Team Number: JMS32
School Name: Jackson Middle school
Area of Science: Analytical
Project Title: A-Maze-ing Efficiency

The Problem

Mazes are very interesting, they are problems waiting to be solved, but what is the best way to solve them. First we have to think about what kinds of mazes we will test on, right now our project focuses on two types of mazes, one being the standard start at the edge and end on the edge, the second being labyrinths which are just mazes that have a start on the edge and end in the center. Then there is the question of how to solve the maze, the first and easiest was the wall follower which was referenced in Pullen's *Maze Algorithm* and it's pretty obvious what it does, it follows the wall , and this works every time in the standard maze, but that's not true for labyrinths or mazes with the exit in the center. So, What is the best algorithm to solve any maze every time?

How are we solving it

We have two different types of mazes to test alongside. A maze that has standard 90 degree turns, and a maze with random angles for it's turn. Several types of agents would be "taught" different types of maze algorithms. These would range from always taking a right hand turn to a group of algorithms. This would allow a way to view the effect of different techniques from inside of the maze. We would then use a an omniscient view to solve the maze with the shortest path. This would help us to compare the techniques to how to solve a maze perfectly.

Progress that we've made

We have had some trouble with the focus of our project and it has changed quite a bit, our interest just became less and less as we continued the project. The First Idea we had for a project was looking at how entropy increases or decreases within a file before and after data compression. We did quite a bit of research about the compression of files and what ways files were compressed. The research for that idea was going really well, and we found a lot of useful information, but we really did not see a clear path to modeling this problem. So we looked at ways we could quantify the information and we could really only look at the size of the file, and that just wasn't enough for an entire project as we could quickly import a file into a python program and get the file size, this program would only be a couple of lines. After realizing we had to do something more, we looked at ways we could modify the compressor, so that it takes away or approximates specific parts of a file, so that we can optimize data compression, but that

would require that we rip a lot of the program apart and we found that we have no interest in that. So After we lost interest in the project, We decided we would look at maze algorithms and what algorithm was the best at solving mazes. So we would create different types of Mazes and have turtles that have their own ways of solving them, such as the right hand wall method, where they follow the right wall. This is where we got a lot of a research done, we looked at the different ways that we can solve mazes and different mazes that we can make. We plan to start the program a soon as possible, but we have setup the initial program like the first maze and the turtles that will move randomly.

Results we expect

We expect to find a method that is close to the perfect path. The different uses of techniques would demonstrate the different types of ways a person could use to solve the maze. This would help people who work at or attend mazes and those types of attractions to have the quickest method of finding their way.

References:

Pullen, W. D. (n.d.). Maze Algorithms. Retrieved November 27, 2017, from http://www.astrolog.org/labyrnth/algrithm.htm

"Cris' Image Analysis Blog." *Cris Image Analysis Blog RSS*, www.crisluengo.net/index.php/archives/277.

"Maze Generation." *Maze Generation - Rosetta Code*, www.rosettacode.org/wiki/Maze_generation

"Programming Theory: Solve a Maze." *Algorithm - Programming Theory: Solve a Maze - Stack Overflow*, www.stackoverflow.com/questions/3097556/programming-theory-solve-a-maze

Rathor, Shantur, et al. "Maze Solving Algorithms." pp. 1–15.