Jasper Shen and Ethan Cuthrell

4/4/2025

Project Intermediate Report

For our project, we are making Mancala according to the rules discussed in class. Currently, we have the game successfully implemented with all of the mentioned rules by using various helper functions that will aid us in future development as we continue to work on the project. We have created a function that will simulate a random player that chooses from a list of possible valid moves. See then Mancala.py file for the implementation of the game along with the various helper functions.

To gather data on our program so far as well as ensure functionality, we ran many games with two random players where Player 1 always moved first. We believed that running 100 games was not sufficient to get an accurate estimate for the required statistics, as the Player 1 win percentage varied between being above and below Player 2's win percentage. Therefore, we ran 1,000,000 games to allow the statistics to converge more. However, we included both results in the Appendix. According to our results, Player 1 won 49.8% of the time, lost 45.9% of the time, and tied with Player 2 4.3% of the time. We also measured the average number of turns each player took and found that Player 1 takes about 20.720339 turns per game while Player 2 takes 20.218352. The number of turns each player took per game on average is similar for both players which is expected since they are both making random choices.

After comparing the win percentages of Player 1, who moved first, and Player 2, who moved second, we realized that there is a small first-turn advantage of 4%. This makes sense because the player that moves first in the game of Mancala gets to start building their board first. However, since both players in our tests are making random choices, the first-turn advantage isn't very significant because they don't strategically take advantage of moving first. We also observed that Player 1 requires slightly less average turns per game than Player 2. This supports the idea of the first-turn advantage as Player 1 starts off the game every time, and ends off the game every time they play the winning move, while Player 2 can only match the number of turns as Player 1, even if they play the winning move.

Now that we have confirmed our current implementation of Mancala and the random player function works as intended by reviewing the data, we will move forward with implementing the player that uses minimax to make its choices. We will then have the ability to compare the minimax player to the random player by running simulated games to see how much of an improvement it makes in the simulated player's decision process. Afterward, we will continue to expand and hone our project to get all of our desired functionality resulting in a very skilled opponent for Mancala for us as well as others to play.

Appendix

1,000,000 Games Statistics:

Player 1 Games Won: 49.8%

Player 1 Games Lost: 45.9%

Player 2 Games Won: 45.9%

Player 2 Games Lost: 49.8%

Games Tied: 4.2%

Average Player 1 Turns per Game: 20.723818

Average Player 2 Turns per Game: 20.221978

Player 1 Relative Win Percentage -> 49.8% / (49.8% + 45.9%) = 52.0%

Player 2 Relative Win Percentage -> 100% - 52.0% = 48%

First Turn Advantage -> 4.0%

100 Games Statistics:

Player 1 Games Won: 52.0%

Player 1 Games Lost: 44.0%

Player 2 Games Won: 44.0%

Player 2 Games Lost: 52.0%

Games Tied: 4.0%

Average Player 1 Turns per Game: 20.59

Average Player 2 Turns per Game: 20.05

Player 1 Relative Win Percentage -> 52.0% / (52.0% + 44.0%) = ~54.2%

Player 2 Relative Win Percentage -> 100% - 54.2% = 45.8%

First Turn Advantage -> 8.4%