The model-based learning works by making a class for each location provided. The class stores internal information such as value, name, and which actions have happened and how often. In addition, the class stores multiple dictionaries which hold the possible actions at the location, the possible results of each reaction, and the probabilities of each of the action/destination pairs. The class also generates which outcome occurs and which action occurs when a random one is needed. More exploration in the beginning leads to a wider, more even spread of probability, as the model tries many options before settling on one, and therefore is more likely to the most optimal path. Focusing on exploitation causes the model to find a decent path quickly, but may not be the optimal one. When finding paths to the goal and filling out the model, the epsilon value determines how likely it is for the model to choose a random action versus the one the model finds the best. A higher epsilon leads to a more even distribution of the probability, as the A.I. is more likely to make non ideal moves that it would not make following the model. The program stops learning once 1000 paths to the goal were reached. That number is significant enough that all combination of action and destination occur, even with a low epsilon. The large data set also provides a decent average, meaning it is closer to the true values. To determine what the value of the location, the values of the adjacent locations are collected. The adjacent values are multiplied by the probability and the gamma value, with a constant 0.04 subtracted to account for the downside of moving. The gamma value going from high to low did not drastically change the probabilities. The total value of the location is the highest of the possible actions. Nicholas Gerold coded the model based learning portion