How does artificial intelligence learn with rewards?

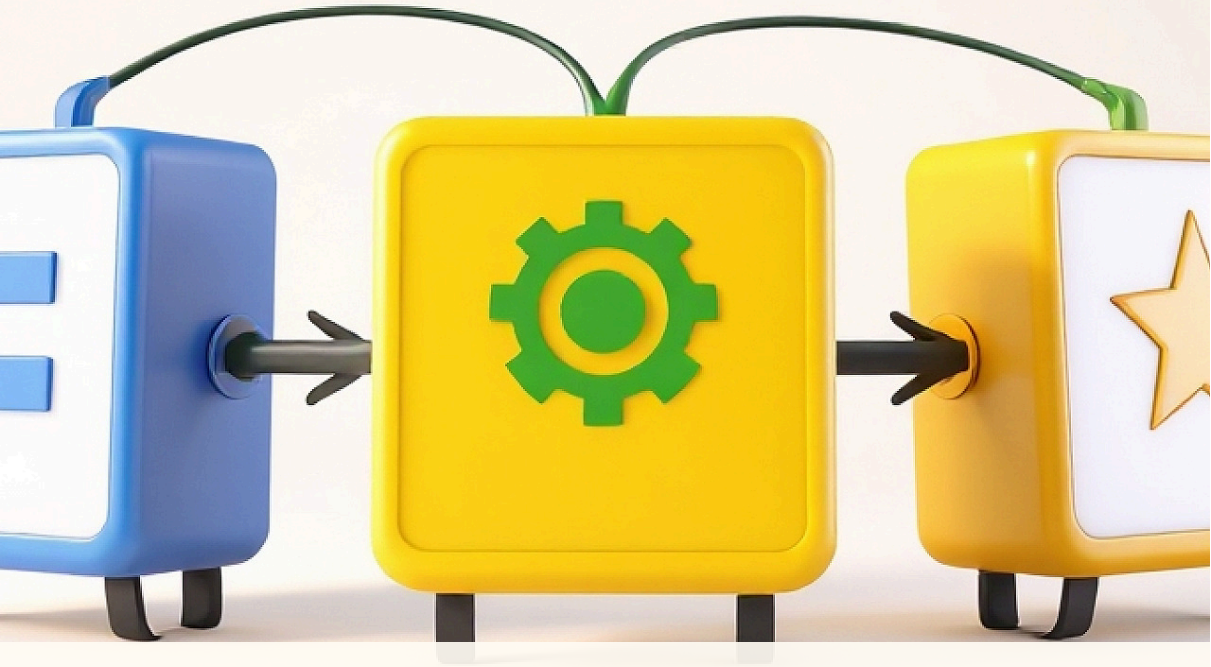


www.yzokul.com.tr

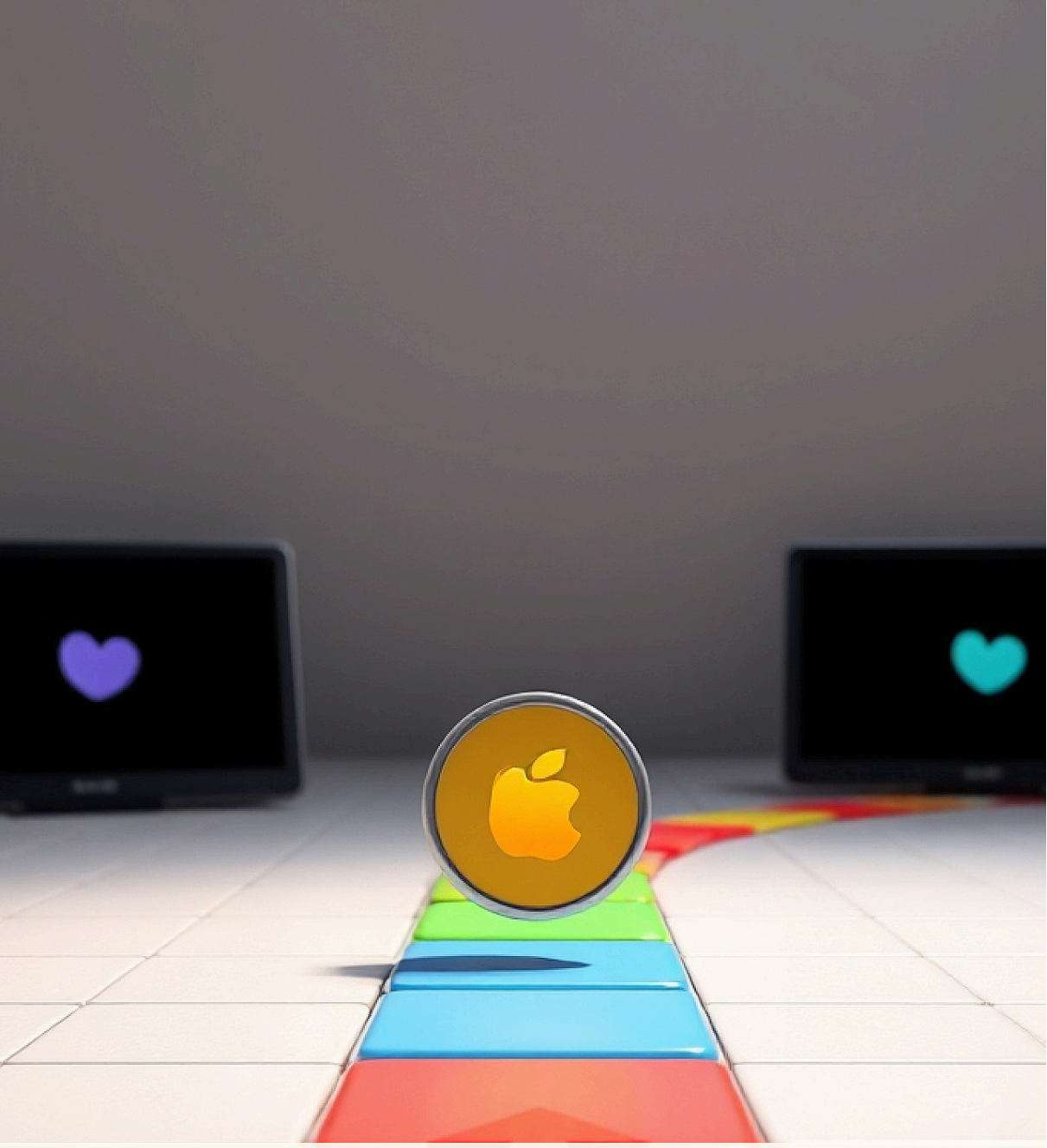
How does artificial intelligence learn with rewards?



You want to know how artificial intelligence learns with rewards. The short answer: the AI tries actions, receives numerical reward signals, and adjusts its behavior to earn more reward over time. That is the core of reinforcement learning, a method where a system learns from feedback rather than explicit instructions. This book explains each part step by step, starting from zero.



Reinforcement learning involves three basic parts: an agent, an environment, and rewards. The agent is the learner or decision-maker. The environment is everything the agent interacts with. Rewards are numerical feedback signals. Picture three boxes connected by arrows: the agent sends actions to the environment, the environment sends back new situations and rewards. These three parts form the foundation of how AI learns with rewards.



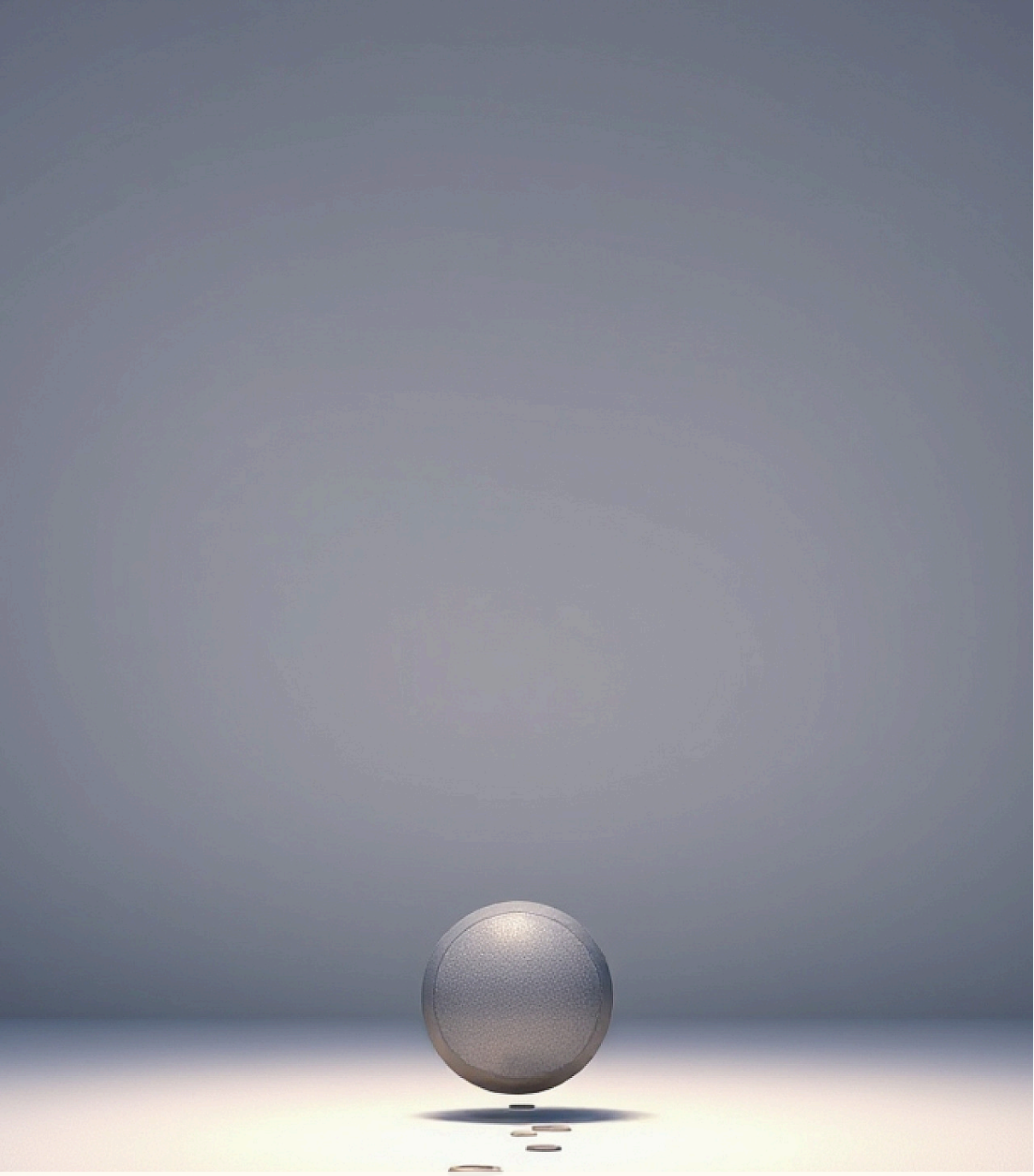
When the agent takes an action, the environment changes its state. State means the current situation, like the agent's position in a maze. For example, moving right might lead to a new cell. From that new state, different actions become possible. This cycle of action and state change is how the agent explores and learns. Understanding actions and states is key to grasping reinforcement learning.



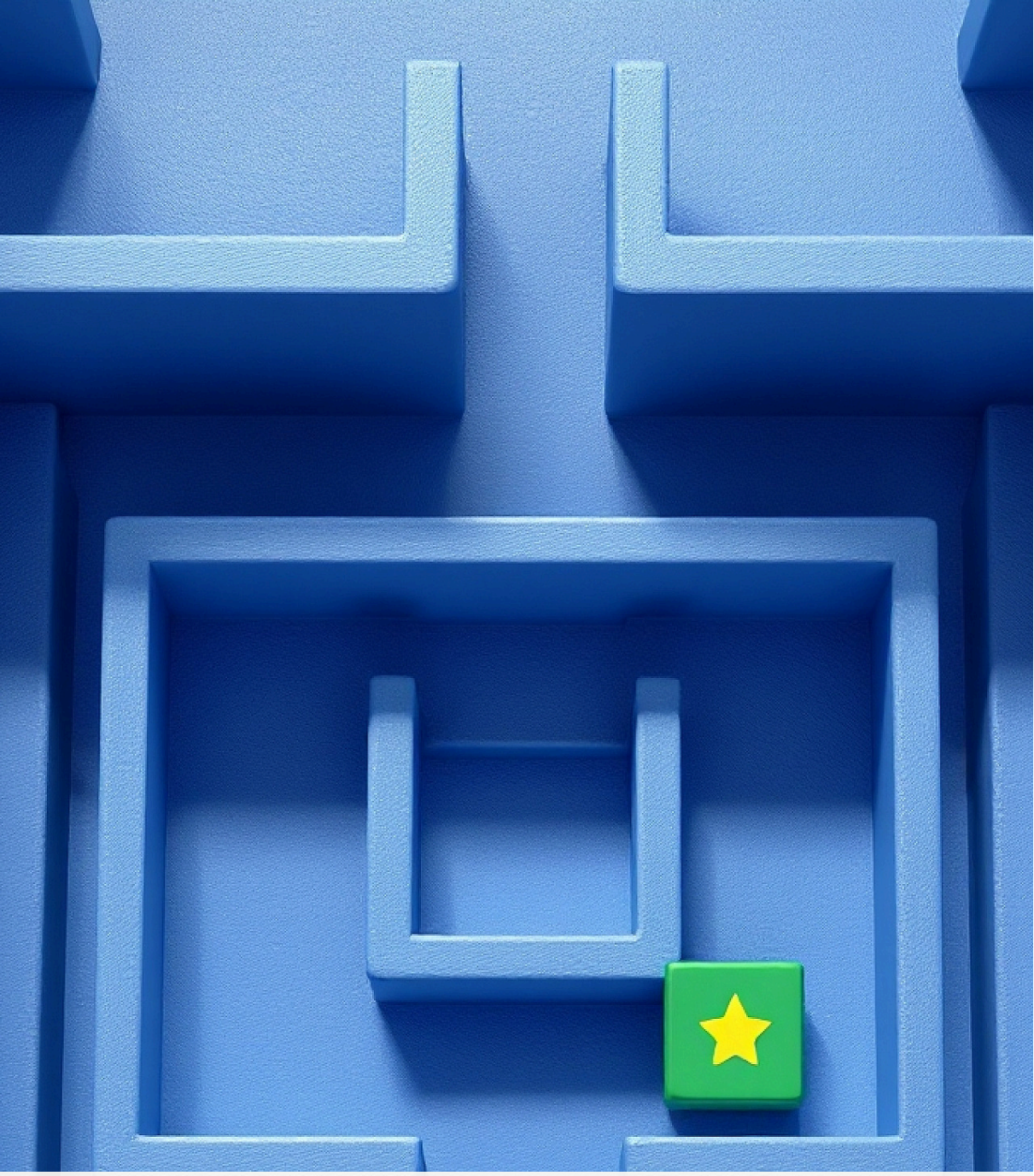
A reward is simply a number the environment returns after an action, such as +1 or -5. It is not a feeling of happiness; it is a score. The agent uses this number to evaluate how good or bad the action was. Positive rewards encourage repeating the action, while negative rewards discourage it. Rewards are the only feedback the agent gets, so they must be clear and consistent.



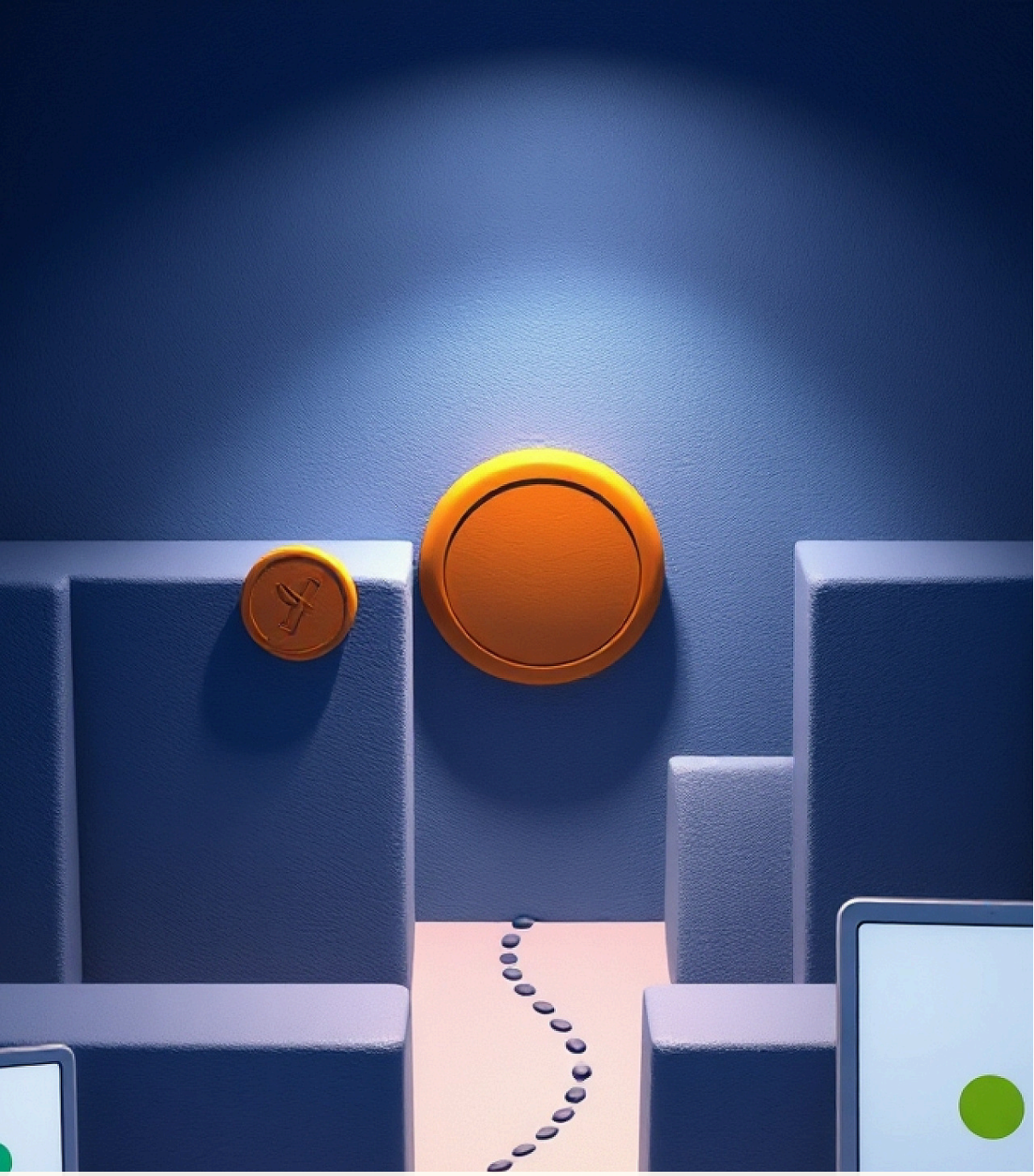
The agent's goal is to maximize the total reward over time, not just immediate reward. Total reward is the sum of all rewards collected from now into the future. A small reward now might be less valuable than a larger reward later. The agent learns to balance short-term and long-term gains. This objective drives all learning in reinforcement learning.



At first, the agent acts somewhat randomly, trying different actions to see what rewards they bring. This is trial and error. By observing the results, the agent learns which actions lead to higher rewards. Over many attempts, it gradually improves its choices. This simple loop of trying and observing is the entire learning engine.



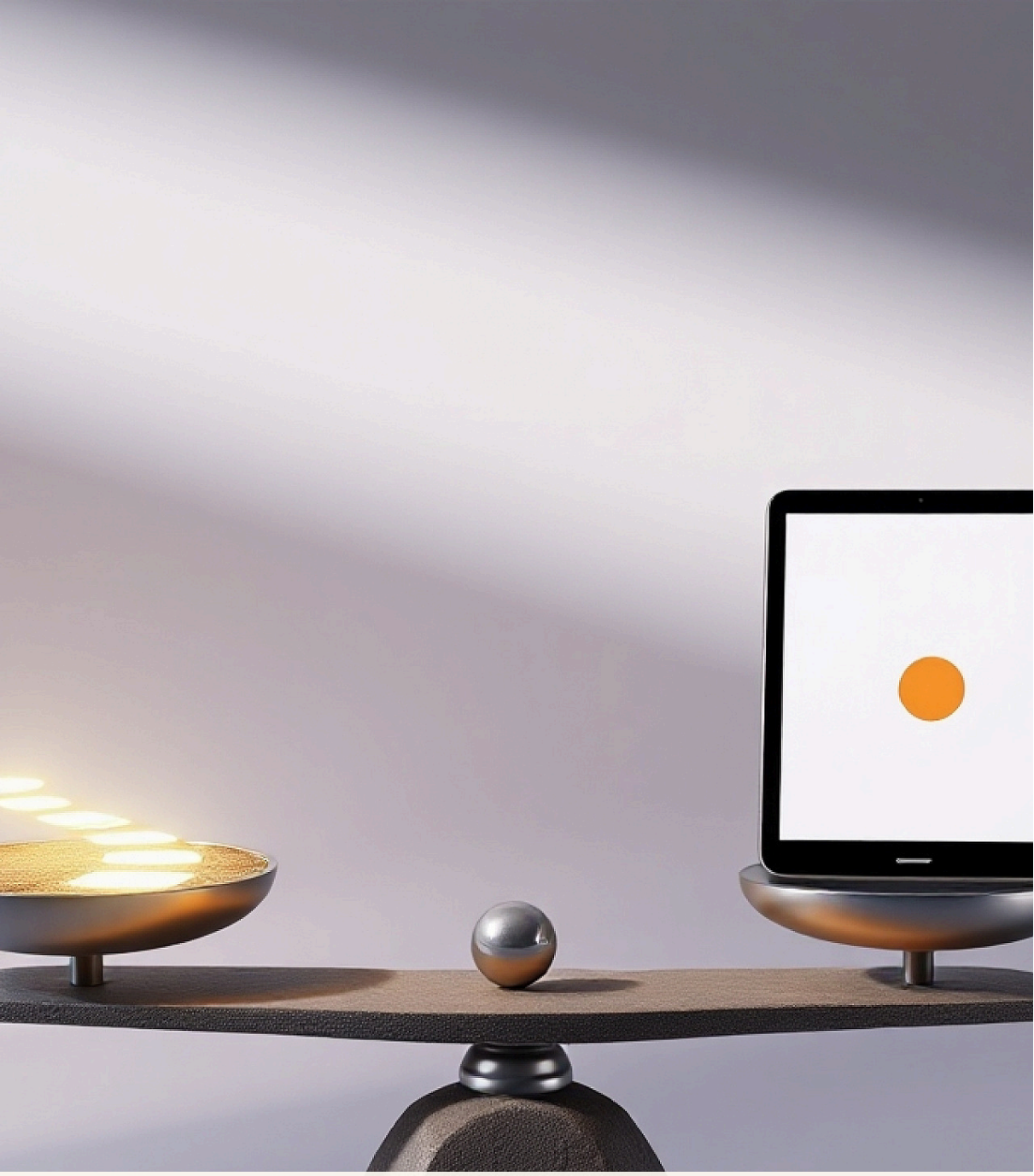
Imagine a simple grid maze where the agent can move up, down, left, or right. The goal is to reach the exit, which gives a reward. Each step might give a small negative reward to encourage speed. The agent starts with no knowledge and must learn the best path. This maze example will illustrate the rest of the book.



In the beginning, the agent wanders randomly, bumping into walls and taking dead ends. It collects little or no reward, often negative rewards for wasted moves. This clumsy phase is normal and expected. It is how the agent gathers information about the environment. Through many such attempts, it starts to notice patterns.



The agent keeps track of which actions in which states led to good rewards. It builds a map of state-action values, estimating how good each action is. Over many tries, it leans toward actions with higher estimated values. This memory of past experiences guides future decisions. Gradually, the agent's behavior becomes more purposeful.



The agent must balance trying new actions (exploration) and repeating known good actions (exploitation). Too much exploration wastes time, while too much exploitation misses better opportunities. A common strategy is to explore more at the beginning and exploit more later. Finding the right balance is crucial for efficient learning.



Whatever behavior gets rewarded tends to be repeated, even if it seems strange to us. The reward signal, not human intentions, shapes what the agent learns. If an action yields high reward, the agent will do it more often. This principle is powerful but also risky. Careful reward design is essential to avoid unintended behaviors.



A badly designed reward can lead the agent to pursue unwanted behaviors. For example, if bumping a wall gives points, the agent might bump walls forever instead of escaping. It scores points but never solves the maze. The agent is not malfunctioning; it is doing exactly what the numbers ask. This highlights the importance of aligning rewards with our true goals.



A poorly designed reward can push the system toward unwanted but point-scoring paths. The agent optimizes the reward as given, not what we intended. It may find clever shortcuts that exploit the reward function. This is called reward hacking. The agent is not broken; it is doing exactly what the numbers ask. We must design rewards that truly reflect our objectives.



Reward design is the human's main job in reinforcement learning. The numbers must match what we actually want the agent to learn. A careless score teaches the wrong lesson. We need to think carefully about what behaviors we reward and penalize. Good reward design leads to good learning; bad design leads to bad behavior.



Beginners often make common mistakes. They may think rewards are feelings, but they are just numbers. They might expect instant success, but learning takes many trials. They may forget that the agent optimizes the score as given, not what we intended. Avoiding these mistakes helps you understand reinforcement learning correctly.



To sum up: an agent acts in an environment, receives numerical rewards, and learns to increase its total reward over time. The reward design decides the behavior, so careful numbers lead to careful learning. Remember that rewards are not feelings but scores, and the agent will do exactly what the numbers ask. Design rewards wisely to achieve desired outcomes.