CS 4701 Individual Report

Richard Greenbaum, rrg73

Pentago AI (Team Egg and Cheese, Croissants Optional)

**Section 1: Individual Contributions to the Project**

My primary individual contributions to the project were writing the functions in GamePlay.py (all except for the printBoard function) and writing the evaluate function in evaluate.py. The GamePlay.py file creates the framework for our python representation of Pentago. A game board is represented as a 6x6 2D array and an action is represented as a class with attributes for the marble x coordinate, marble y coordinate, index of tile to rotate, and direction of rotation. The file has a take\_action function which takes in a board, action, and player returns a new board that results from the given player taking the given action. It also has a rotate function that handles the array manipulation necessary for a tile rotation.

The evaluation function is the core of our system’s decision making process. It takes in a game board and returns a number that represents the strength of the given position for the bot. Having played Pentago growing up, I knew that a strong strategy was to create sequences when possible. Thus the first iteration of the evaluate function involved looping through every row, column, and game winning diagonal and counting sequences of each length for the player and the bot. Once the rest of the system was up and running, I played games against the bot and uploaded positions reached to the Perfect Pentago Interactive Explorer (<https://perfect-pentago.net/>). By seeing positions where our bot deviated from the optimal move, I was able to improve the evaluation function by filling in gaps in our bot’s logic. Improvements included adding a bonus for each marble in the center of its tile and for each marble that was not on the board’s perimeter. A more in-depth description of the evaluation function can be found in our team’s project report.

I would also like to add that everyone in the group (myself included) contributed to certain tasks including reviewing each other’s code, making the report and presentation, testing the project by playing against the bot, and getting other people to play against the bot.

**Section 2: Personal Lessons Learned**

One lesson I learned is the importance of code documentation when working on a group project. The code that I wrote for board manipulation in GamePlay.py is written using 0 as the first index, so board positions are represented using two integers ranging from 0 to 5. However, when Max wrote the minimax functions in minimax.py, he looped through all the moves to analyze by iterating from 1 to 6. This caused an error when he called my functions that took us some time to figure out. This problem would probably never have come up at all if I had written better comments for my functions, as I assume that Max looked in the GamePlay.py file before using the functions in that file.

I also learned the importance of getting outside individuals to use and review our project. One friend who played against the bot immediately told me that the color scheme used for printing the board would be clearer if the colors for the lines separating the 4 tiles were different from the colors of the rest of the lines. I had been playing against the bot so much that the color scheme hardly mattered, but it made a difference for a new user. It is important for the representation of our game board to be clear if we want to be sure that the results are due to the strength of our bot and not to player confusion. Another friend who played against our bot exposed an error by reaching a position that the bot incorrectly thought was a tie. The bug was brought about by a change that I had recently made to the evaluation function. Most of my games against the bot follow a certain path, but my friend took a new route and exposed the bug.

**Section 3: Role in other team’s projects**

I did not play a role in any other team’s project.