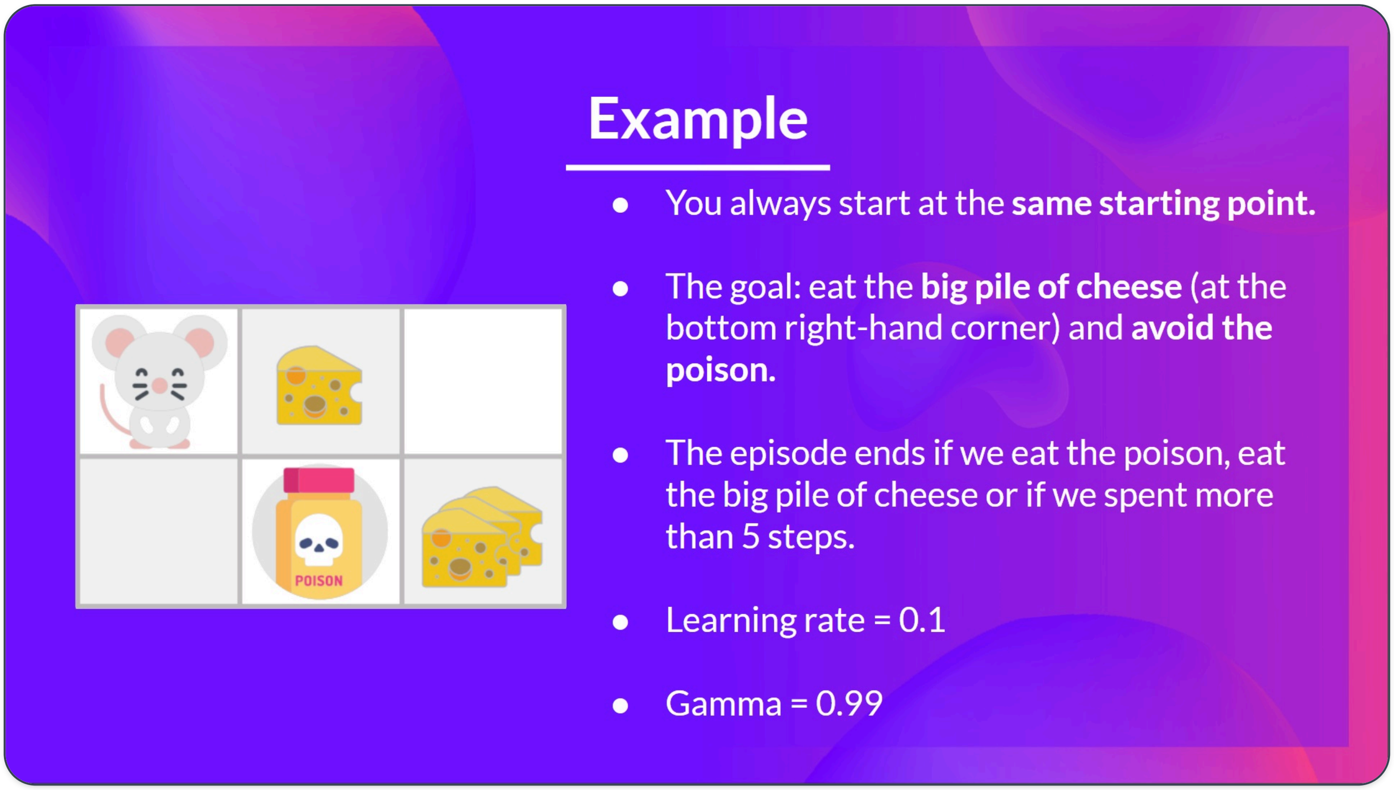
A Q-Learning example

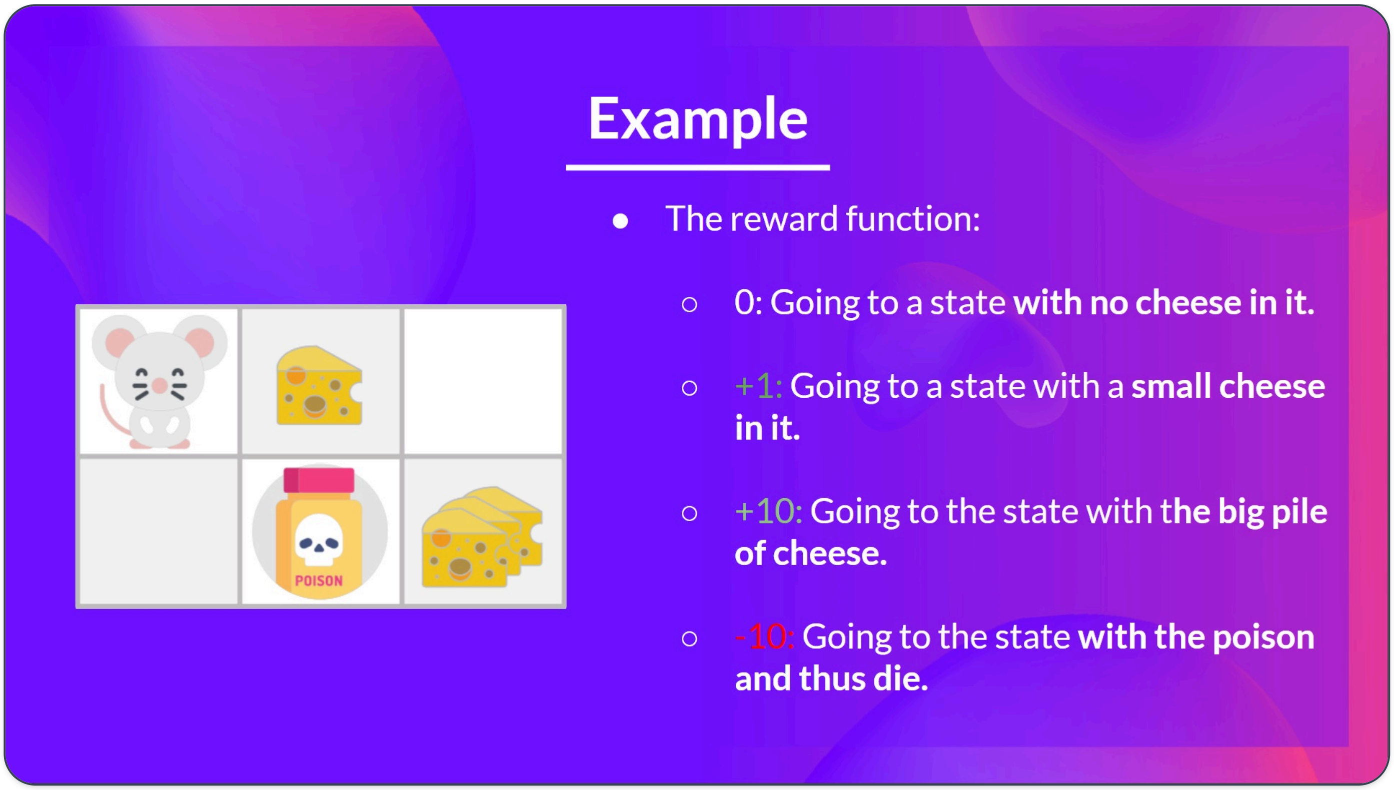
To better understand Q-Learning, let's take a simple example:

- You're a mouse in this tiny maze. You always **start at the same starting point.**
- The goal is **to eat the big pile of cheese at the bottom right-hand corner** and avoid the poison. After all, who doesn't like cheese?
- The episode ends if we eat the poison, **eat the big pile of cheese**, or if we take more than five steps.
- The learning rate is 0.1
- The discount rate (gamma) is 0.99



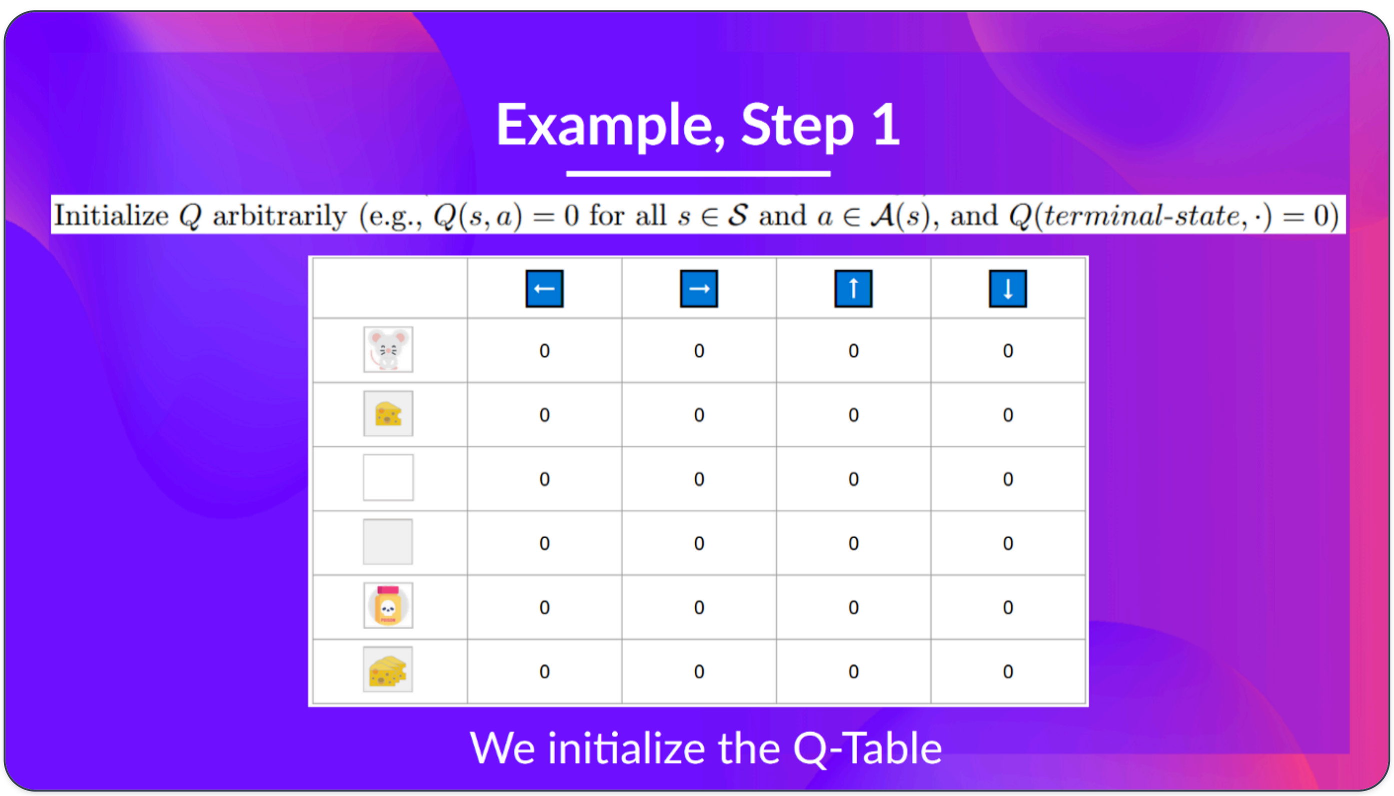
The reward function goes like this:

- **+0:** Going to a state with no cheese in it.
- **+1:** Going to a state with a small cheese in it.
- **+10:** Going to the state with the big pile of cheese.
- **-10:** Going to the state with the poison and thus dying.
- +0 If we take more than five steps.



To train our agent to have an optimal policy (so a policy that goes right, right, down), **we will use the Q-Learning algorithm.**

Step 1: Initialize the Q-table



	←	→	↑	↓
	0	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0

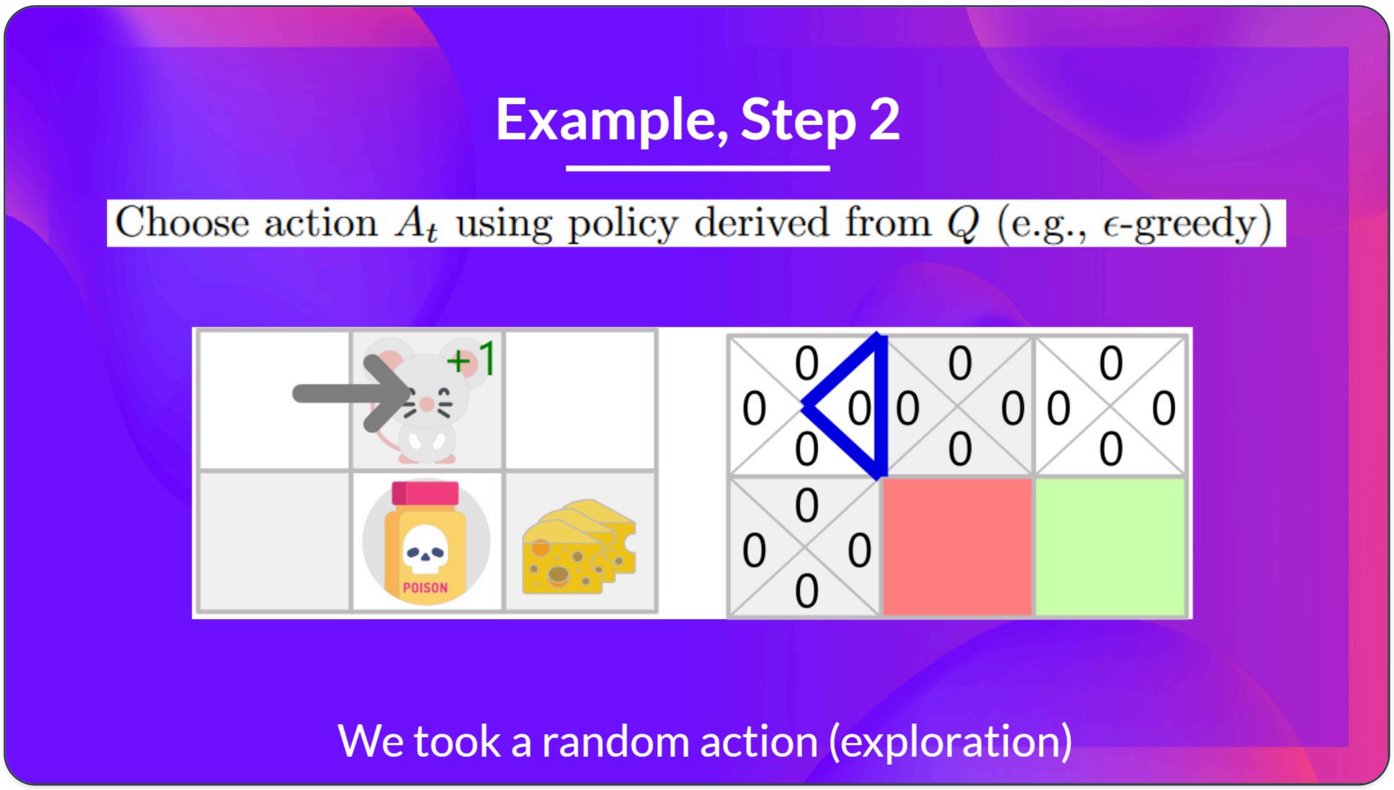
So, for now, **our Q-table is useless**; we need **to train our Q-function using the Q-Learning algorithm.**

Let's do it for 2 training timesteps:

Training timestep 1:

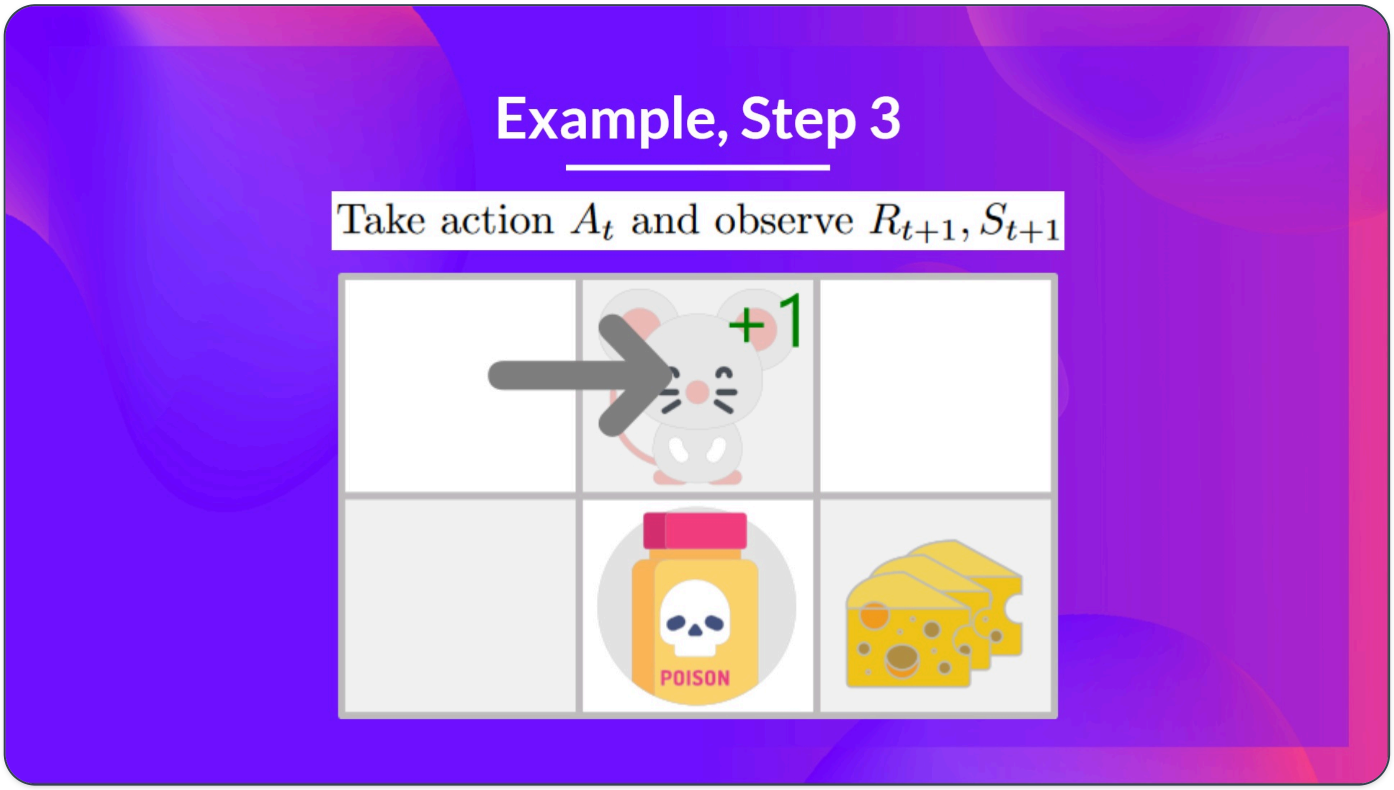
Step 2: Choose an action using the Epsilon Greedy Strategy

Because epsilon is big (= 1.0), I take a random action. In this case, I go right.



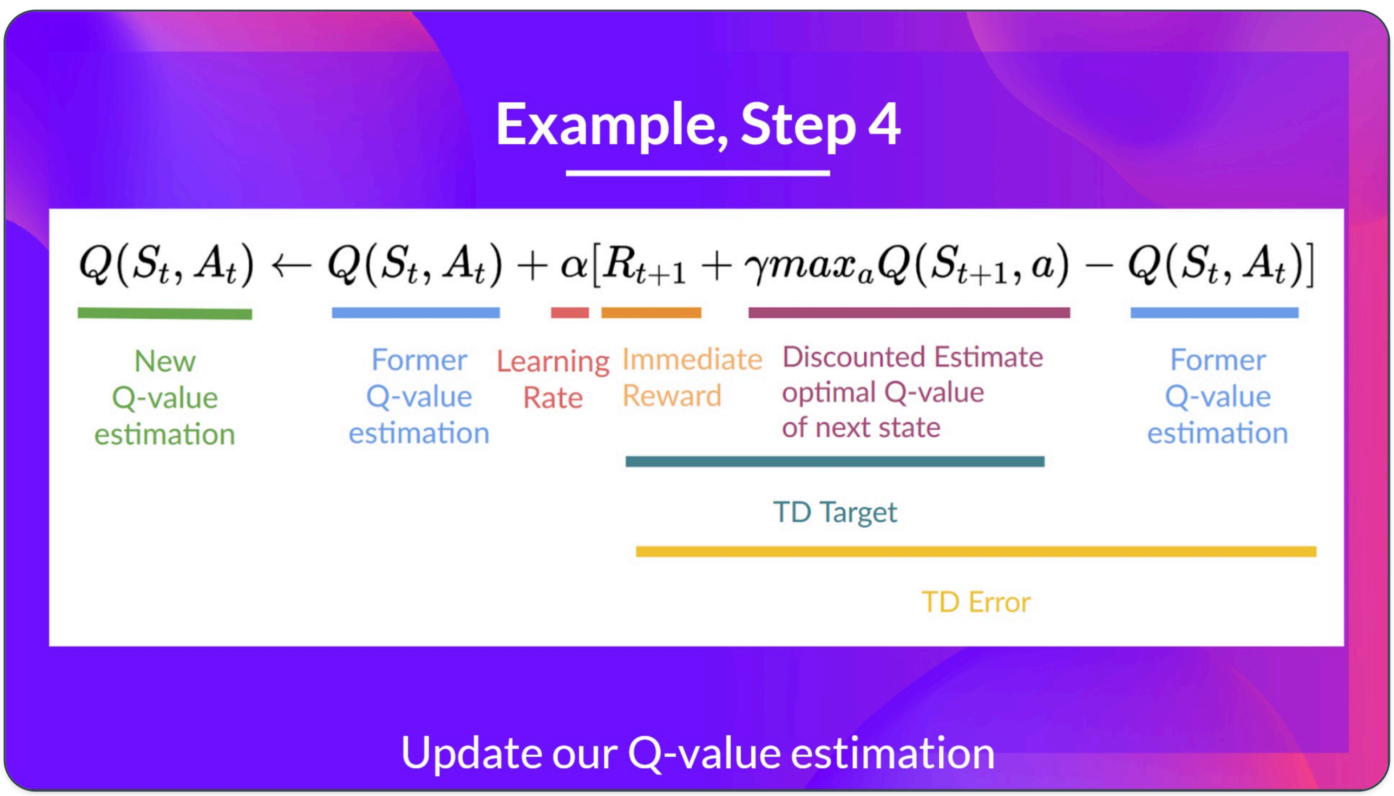
Step 3: Perform action At, get Rt+1 and St+1

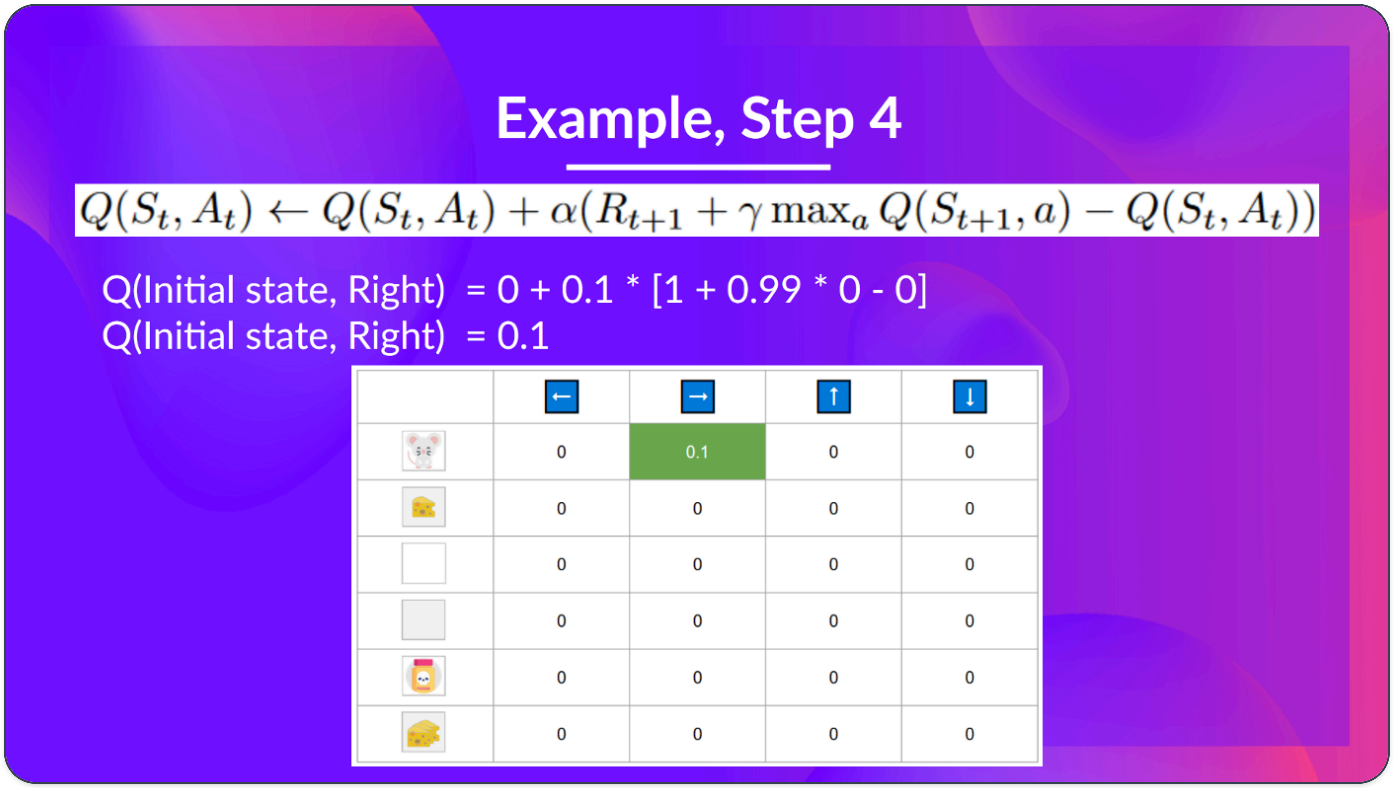
By going right, I get a small cheese, so $R_{t+1} = 1$ and I'm in a new state.



Step 4: Update Q(St, At)

We can now update $Q(S_t, A_t)$ using our formula.



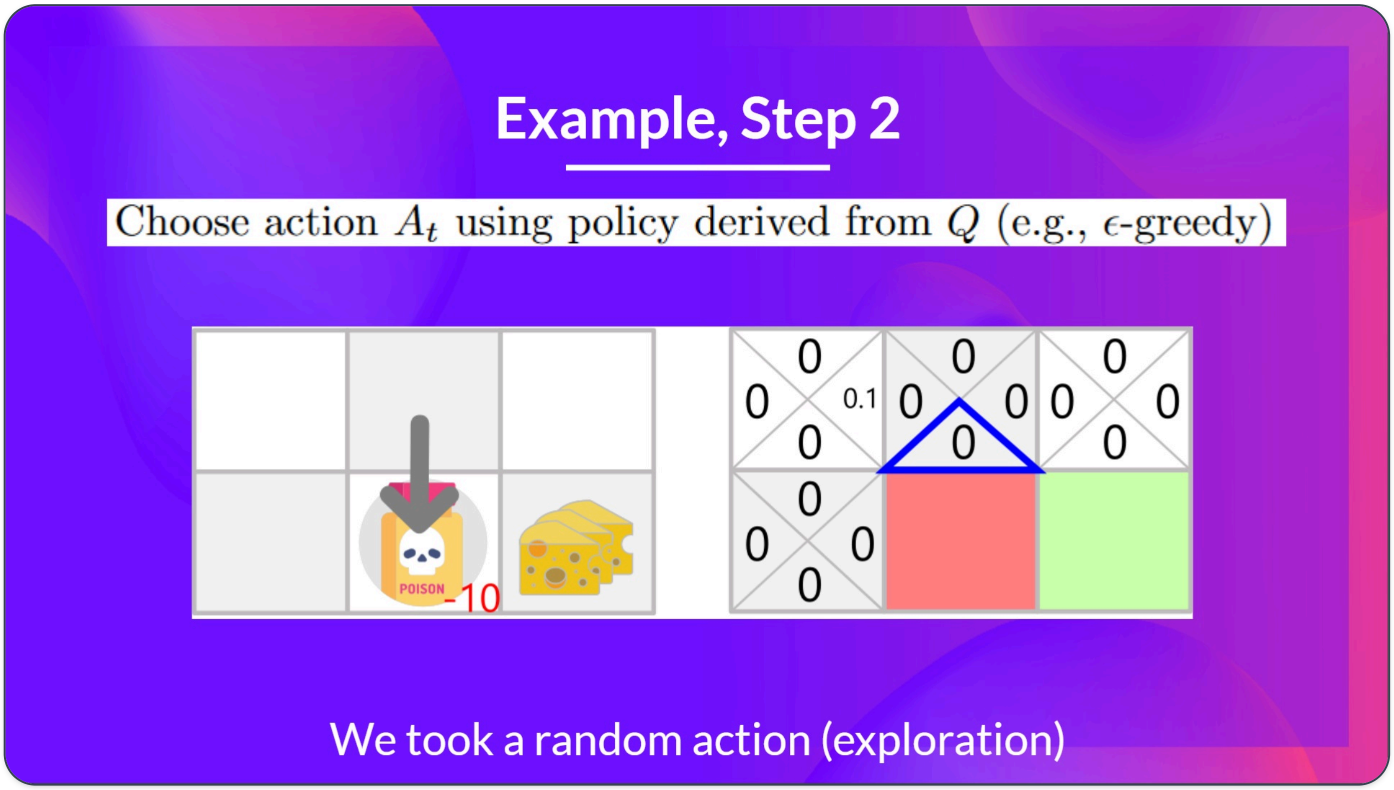


Training timestep 2:

Step 2: Choose an action using the Epsilon Greedy Strategy

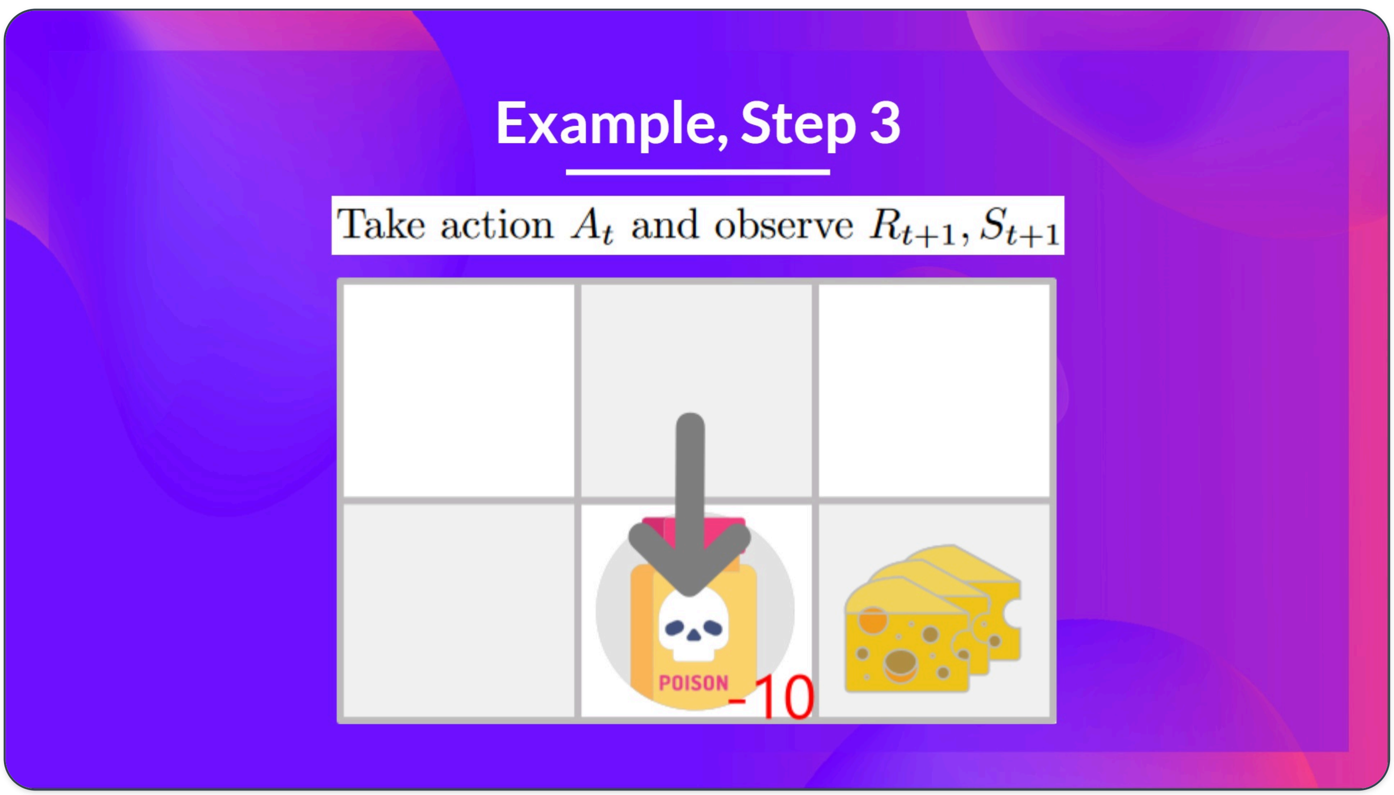
I take a random action again, since epsilon=0.99 is big. (Notice we decay epsilon a little bit because, as the training progress, we want less and less exploration).

I took the action 'down'. **This is not a good action since it leads me to the poison.**

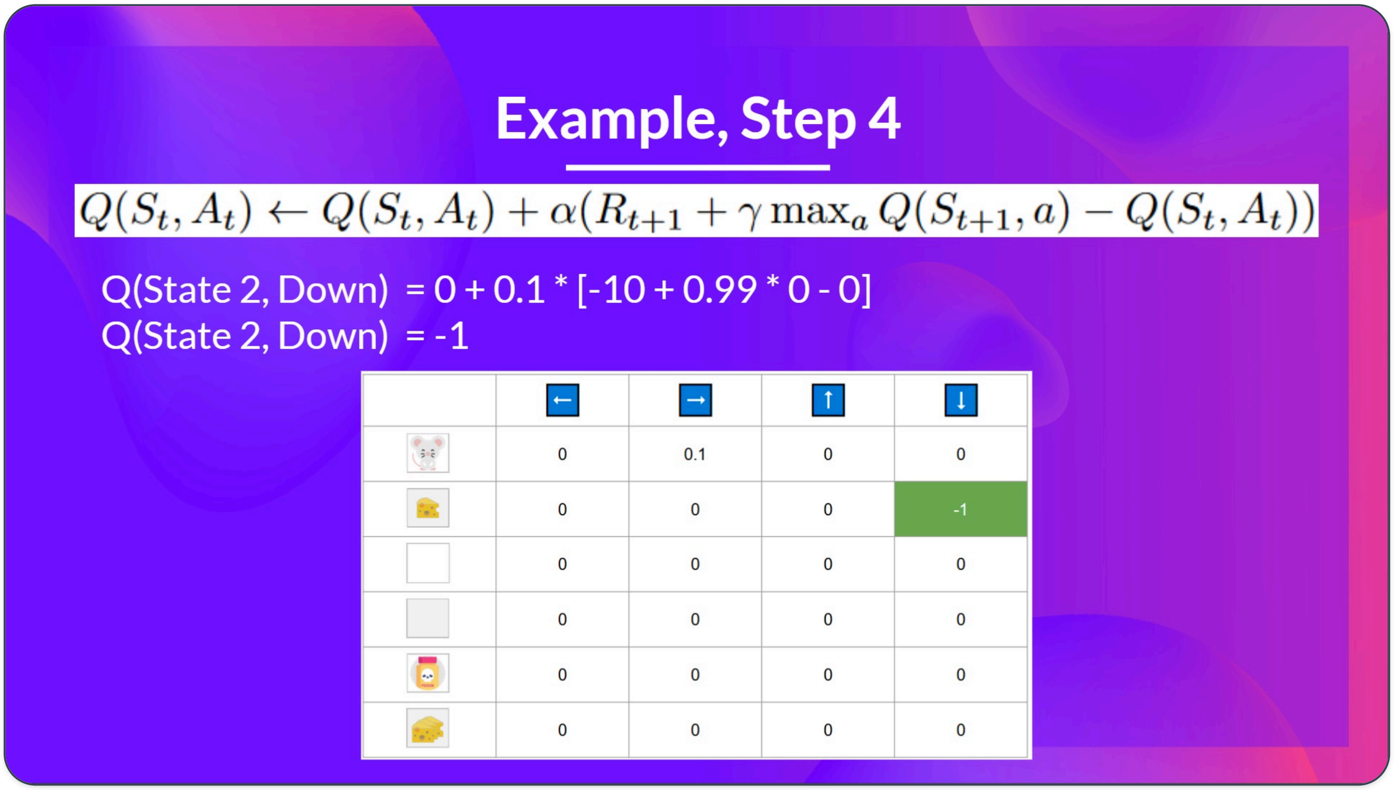


Step 3: Perform action At, get Rt+1 and St+1

Because I ate poison, I get $R_{t+1} = -10$, and I die.



Step 4: Update Q(St, At)



Because we're dead, we start a new episode. But what we see here is that, **with two explorations steps, my agent became smarter.**

As we continue exploring and exploiting the environment and updating Q-values using the TD target, the **Q-table will give us a better and better approximation. At the end of the training, we'll get an estimate of the optimal Q-function.**

← Introducing Q-Learning

Q-Learning Recap →