1 Introduction

This project looks to simulate and analyze naive artificial intelligence agents playing a game similar to *Chutes and Ladders*. An extension of the game simulated in three dimensional space with specific player conditions, this iteration introduces a *Maze* space to the board. Also, scoring and end game conditions dramatically change which may alter the effectiveness of each player personality. To ascertain the dominant strategy given these new rules, eight AI personalities were created to play the game including Random, Anti-Hold, Treasure Hunter, Sprinter, Smart Sprinter, Tortoise, Hybrid, and Maze Runner players. All personalities, save the Maze Runner, were analyzed in the second iteration of this project. The Maze Runner personality was added to this game for reasons outlined in the Approach section of this paper. We hypothesis that the Treasure Hunter personality will preform the best of any strategy as in the second project. To closer analyze the Sprinter and Tortoise players, we offer an experiment that pits the personalities against one another. As the game end conditions have changed, we hypothesis that the Tortoise player will perform better than the Sprinter personality. We also offer a complexity analysis of the game as a function of the number of tokens. As the number of tokens rises we expect the game to take a longer time to finish. We expect linear behavior of our program as this variable is increased. Finally, we test how the Maze Runner performs as the number of mazes in the board increase. We expect that the personality will win more frequently as the number of mazes increase. Analysis of these hypotheses can be found in Section 4.

2 Approach

2.1 Game Configuration Summary

From the preliminary simulations, a modified version of *Chutes and Ladders* was constructed. This project is an extension of the initial simulations. To benefit the reader, we offer a summary of the game structure from the initial projects to motivate the remainder of the analysis. The game board is represented by a three dimensional Array List[1] in which the inner most array is a collection of spaces. These spaces include aids and obstacle spaces such as treasure pots and hold queues. New to this list of spaces is a *Maze* space (see Section 2.2) which serves as a labyrinth moving players through inter-dimensional space to different spaces of the game board. A player class tree manages the players participating in the game. Each player inherits a *TokenHolder* object as well as a personality that will be used when moving tokens around the board. New to the personalities is a Maze Runner personality (see Section 2.3) Additional classes, such as a *Reader* and *Statics* class manage the necessary input files and high level management of the simulation. Classes were also constructed for exporting data and managing the game interface via the terminal. For a more in depth description of these classes and their functionality, we advise reading the reports given for the initial simulations[2].

2.2 Board Modifications

In this project, the game board now includes a *Maze* space. This space contains a weighted, directed graph[6] represented by two lists[3]. Once a player lands on this space, they must traverse the graph until they reach an exit. We consider an exit a node in the graph with no outgoing edges. If a player traverses to this space, they are moved to a corresponding exit space elsewhere on the board[6]. Upon moving to this exit space, the player is rewarded the length of their paths treasure. A player is allowed to move a token in a graph if their die roll matches an edge weight to the outgoing edges of the current node in the graph.

In addition to the *Maze* space, we add an modification to the previous *JStack* space. To ensure that no infinite *JStack* spaces arise, we ensure that each *JStack* be used only once per round. Once a player lands on *JStack* twice in a round, the token is placed on that space for the remainder of the round. In removing this problem, we ensure that our games finish in a timely manor.

2.3 Scoring Modifications

Scoring and end game modifications were made to the game. Each player is now rewarded for making it to a new level before any other token. One major modification includes the new end game condition. Instead of a single token finishing, we insist that all tokens in the game finish. Because of this new end game condition, we expect that the treasure in any game has dramatically increased value[5]. For this reason, and the maze construction, we create a new player personality, Maze Runner. The Maze Runner looks to enter every maze whenever possible. In doing so, their tokens are consistently lost in mazes, but are collecting valuable treasure. An analysis of this player personality can be found in Section 4.

3 Methods

As discussed in the introduction of this paper, this project looks to find the dominant game playing strategy for this version of *Chutes and Ladders*. To test this game, we run a wide ranging experiment that included varying dice number from 2-10, the number of players from 2-8 by two, and the number of tokens from 10-50 by ten. Each combination was simulated 40 times. From this experiment, finishing data was recorded including winner personality, treasure totals, and round totals. This experiment was completed on the *maze-25-50* graph configuration with the default configuration file. The default configuration file is summarized in **Table 1**

board	10	10	5
values	33	1.5	
HoldQ	5	2	5
Hold	10	-3	-1
treasurePotA	15	5	50
PriorityHold	5	-2	2
treasurePotB	10	60	
JStack	10		

Table 1: Default configuration file

To test the Sprinter and Tortoise personalities against one another, we complete the same experiment summarized above. This time however, we restrict the players to having only Sprinter or Tortoise personalities. This will allow for a direct comparison. Further, to complete our complexity analysis, we run the same baseline experiment as outline above but fix number of players to five and the die number to six. We then vary the number of tokens from 1-50.

To test our final hypothesis, we run the same baseline experiment as outlined above, yet add a maze for each iteration. We first run an experiment with only one maze, *maze-25-50*. We then add *maze-30-60*, *maze-45-90*, and *maze-50-100* respectively. This experiment will allow for a direct analysis of the relative progress of each player personality.

4 Data and Analysis

4.1 Dominant Strategy

To test which strategy was most dominant, a series of games were completed as outlined in Section 3. Consider **Figure 1** a summary of the number of wins by personality.

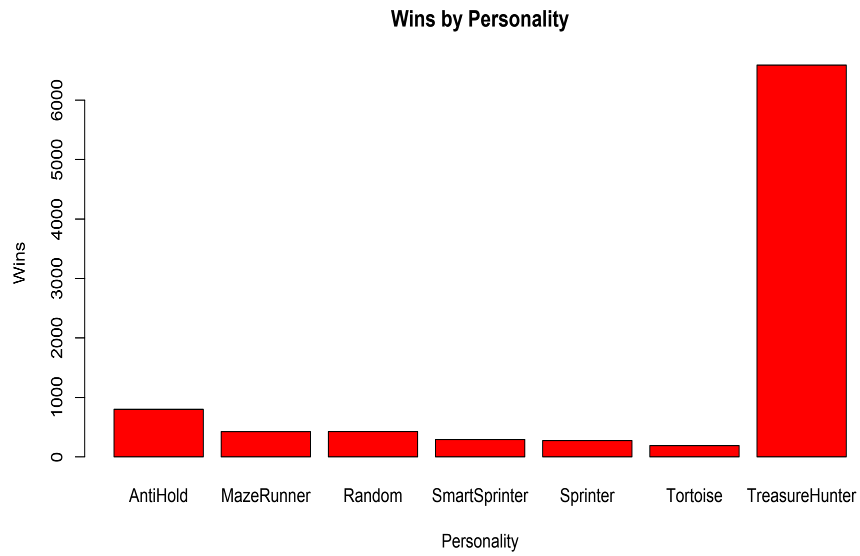

Figure 1: A summary of wins by personality

Clearly, the Treasure Hunter personality performed the best of any personality. Notice how there is not a Hybrid personality on this chart. Recall that the Hybrid player takes on the Treasure Hunter personality whenever the finishing total is under 600 points. Since our experiment was completed on the default configuration, the Hybrid experiment always took on the Treasure Hunter personality. For this reason, we combine the two personalities in the chart above. These results are consistent with the hypothesis described in Section 1 of this paper.

To better understand the ordering of the other personalities, consider **Figure 2** which offers a more granular version of the chart above.

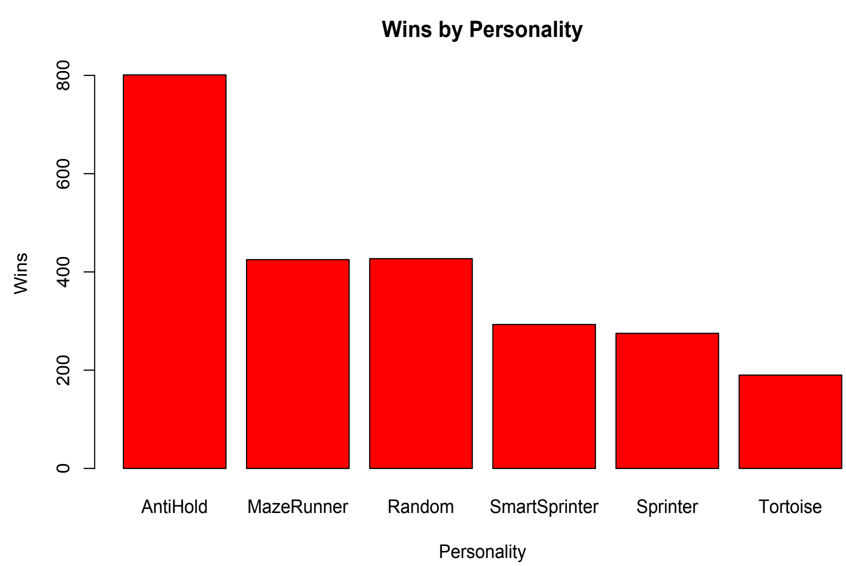

Figure 2: A summary of wins by personality without Treasure Hunter

Notice that the Anti-Hold personality performs better than any other personality. Further notice that the Sprinter, Tortoise, and Smart Sprinter all perform worse than the Random personality. This result shows that these strategies aren't only inferior, but are worse than the most naive strategy, Random. The Maze Runner's performance, while not strictly better than the Random player, suggests that this strategy performs better than the previously devised personalities. This suggests that the Maze Runner will perform better as more mazes are added to our analysis. This question will be explored in Section 4.4.

4.2 Sprinter vs Tortoise

As discussed in the Approach section, players are now rewarded for reaching new levels of the board before any other token. Therefore, we expect that Sprinter personalities will achieve more points than the Tortoise personality by traversing these levels faster. The Tortoise player, however, will attempt to move its entire team together across the game board. This suggests that the Tortoise will be able to attain more treasure from the Treasure Pots on its initial board traversal. As we know treasure in this game is more important than finishing points, we expect the Tortoise player to out perform the Sprinter personality. As discussed in Section 3, we devise an experiment to test this relationship. Consider **Figure 3**, a summary of number of wins by personality in this experiment.

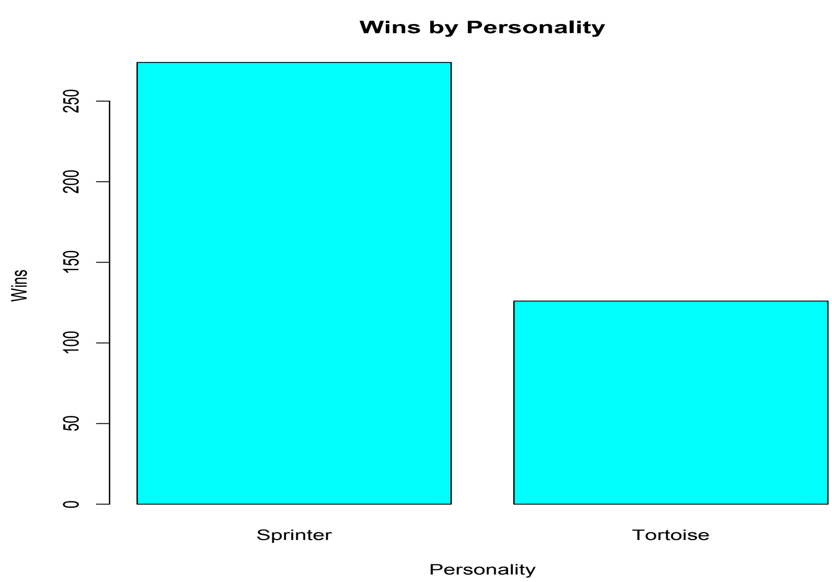

Figure 3: A summary of wins by personality

Clearly, the Sprinter personality won the majority of the games in this experiment. This disproves the hypothesis discussed in the introduction of this paper. This could be due to the fact that Tortoise players never attain the finishing and new level bonuses in addition to never attaining additional treasures because it does not actively seek treasure.

4.3 Token Complexity

To test the complexity of the number of tokens, an experiment was devised to vary the number of tokens while keeping all other variables constant. The details of this experiment can be found in Section 3. Consider **Figure 4** a basic plot of the complexity analysis.

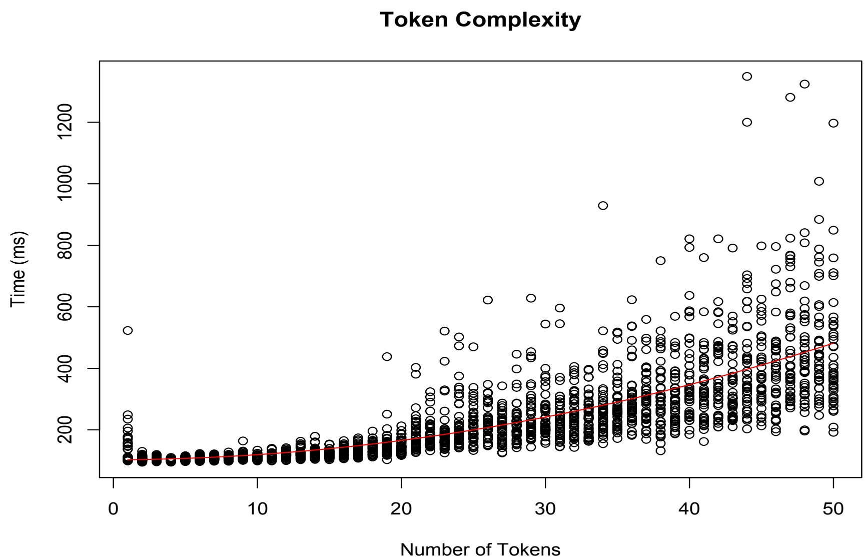

Figure 4: Number of Tokens by Time of Completion

Clearly, as the number of tokens increases so does the time to completion. It was hypothesized that this complexity was $O(n)$. The model fit here is of order 2 suggesting complexity $O(n^2)$. With such strong heteroskedasticity, it is hard to make a compelling argument for either complexity of $O(n)$ or $O(n^2)$. Through a general transform and a more robust experiment, the true experimental complexity may be uncovered. For this reason, we have no evidence to either support or reject our hypothesis[2].

4.4 Maze Runner

The Maze Runner was the only personality added to this new version of the game. A natural question that arises is if this player performs better or worse than the original personalities. We expect that the Maze Runner performs superior to comparable strategies as the number of mazes increases. To test this claim, we devise an experiment as discussed in Section 3. Consider **Figure 5**, a summary of number of wins by personalities with variable number of mazes.

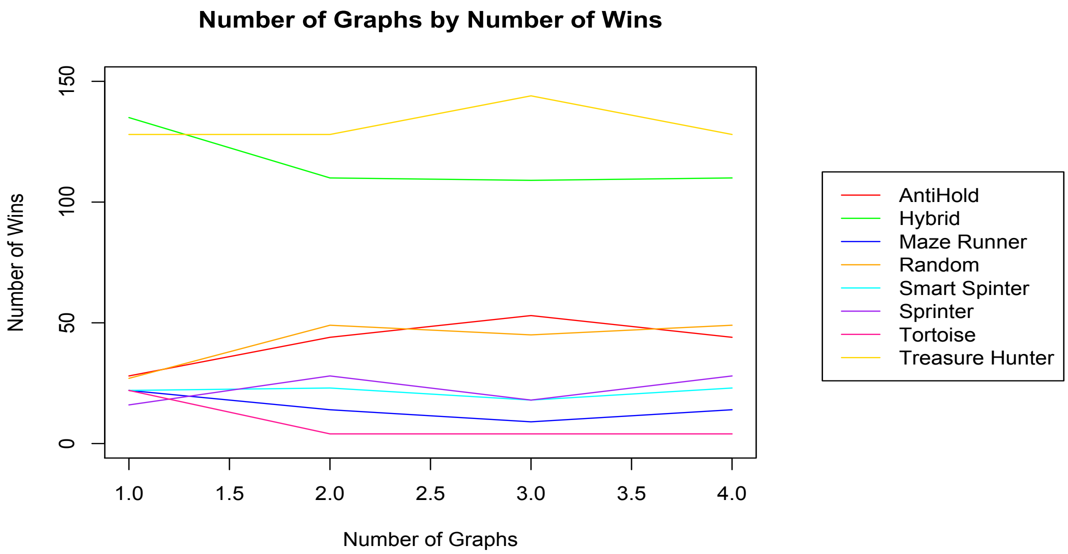

Figure 5: Number of wins by number of graphs

Notice that the Hybrid and Treasure Hunter strategies perform strictly superior to all other strategies. Furthermore, notice that the Anti-Hold and Random players perform better than the remaining strategies regardless of number of mazes in the analysis. This suggests that the Tortoise, Sprinter, Smart Sprinter, and Maze Runner, are all inferior to even the most naive strategy. Furthermore, we see that all lines have relatively low slopes. This suggests that as the number of mazes have little affect on game play. This directly contradicts our hypothesis. These results could be due to the fact that the game was forced to terminate after 100,000 rounds. Therefore any tokens caught in mazes were not utilized throughout the game. For a more robust analysis, removing this constraint might offer radically different results.

5 Conclusion

This project extended the initial simulations of *Chutes and Ladders* to included a Maze Space. This paper completed analysis on dominant game playing strategies. We found that the Treasure Hunter was again the dominant strategy. In addition, we found that several of the previously designed personalities performed worse than even the Random player. In addition to this AI analysis, this paper finds the complexity of the number of tokens to be of order $O(n)$ or $O(n^2)$. The analysis of this final iteration of *Chutes and Ladders* finds that the treasure spaces are more valuable than any other in this game design.

6 References

1. "ArrayList (Java Platform SE 8)." *ArrayList (Java Platform SE 8).* `https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html` N.p., n.d. Web. 19 Feb. 2016.

2. Degroot, Morris and Mark Schervish. *Probability and Statistics.* Boston, MA: Addison Wesley, 2002. Print.

3. "Project Description 2." `http://cs.lafayette.edu/~liew/courses/cs150/` N.p., n.d. Web. 17 Feb. 2016.

4. "Project Description 3." `http://cs.lafayette.edu/~liew/courses/cs150/` N.p., n.d. Web. 17 Feb. 2016.

5. "LinkedList (Java Platform SE 8)." *LinkedList (Java Platform SE 8).* `https://docs.oracle.com/javase/8/docs/api/java/util/LinkedList.html` N.p., n.d. Web. 19 Feb 201.6.

6. Weiss, Mark Allen. *Data Structures and Problem Solving Using Java.* ed 4.Webing, MA: Addison Wesley, 2010. Print.