LOG

3/12/16 (6 hr)

* Created the basic layout for the project. Since there had to be changes from last project.
* In StartActivity class, there are two options since we have two buttons. Start new game and load a game. If new game button is pressed, it takes the existing format from our RAW resource. Then it is rendered and the data is extracted from that file.
* I am guessing that there will be the initial file given to us for serialization. So, this was the purpose.
* Tried using the animation for the coin toss. Didn’t work out. Too complex.
* Tried using the dialog for the coin toss. Didn’t work out either since it is too much complex for a simple task.
* Just did the random and displayed the result.
* TO DO: currently, it does not let the user choose heads or tail. However, it does the random. So, even if the user chooses, there won’t be any problem.
* Tried the file system for saving the current games. It is kind of confusing since I don’t know where the file is saved. I searched the entire storage but didn’t find the file name. However, it runs perfectly fine.
* Changed the function readData(rawFile) to readData(rawFile, filename, indicator) where indicator indicates whether to read from raw data or the filename the user provided.
* Checked in StartPageActivity if the user entered filename already exists to open. If not, we tell the user about it and do not go through the entire trouble of reaching up to fileAccess.
* TO DO: write into the file to save the game.
* SOLUTION: While writing, open the board and go through each lines. For each column, just add a space and append. For each row, add \n so that we can distinguish later.

3/14/2016

(total – 4 hours)

* Changed the AlertDialog type for handling the actual game of heads or tails. Then, once the user chooses it, there will be another dialog box that will give the result. Then, it is directed to the main game.
* In MainActivity, things are changed now. I was under the impression that the new game had to be loaded from the RAW but it wasn’t. New game was completely new. So, I just separated the fileAccess part for the load game section.
* Made Save Game button which saves the game as savedGame.txt file.
* Previously I was using internal storage. So, changed it to external storage and formatted it with the given serialization.
* So, Lots of previous techniques were changed here.
* TO DO: Show the player’s next tile. Also option to see further in the stock.
* Showed the next tile until user satisfies. It gets next from the stock. Also, made the board clickable as I had started this project entirely new. (didn’t update from last one since I had issues with model/view).

3/16/2016

(total: 10 hours)

* Started the Min Max algorithm. The algorithm runs in Board class. So, it is the recursive function.
* In Board, made a function findNextAvailableTileNode(TileNode tileNode) which takes in the tileNode and checks for the existing location. If it doesn’t exist, that means that the tileNode is there for the first time and we start to search for the plausible location from (0,0).
* If not, then, we start from the position next to the given position.
* Made a function startAlgorithm() which is the function that is called by the other classes (especially MainActivity) to initiate the algorithm. From the MainActivity, the stock and stock index, human score, computer score and the cutoff value is taken. The stock and stock index is stored locally in the class so that we don’t change the main functions.
* Created a performMinMax() function that returns a TileNode. It takes in arguments such as TileNode as root, isMaximizer, isComputer (computer’s turn), cutoffVal, humanScore, computerScore, alphaH, betaH).
* It runs the basic algorithm.
* The cutoff value is decreased up to 0. If 0 is hit, then, we know it’s the leaf node. So, heuristic is put. I first put heuristic value generally and then put absolute sign to get the positive result. But heuristic can be negative too. So, changed it to given instruction from Prof. Kumar’s website.
* I created a GodMode which is Alpha-Beta pruning. So, inside the performMinMax() function, there is a place where it checks the condition whether to enable GodMode is true or not. If yes, then it does the alpha beta pruning.

3/19/2016

(total: 8 hours)

* Tested the algorithm

3/20/2016

* I was doing it all wrong. I am calculating the heuristic by also taking them as the minimizer or maximizer. While the minimizer is supposed to get the minimum value from its children nodes, it get the maximum from its children until now. So, I am changing it by reverting the process and actually making the root with no tile as minimizer. The whole process starts as a minimizer, this will make the tree actually senseful.
* RECORD:
  + cutoff of 5:
    - Without Alpha beta: 2.2 seconds
    - With Alpha Beta: 98 ms
  + Cutoff of 10:
    - With Alpha Beta: 15.6 sec
    - Without Alpha Beta: