Reinforcement Learning in a Dynamic Maze

Jesse Victors

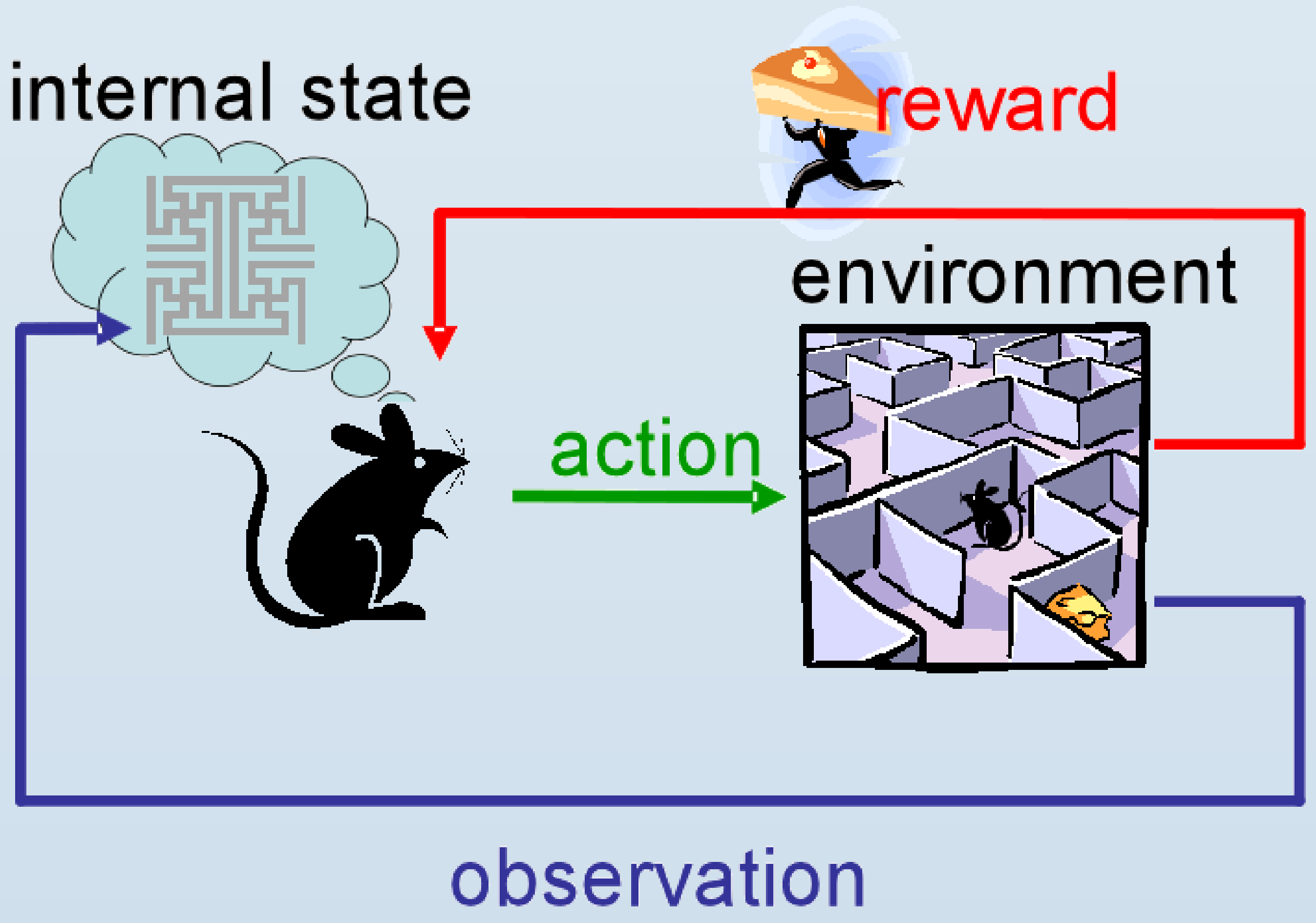
Department of Computer Science
Utah State University

Abstract

Reinforcement Learning: How Flexible is it?

Reinforcement learning is a useful technique for developing intelligent systems and has a wide range of applications. In this project, I explore the efficiency and flexibility of RL. How quickly can an agent driven by RL solve a simple system? Can it optimize its solution? How quick is it to adapt when a learned solution is no longer possible? To explore these questions I built a mouse-in-a-maze simulator.

The simulator randomly generates a maze and places a mouse at the entrance. The mouse's goal is to find the exit at the other side. It only concerns itself with open and neighboring cells around it and makes its navigation decisions based on its memory of them. Even with relatively simple rules governing its behavior, I observed that the mouse was often very fast at finding the exit and on subsequent explorations that it quickly chose an optimal path. Furthermore, when I blocked the learned path, the mouse found and optimized alternative routes. This suggests that reinforcement learning is a viable approach for maze solving.



Background

Reinforcement learning is a technique in machine learning wherein software agents learn to take actions in their environment such that they maximize a cumulative reward. To achieve this, these agents must find a balance between exploration of uncharted territory and exploitation of current knowledge. They learn the consequences of their actions by trial and error and are fed a reinforcement signal indicating the success or failure of an action they take. Over time, reinforcement learning can often lead to optimal or near-optimal solutions.

Reinforcement learning methods are used in a wide range of applications including systems control, game playing, and simulations of animal conditioning and learning.

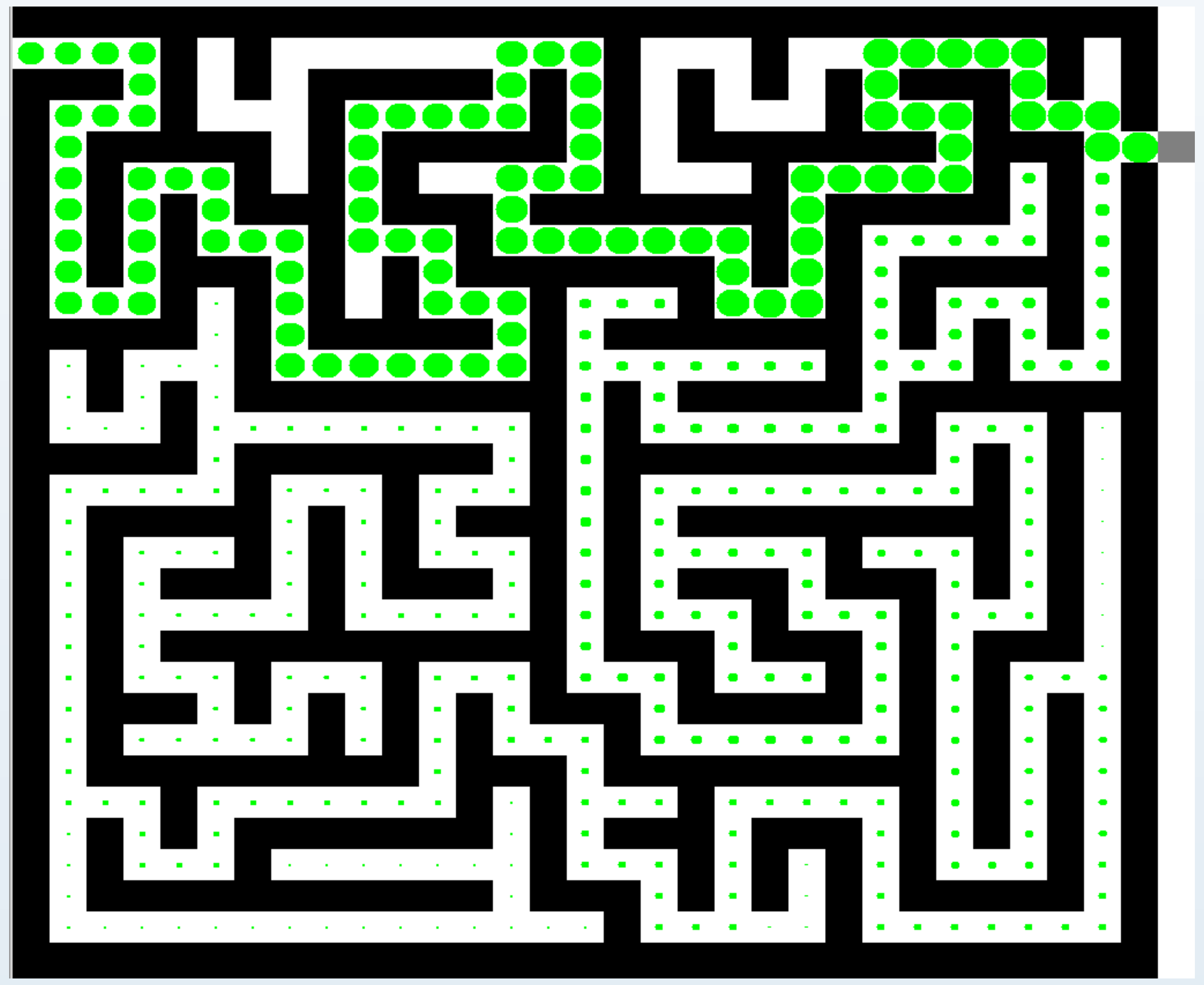
Methods

The maze itself was randomly generated by first constructing a 64-by-64 grid of walls and then applying a recursive depth first search algorithm to open up paths. The entrance of the maze was statically placed in the upper-left corner and the right-most column became the goal.

The mouse was then placed at the entrance of the unexplored maze. As it navigated, it left a trail of memory behind that geometrically decayed over time. In this explore mode, the mouse preferred moving to a neighboring cell that was least remembered. When it completed the maze, the mouse was repositioned at the entrance.

During the next round of exploration the mouse continued to explore the maze, but preferred neighbor cells that were remembered the most. In this way, the mouse was rewarded by its memory of the cells. If the remembered path became blocked, the mouse switched to explore mode and blazed a new path to the goal. Its use of the memory trail improved navigation and flexibility.

Results



When presented with an unexplored maze, the mouse moved aggressively through the paths, preferring unexplored territory over known areas and only backtracking when dead ends were encountered.

Here, its memory of explored cells is represented by green circles; the larger circles indicate stronger memories. In the example on the left, the mouse found the exit after initially exploring most of the maze.

In subsequent explorations the mouse exploited its previous knowledge to optimize the path. Although it had originally solved the maze with a longer route, the path intersects itself near the exit. The mouse's preference for stronger memories caused it to jump across, avoiding wasting time with the much longer path.

When a wall was introduced that blocked the learned route, the mouse recognized the dead end and began exploring the maze for an alternative solution. Once one was found, the mouse exploited that learned route and optimized it.

Conclusion

Reinforcement learning is a useful technique when designing intelligent systems. Here I demonstrate that RL methods and memory can be very effective for maze solving. These methods also lead to emergent behavior, such as the on-the-fly optimization of the path through the maze. In almost all cases, RL lead to the optimal solution even when parts of the maze remained unexplored. The mouse was also able to adapt; when a learned path was no longer possible, it used its memory to find an alternative route and later optimized it.

This behavior has many applications. Robotic systems may use RL to develop near-optimal solutions to complex problems. If components break or become worn, the robot may be forced to find an alternative solution under this constraint. Humans and other animals often use reinforcement learning to understand how to do a task, which we may have to relearn if we are injured. It is very interesting to demonstrate that simple learning techniques can lead to such behavior.