRipUrRam("KickMe");
- Depth:
  - 12 by default

MiniMax

This is the MiniMax algorithm that generates the entire tree and then explores the entire tree. Its very slow, and uses a lot of RAM during runtime. It doesn't make bad moves, and will beat the AIs most of the time.

- Strengths
    - It beats the Baab army. It wins with a fairly wide margin so its not a close battle.
- Weaknesses
    - It is slow. It takes a fair chunk of RAM and make my computer fans angry.
- Syntax:
    - BOT_Player = new ArmyOfRipUrRam("MiniMax");
- Depth:
    - 9 by default