**Design Defense**

Humans approach problems like navigating a maze using intuition, past experiences, and reasoning. When they encounter obstacles, they adjust their approach, relying on learned strategies or trial and error. For example, a human might backtrack and try a different path when they reach a dead-end, making decisions based on logic or memory. This process is flexible, with humans quickly adapting to changing conditions (Sutton & Barto, 2018).

In contrast, a machine learning agent starts with no prior knowledge of the environment. It must learn from its own interactions and the consequences of its actions. At first, the agent explores the environment by making random decisions or by following a policy of experimentation. Over time, as the agent encounters different states and experiences the outcomes of its actions, it begins to refine its strategy based on the feedback it receives. This process enables the agent to gradually improve its pathfinding performance, ultimately converging on an optimal solution through repetition and learning (Sutton & Barto, 2018).

Humans solve pathfinding problems by exploring different routes, quickly discarding paths that don't lead to the goal, and focusing on successful ones. They form hypotheses about potential paths, using knowledge of their surroundings and past experiences. If a path is blocked, humans adapt their strategies, learning from their actions and recalculating based on what they already know (Sutton & Barto, 2018).

In contrast, an RL agent starts without any knowledge of the environment and must explore to understand which paths are effective. It doesn’t have a mental map like a human does; instead, it tests different actions, receives rewards for good moves, and is penalized for bad ones. As the agent accumulates more experience, it fine-tunes its decision-making process and eventually develops a strategy that helps it reach its goal more efficiently (Mnih et al., 2015).

A key concept in reinforcement learning is balancing exploration and exploitation. Exploration involves trying new actions to gather more information, while exploitation refers to using actions that have already proven to be effective. At the start of training, exploration is vital, as the agent needs to discover what works and what doesn’t. Once it gathers enough knowledge, the focus shifts to exploitation, where it uses the most successful strategies (Sutton & Barto, 2018).

The right balance between exploration and exploitation is crucial. Too much exploration can lead to inefficiency, while excessive exploitation may prevent the agent from finding new and better paths. An optimal mix allows the agent to refine its behavior while still leaving room for discovering improvements (Sutton & Barto, 2018).

Reinforcement learning excels in pathfinding because it enables the agent to learn by doing, without needing a predefined solution. The agent interacts with the environment, receiving feedback that informs its next actions. If a move brings the agent closer to its goal, it is rewarded, reinforcing the action. If a move results in failure, the agent is penalized, discouraging that path in the future (Sutton & Barto, 2018).

Unlike traditional algorithms that may require a complete map or set of instructions, RL allows the agent to learn from experience, making it adaptable to dynamic environments. The agent does not rely on having prior knowledge about the path; it learns which actions lead to success through its own trial and error (Mnih et al., 2015).

Deep Q-Learning (DQN) enhances the standard Q-learning algorithm by using deep neural networks to approximate the value of actions in various states. Traditional Q-learning stores values for each state-action pair in a table, which can become impractical for large or complex environments. DQN, on the other hand, allows the agent to handle environments with large state spaces by leveraging neural networks to generalize across these states, predicting the expected future rewards for different actions (Mnih et al., 2015).

One of the key features of DQN is experience replay, a technique that helps stabilize the learning process. Instead of using the most recent experiences to update its knowledge, the agent stores previous interactions in a buffer and samples them randomly. This reduces correlations between consecutive updates and ensures the agent learns from a variety of experiences. By continuously refining its decision-making using neural networks, the agent becomes more efficient at finding the optimal path and reaching its goal, like locating the treasure in a game, even in complex environments (Mnih et al., 2015; Watkins & Dayan, 1992).

**References**

* Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). Human-level control through deep reinforcement learning. <https://user.ceng.metu.edu.tr/~emre/resources/courses/AdvancedDL_Spring2017/DQN_Muhammed.pdf>
* Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). <http://incompleteideas.net/book/RLbook2020.pdf>
* Watkins, C. J. C. H., & Dayan, P. (1992). *Q-learning*. *Machine Learning*, 8(3), 279-292. <https://doi.org/10.1007/BF00992698>