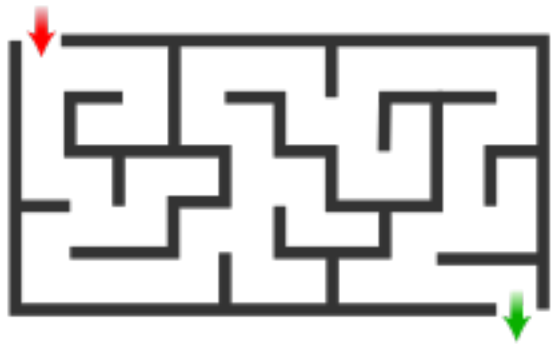
The Mouse and The Maze

Imagine you're a mouse and you need to get from the red arrow to the green arrow in the maze below.

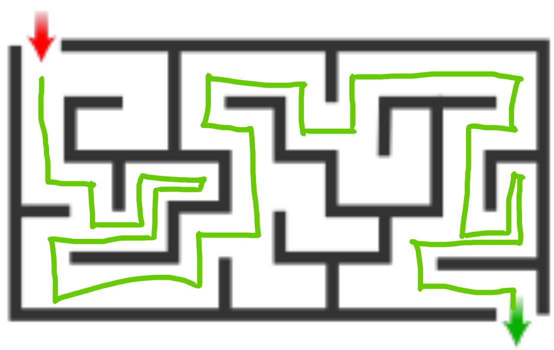


Source: Wikipedia

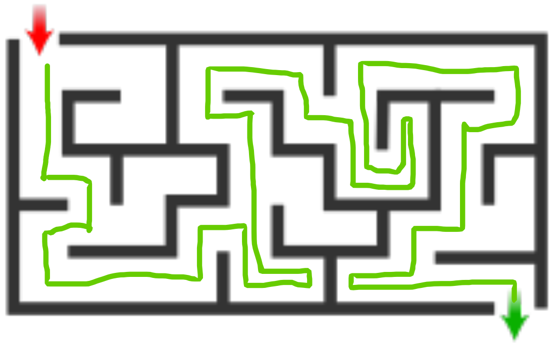
Now, being a smart mouse (smart for a mouse at least), you know that to find the end of the maze you only need follow a simple set of instructions.

1. Keep moving forward until you hit a wall
2. When you hit a wall rotate left 90 degrees (turn left) and try to move forward.
3. If you are still facing a wall, turn around completely (now you are turning right).
4. If you still can't move forward, face the way you came (the opposite direction you were moving in step 1) and go back to step 1.

If you followed that strategy your path would look something like this:

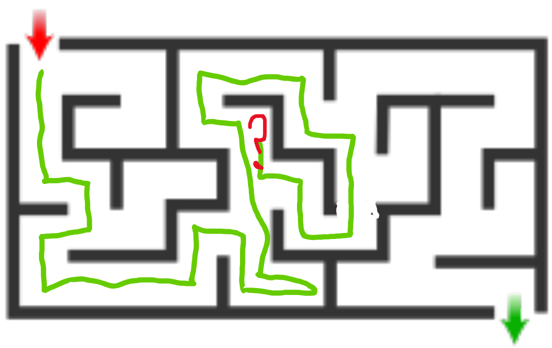


If you instead decided to turn right first would look like this:

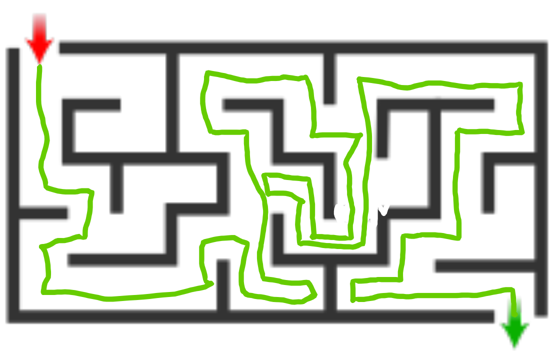


This approach, always turning left or right, is called the algorithm and if you, as the mouse, blindly follow this algorithm you will always make it out of the maze. Another, very important observation is that it doesn't matter where in the maze the mouse is - the algorithm doesn't change. We could drop the mouse anywhere in the maze and as long as the steps are followed, the mouse will make it out. The mouse doesn't have to have a memory of where it's been or even a notion of where it's going.

If we change the maze just a little bit, say by removing a wall we can "break" the algorithm



Once the mouse hits the loop it will continue in that loop forever. This can be fixed by changing our algorithm so that when we arrive at somewhere we've been before we can make use of that, say by turning around and retracing our steps.



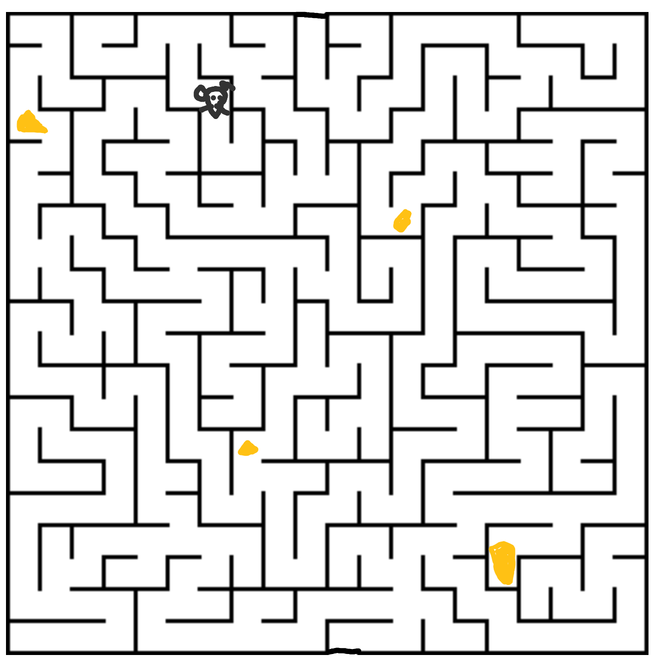
That's all that we'll talk about loops for now. Knowing that loops exist is important and any mouse that finds itself in a maze with a loop needs to know how to navigate the maze and not get stuck.

Let's Play a Game

Let's reframe our discussion just a bit. Instead of the imaginary mouse going through some maze, we'll invent a game; The Mouse and The Maze. The game consists of a maze, a mouse, and one or more pieces of cheese. Each piece of cheese is worth different amounts ranging from 1 - 100. The game begins placing the Mouse and the pieces of cheese randomly in the maze. How many pieces of cheese to use and the values of each piece of cheese are up to the players. The rules of the game are:

1. The Mouse moves one step at a time through the maze. Turning does not take a step.
2. The Mouse cannot change the maze.
3. The Mouse may not move through a wall.
4. The game ends when the Mouse arrives at a piece of cheese.
5. The score is the value of the cheese minus the number of steps taken. For example, if the cheese is worth 100 and it takes 10 steps to find it the score is 90.
6. When the game starts the Mouse knows nothing about the maze including how big the maze is. The only information that the Mouse is given is what direction (forward, backward, right, left) they are allowed to move at any step.
7. The maze will not have any loops.

The goal of the game is to score as high as possible.



We can see an example game of The Mouse and The Maze in the above image. If you recall the previous discussion about how to move through a maze from start to finish you will also see that this technique can be used to play this game. The start of the maze is the starting point of the Mouse and the exit is a piece of cheese. So one way to play the game would be to brute force through until a piece of cheese was found. This also means that we can define an algorithm to play the game.

It turns out that if those are the only rules of the game, the brute force method is the only method that makes sense. The game at this point is just a mouse, blindly wandering through the maze hoping to find the cheese and the only thing that the score depends on is where the Mouse and cheeses randomly start, e.g. the Mouse starts on top of the largest piece of cheese, zero steps are needed so the score is the value.

Skill is not required to play this game.

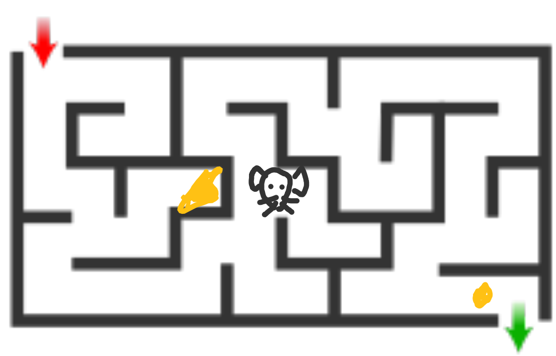
While this sort of game might be ok as the basis of a gambling game the point of this article is really about AI and while AI does have elements of randomness in it, it's not completely random.

There is actually a very easy change we can make to add some skill. We allow the Mouse to know three things about each piece of cheese.

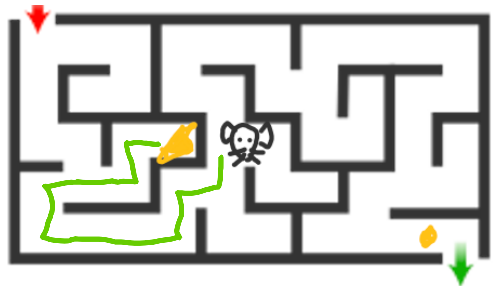
1. The value of the piece.
2. How far away the piece is.
3. What direction the piece is.

If we allow the Mouse to know this information the Mouse can know make informed decisions about where to move next. We use the word informed here because the Mouse now knows more about the maze than just then what's immediately around it.

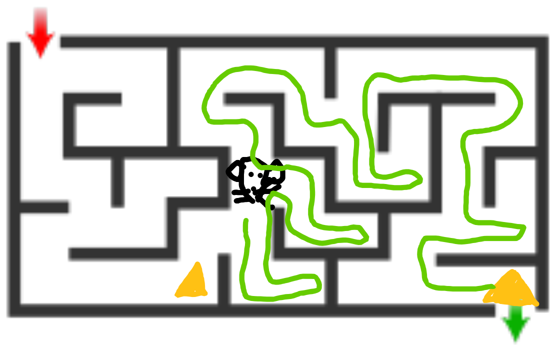
Let's use our simple maze for the game to show how this might work.



In the maze above the big piece of cheese 1 away to the left and the small piece of cheese is 4 away and down to the right. Since the Mouse knows this information the Mouse can start making decisions on which way to go based on that; in this case it's obvious that the Mouse should move toward the biggest piece of cheese.



This is a much more direct path and the Mouse will get a higher score. Here is another maze but with the cheese in different locations. Remember, the Mouse always moves in the direction of the largest cheese.



This time the path is a little less direct but eventually we get to the biggest piece of cheese. This may not always return the highest score though. Remember that the score is the value of the cheese minus the number of steps taken. So in the map above, if the values of the cheeses are 100 and 95 and it takes 20 steps to get to the larger cheese but only 3 steps to get to the smaller cheese then the Mouse would score 80 for getting to the larger cheese but would score 93 if instead they get the smaller cheese.

We can make the Mouse smarter if we want. One simple IQ boost is to make the Mouse smart enough to calculate what the score might be at each step. Let the distance to the smaller cheese be 2 and the distance to the larger cheese be 10 (note, distance, not steps since we don't know what walls are between the Mouse and the cheese). The Mouse then knows that if they head to the smaller cheese the score could be 93 and if they move to the larger cheese the score could be 90 so the smart thing to do (smart == higher score in this game) is to move toward the smaller cheese.

We're going to make one last change to how we score the game. Now the score is:

score = value of cheese – steps taken – time taken – food eaten

The Mouse needs time to think about what decision to make. The amount of time depends on how complex and how many calculations the Mouse must do to decide which way to go. For example, the Brute Force Mouse need doesn't need time to decide, just like a Roomba, it moves forward until it bounces off a wall then changes direction.

Our distance calculating, score guessing, Mouse needs time to do the math at each step. The more time it takes for a Mouse to make a decision the lower the score (let's say the cheese rots over time).

Let's also give our Mouse a super-ability. That super ability is to be able to teleport back to any point in the maze that it has already been. Teleporting still counts as a step but only one step. Let's say our Mouse has taken 5 steps down a corridor in the maze and hit a dead end. Since we're counting steps, the steps to move backward also count. The dead-end would count for 10 steps (5 down, 5 back). Adding teleport allows the Mouse to now only need 6 steps (5 down, 1 back) which means the Mouse will get a higher score.

For our Mouse to teleport though, it must memorize where it has been. The Mouse can only teleport to somewhere it remembers, and memorizing something requires food. For our simple version of The Mouse and The Maze the rule is that memorizing one location requires one piece of food.

There are other super powers we can add, for example, we might give our Mouse two brains, or maybe an extra fast brain. Regardless of the additional powers, they all will require food.

Now we have a game of skill. The Mouse can be smart and be made smarter. Also at this point, hopefully you've played along, maybe even played a game in your mind, you have a good mental model to understand some very complex computer science topics. For example, take a look at the [A* Algorithm] (Wikipedia link). If you don't have a computer science background (and that's who I hope I'm writing for right now) at first it may seem like techno-babble. And for the most part, it really is. Now, there are valid reasons for that techno-babble within the field, but I don't think it's always necessary to get at the heart of a thing. And at the heart of the thing, A* is just a smart Mouse. The way the Mouse decides on which way to go is based on the following:

Using a game of The Mouse and The Maze with one cheese in it.

Start the game by moving a step in whatever direction moves the Mouse closer to the cheese. Memorize that step 0 is the distance to the cheese. From our first step, find the distance to the cheese. Take that value and add 1 (for step 1). Next, for each direction that the Mouse can take, either forward, left, or right (we don't need to calculate backwards) take the distance from that step to the cheese and add 2 (for step 2).

The Mouse then moves to the step that has the lowest value. So if forward was 5 from the cheese, left was 4, and right was 6, assign values of 7, 6, and 8 to forward, left, and right. However, we might not take those steps, because the Mouse always moves to the lowest value step that it has memorized, including step 0. This is where teleporting comes in handy.

The Mouse can now explore down a path, but also recognize that the path it taking it too far away from the goal. Remember the dead-end corridor from earlier? What if the Mouse realized after taking two steps down the corridor that it would be better to explore somewhere else that might be a better option, meaning takes less steps, meaning a higher score.

If you were to draw a maze and play a game of The Mouse and The Maze using A* as the brains of the Mouse chances are very good that that Mouse would outperform other types of Mice, for example a Brute Force Mouse. You'd have a smarter Mouse.

The Right Tool

Up until this point The Mouse and The Maze is actually playable. Let me rephrase that, up until this point we haven't talked about versions of The Mouse and The Maze that are not practical to try to play. From here are things get much more abstract and may seem complex at first but just remember that at the end of it all, it's just a smart Mouse running around a maze trying to find the best cheese in the fewest steps, the quickest time, and eating the least amount of food.

Now image a maze that's a large as the United States. Each step, though, is only a foot. Maze as far as the eye can see. If there's only one cheese, there's only one goal and regardless of our efforts the only way to get to the cheese is to take those steps. The Mouse must take the steps, there are no alternatives. So, for these kinds of problems our Mouse has to be efficient. Remember our equation,

score = cheese value – steps – time taken – food eaten.

The value of the cheese never changes, there is only one cheese in this game. The only way to raise the score of the Mouse is to reduce the steps; reduce the amount of time it takes to make a decision; reduce the amount of food required for the Mouse (remember, smarter mice eat more food). It's a balance though, because even though smarter mice eat more food, smarter mice may also be able to make decisions faster. Note that I said may also and not always. Being smart in the right way to get the best score is what matters. In gaming there are whole sites dedicated to this sort of optimization problem, how to I max my Barb, what are the best Sorc builds? I want an Ice Sorc, how should I build that? Those trade-offs and calculations are what the study of Algorithms is about for a computer scientist. It's just knowing the right tool.

Multiple goals add complexity

Now let's make things more difficult for our mice. Instead of just one cheese, now we have multiple cheeses spread through the maze. Our A* Mouse is in trouble now because that Algorithm doesn't take the value of the cheese into account. Instead, the A* would just blindly (remember that word) head towards the nearest cheese as efficiently as possible. We could modify A* to take value into account, maybe subtract the value of the cheese from the distance and steps? Regardless, the main point is that just by adding a second cheese we've made the problem way more complex. Now the Mouse has to consider value when making a decision. Up until this point (where we add more cheese!) only cost was a factor. I think the best real-world analogy would be choosing from 2 different jobs offers in different cities. Sure, you might get paid 15% to work in New York City but are your living expenses in New York only going to be 15% more?

There are algorithms that handle these sorts of problems. You might run across names like constraint satisfaction, multi-variate optimization, again for purposes of The Mouse and The Maze, all techo-babble. By recognizing the names though it might help you to understand the problem being solved. How many cheeses are in the maze?

You don't need a lot of examples, just the right ones

You have probably used Google Lens or something similar or taken a picture of a leaf and used an app to identify the tree. All those things fall under something called computer vision. It's literally just taking the output of a camera and doing useful things with that information, like "self" driving. Powering many of those applications are something called convolutional neural networks or CNNs. I'm going to skip the convolutional part of it and just focus on the neural network aspect. To do this we need to talk about a different way to train a Mouse.

Imagine that you're at a The Mouse and The Maze Association of America 4th Annual Competition (look up puzzle competitions if you're skeptical this could be a thing). You want your Mouse to get the highest score obviously and so you do some research on past competitions. You notice that the competition mazes are similar. They are not the same, but maybe you see that certain patterns within the maze are repeated and when the cheeses are within those patterns there is an obvious best path to get to the cheese. What you could do is you could train your mouse to remember those patterns and also remember the best path through that pattern as well as how to best avoid dead-ends.

The week before the competition you start training your Mouse. Now, being a good trainer, you keep meticulous records of your Mouse's performance. At the end of a practice run you adjust your Mouse, make small corrections, then do another practice run. You do this several times until you thing the Mouse remembering the patterns.

At the end of each day you test your Mouse using some mazes you didn't use for training. You test each maze with a variety of cheese locations and average the scores. At the end of each day, if your Mouse is being trained correctly, the scores should get higher.

When the competition finally arrives what you hope, and what you researched for, is that the maze presented at competition follows similar patterns. The Mouse must make a decision to move a step in one of the directions. If your Mouse has been trained on mazes that are similar to the one being used in competition the Mouse will make better decisions, the Mouse “has seen this before”. If instead, the Mouse is put into a maze where the pattern has never been seen before, the Mouse will probably be confused and head off into strange directions. In neural network terms, if you train a neural network using only images of apples and then show it a picture of an orange, you’ll probably get a strange answer. You have to educate your Mouse using the right information.

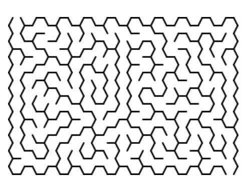
That’s all neural networks are, showing a Mouse lots of patterns. The key to training a Mouse to win a competition, or a neural network, to identify an orange, is to use the right mazes. One other thing, neural networks answer fast, just like we would expect our Mouse to do. The Mouse should make decisions quickly because it’s memorized the patterns. The other side of that is that they take a large amount of resources to train. You had to spend a week, endlessly running the Mouse through different training mazes. That takes time and resources (money, space, food, etc.). Training a mouse this way can be expensive but if there are patterns and you use the right ones, it can pay off.

You Can’t Solve a Problem You Don’t Understand

The simplest maze there is has just one spot. When the game starts, the game immediately ends because the Mouse and all the cheese are in the same place. It’s not a very good maze, but it is a maze. A maze could also be an empty room. Mice put into empty rooms don’t have to be smart at all, they just follow their noses to the biggest pile of cheese (btw, what I just described is algebra). Mazes don’t have to be like any of the ones I’ve shown so far.

Mazes might be so large as to practically be infinite or may actually be infinite. Think back to the maze the size of the United States. If you were navigating that maze would it matter if it were country sized or world sized? It only matters if it affects how close the cheese is or, as we’ll talk about later, how much food we need to feed our mice.

We could create mazes that have loops and any Mouse that gets caught in one is disqualified. The only key requirements are that the maze is made up of steps (you’ll hear the word discrete used) and may have walls that block the movement from one step to another. We can create mazes that are not made up of squares.



Hexagonal maze

Think about where you live? Is that a maze? If you started your front door and had to get to your bedroom to take a nap (the cheese), is there one path or multiple? Is one of those paths better than the other? What if you were also hungry and stopped by the kitchen to grab a small snack on your way? It's very likely that even before you walked in the front door you had already formulated a plan: walk into the house, walk to the kitchen, grab an apple, eat the apple, then head to your bedroom to take a nap. Those sets of steps are you navigating through a maze.

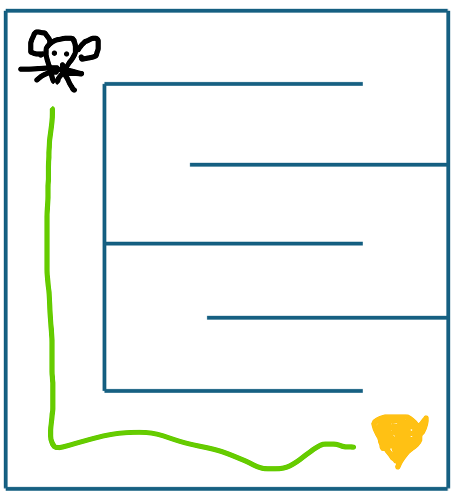
Now what happens if you get to the kitchen and there are no apples left? If you happened to know about it ahead of time, for example, you knew what the maze was, you might have stopped to buy some at the store on the way home. If, on the other hand, no apples was a surprise, you've come across a part of the maze that you didn't expect. Now you have to make a decision (forward, left, right). You could skip the apple, go buy some more at the store, or eat an orange instead. Your ability to both quickly make that decision (remember, you want to take a nap) along with making the correct decision (maybe eat an orange to satisfy that hunger) helps you then find the right cheese and most important of all, get a high score.

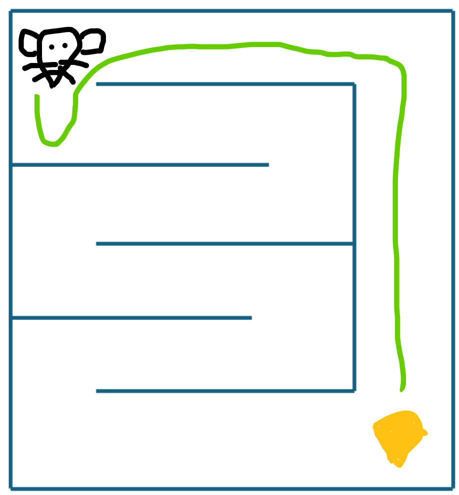
Every one of those steps, each decision point, is simply a step in the maze. The actions you must take to get from your starting point to the cheese (taking a nap) is the maze you have to navigate through to get there. If you, The Mouse, have information about the maze ahead, for example, by knowing there are no apples, you can navigate the maze more efficiently (stopping at the store on the way home). You have used your memory (remembering there are no apples) to take a path through the maze that should require fewer steps than if you discovered there were no apples once you reached the kitchen, let's say you absolutely had to have an apple so the only solution is to get back into your car and go buy apples (backtracking). Your score for your nap would be lower than if you had remembered to get the apples on the way home because remembering allowed you to take fewer steps; no backtracking (we can't teleport in real-life).

The real-life maze we all navigate is too complex to visualize. The small things in life, the direct tasks that we all have to do (eating an apple) are simple, but even within those seemingly simple tasks there are an impossible to count set of smaller mazes. Taking a bite out of that apple is a maze and human brains and nervous system have developed in a way that we are able to carry out those things efficiently. If you don't think that taking a bite of an apple is "hard", try writing down the instructions to do so for someone who has never taken a bite of an apple. Did you include something about not taking a bite of the stem? What about not putting the entire apple in your mouth? Those are both decisions that a person (or machine) following your instructions would want to avoid to maximize the score? Eating the stem is a very low value cheese (it could even be negative, think putting your hand on a hot stove). Getting the right, tastiest, easiest to chew bite is the highest value cheese and finding the best cheese is the whole point of the game.

I’ll finish this discussion about mazes with two real-world AI examples where what the maze the Mouse thought it was navigating and the maze that the Mouse was actually in were two different mazes.

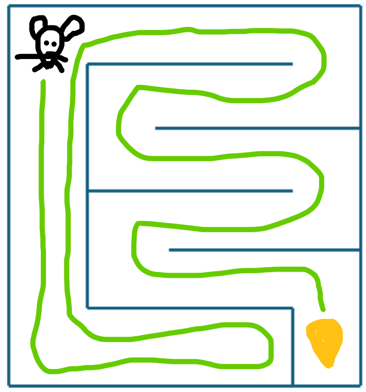
The first is an urban legend within the AI community about finding tanks. You can easily find more about this by a web search for “computer vision tank urban legend”. The gist of the story is that researching training a neural network to find tanks within pictures discovered that instead of finding tanks, the neural network was instead detecting the different between morning and afternoon (all the pictures with tanks were taken in the morning and those without in the afternoon). To put this in terms of our game, let’s create a couple mazes that might help explain why this would occur.





These mazes are very simple but there is a strategy that will help the Mouse determine what the best path is. The Mouse starts by moving down one step. If the Mouse now faces a wall it should turn around and go to the left (not right, remember the Mouse turned around). This sort of rule is called a heuristic. Heuristics are shortcuts to make decisions without having to think about the decision, for example, always turn left.

What happens if the Mouse is put in this maze?



The Mouse, following the strategy it was trained on, now takes the worst path possible through the maze. Even if our Mouse can teleport (it also needs more food and so gets a lower score) the path still takes it down to the dead-end and this means the mouse gets a lower score. Remember, above all things, the point of the game isn't to find the cheese, the point of the game is to get the highest score.

The mazes used to train the Mouse were not the right mazes and this is because the trainer did not understand the mazes that were being used.

The second example is a thought exercise about basketball. I'll go out on a limb here and say that anyone reading this knows what basketball is. The goal of basketball is to put a ball through a basket with a hole in it. The adult height for the basket is 10 feet. To accomplish this, let's say from the free throw line to keep this simple, and get good at it you have to practice (train). Every step in making the free throw involves hundreds/thousands of small decisions (steps) and making the wrong decision will cause you to miss. Think about the first decision you must make, the technique, do you shoot with both hands, your left, your right, or even underhanded. That decision is simply the first step in making the shot. By the way, the cheese is not making a goal, the cheese is the shot. A shot that goes in is a cheese with a higher value. Missing a shot is still cheese but just not worth as much.

Now imagine that there is a competition coming up that if you make 3 in a row you win $100. You hit your local gym and for an hour every day you practice your free throws. At the end of that practice you ideally should be pretty good at scoring (getting the highest value cheese). What would happen though, if you showed up and the basket was now 12 feet above the floor? Do you think you could still make 3 in a row? Maybe, but it will be a lot harder to do so.

What makes the 12 foot goal hard is that the maze you have to navigate through to get to the best cheese is different enough from the mazes you trained on that trying to navigate those differences means you have to fall back on something other than what you trained on. Remember, you can always find the cheese using the brute force approach (always turn left). But that approach can mean you take many more steps than necessary which means you'll get a lower score.

A Mouse created for one type of maze may simply not work in another or may be so expensive in terms of time or food that the cost overwhelms the value of the cheese and the score suffers.

Chatting for Fun and Profit

Let's alter the rules of The Mouse and The Maze just a bit. Here are the new rules:

1. Instead of single pieces of cheese being placed in the maze, instead we put clusters of cheese where the values of the cheese are all close to one another. For example, I might have a cheese worth 100 and around it I put a few cheeses worth between 90 and 95.
2. At the beginning of the game, the mouse is shown the maze but does not get a copy of the maze to keep, i.e. it must memorize what it has seen. The Mouse cannot make any decisions at this point (take any steps) but the Mouse can memorize the map.

What kind of Mouse would you use for this sort of maze and how would you prepare the Mouse for a competition? Here are a few ideas;

1. The Mouse needs to be able to memorize the map that it has been shown, but how much memory should our Mouse use? Remember, bigger brains (more memory) require more food, which means a lower score. Which part of the map gets memorized can be important as well.
2. It will be helpful to know the distance to the cheese. We can do this by triangulating (checking the direction from different points) or at least be able to guess how far away it is. This will help knowing which cheese the Mouse should be trying to get to maximize the score but getting this triangulation takes steps, and more steps means a lower score.

Memorizing the entire map seems like an obvious strategy but that only works with small maps. There are other techniques that we could use as well where the entire map doesn't have to be memorized but instead the Mouse is trained to memorize features about parts of the map that might help it navigate. For example, if there is a long corridor in the maze that ends in a dead-end, the Mouse does not have to memorize each and every step of the corridor to know it should not go that way. The Mouse only needs to memorize that it should not go down that corridor at all.

If we have past example mazes that will be similar to the maze that the Mouse will be using in one of our hypothetical competitions, we might even assign percentages to some of those observations. For example, say that in 80% of the maps we train the Mouse on a corridor that is longer than 5 steps is the best way to get to a cheese that is in the direction of the corridor. We could create a rule for the Mouse that says always take that corridor. However, if we spend more time analyzing and training our Mouse, we might find that for that remaining 20%, if the Mouse has already taken 100 steps, then the corridor is a dead end. Having that little extra bit of information about the maze that the Mouse is likely to navigate could mean fewer steps which means a higher score. One other possibility is to give the mouse a coin and have the mouse flip the coin (or roll a die) whenever the answer isn't obvious or whenever probabilities (probabilities are just percentages) are present. We could, in the 80% case, have the Mouse take the corridor but only do so if the Mouse rolls a 4 or below on a 20 sided die.

The game I just described and the ways to prepare the Mouse to navigate the maze are a simplification (a model) of LLMs. Each cheese in the maze is a LLM response. Some responses are better or worse but a lot of that depends on where you start. Showing the Mouse the maze at the start, that's your prompt. The rest is just the Mouse finding a path through the maze to get the highest score; finding a high value cheese, taking a little time as possible, eating as little food as possible, in as few steps as possible. Even something like RAG is just playing this exact game. RAG adds to the prompt (the mouse memorizes more of the maze) but does require more context length (food) so there has to be a balance. For cases where the LLM is being used creatively, sometimes hallucinating is good (high score) so giving the Mouse extra food is not necessarily a good thing. When the LLM is being used in a way that required accuracy, you need a more expensive Mouse.

Any AI is just a variation of The Mouse and The Maze

Just like the game Monopoly™, we can add new rules or tweak existing ones (think Free Parking). Fundamentally, we can't change the basics, otherwise it's a different game but the ways to play the game are as varied as people are. There is a Mouse, there is a maze, and somewhere in the maze there is some cheese. The goal of the game is always to get the highest score. For AI, there is the algorithm, the search space, and the goal.

To wrap this up, here are a few variations of The Mouse and The Maze that could be played. Think about how a Mouse might navigate through these types of mazes. What strategies would a Mouse need to use? Always remember that it's the highest score that wins; not that the best cheese is found or that the cheese is found in the fewest steps. Only the score matters.

Mystery cheese:
You don't know the values of the cheese, you just know where they are.

Caves of Mazes:
There are several mazes, each connected by long corridors.

Geiger Cheese:
The Mouse has to use a device to locate the cheese. The device tells the Mouse in which direction the cheese is but only gives the value of the cheese divided by the distance, for example a cheese with a value of 50 that is 25 away shows up on the device with a value of 2. The closer you are (the shorter the distance) the higher the value. So if you were 1 away from a cheese worth 50, the value would show 50.

Open Room:
The maze is just a big open room and there are several cheeses. Remember, the game ends when the Mouse finds the first cheese so in this game, the Mouse must make a choice between a close, lower valued cheese and finding a way to get around that cheese to get to a higher value. The game is only about score.

Multi-dimensional:
The maze is not 2 dimensional but 3 (think a building) or higher or a maze where there are portals that jump the Mouse from one spot in the maze to another.

One way corridors:
In some parts of the maze (or maybe all), the Mouse cannot go backwards, only forward. There may be places in the maze where once the Mouse starts down a path it gets trapped if there is a dead-end and the game ends.

Gremlin maze:
There is another player in the game whose purpose is to lower your score. While the maze cannot be changed, the player may do things like confuse the Mouse which direction it can go or try to slow the Mouse down.