Module 7 – Project Two

Duvan Diaz

CS 370

SNHU

February 27, 2025

Design Defense – Project Two

For project two, the following writings will support and explain how I was able to generate a Deep Q Network in python, so that the pirate agent was able to navigate the maze to ultimately reach its end goal which was to find the treasure.

When comparing how a human would reach this goal, against how a machine system approaches this, we need to understand that humans and machines approach to problem solving are different from each other. The human approach relies on patter recognition, prior experience, and intuition. For example the human approach would first identify the starting point and the end point which would be the treasure location, Next, the human would map out possible routes after visually scanning, then start navigating via trial-and-error, and if a dead end is encountered the human would backtrack and map out a different route to try, and continue like this until reaching the end goal (treasure).

Now for the machine approach, the system would work based on a structured learning process, navigating through multiple tries and accordingly adjusting the actions taken next based on rewards/penalties (feedback). The machine agent would start at a set position in the maze grid, then it would start exploring the available actions based on the maze environment’s state. Next, using the Deep Q-values and reinforcement learning, the agent would navigate and adjust the routes/path accordingly through multiple tries and reducing the penalties, therefore increasingly improving the rewards and encountering less obstacles/dead ends. Ultimately the agent will find the appropriate routes/path after adjusting its actions based on the previous tries and reaching the treasure. Therefore, we can note that even though both approaches rely on experience, the machine agent will only use a structured reward/penalty learning system. Whereas we humans would try intuitively too, to be able to find the correct route, making the machine system a faster approach to solve the maze problem.

For this project two submission, the agent I worked on, was trained using a reward/penalty-based system, as mentioned above. The agent had the choice to move in four different directions (up, down, left, right) and then based on the penalties/reward values, which vary from -1 to 1 point, move accordingly. For example, if finding a cell in the maze where an obstacle was encountered, it would receive a penalty of -0.75 points, similarly, if attempting to exit the matrix structure of the maze, would then get a penalty of -0.8 points. Additionally, through multiple episodes, the agent would also adjust the winning rate until reaching 100% and minimizing the losses (penalties/obstacles encounters).

For the purpose of the intelligent agent for pathfinding, as mentioned above, via Q-learning through multiple trial-and-error, the agent is able to develop an efficient way to finally reach the end goal (treasure), regardless of a changing environment. This is key for AI applications like game dev. and robotics, where problem solving decisions must be able to adapt and be autonomous. This is where exploitation and exploration in reinforcement learning and the ideal balance between the two come in the picture. Exploitation being the use of the best-known strategy to maximize rewards, and exploration being the use of finding and trying new paths to look for a better solution/route. As previously mentioned, the ideal balance between the two and its implementation in the agent, the ideal proportion comes to 80% exploitation and 20% exploration, causing the agent to prioritize the most effective routes, while simultaneously trying new alternative routes to enhance the strategy. Therefore, we can see reinforcement learning aids the agent in determining the optimal route by rewarding the agent for finding the treasure and penalizing it when agent is hitting the obstacles, and through multiple tries, the agent learns the actions that lead to the rewards and best results, optimizing its navigation strategy.

For the implementation of Q-learning to the agent, as mentioned above, its reward/penalty-based system was optimal in helping the agent find the treasure. Thought its training process during multiple episodes, the agent was able to update its Q-values by processing state-action pairs to increasingly approximate the values. The agent also evaluated the losses and winnings throughout all the episodes tried, and using experience replay, which in this case was the win history, it refined its winning rate until eventually reaching the treasure with minimized penalties. Please see the code below and maze matrix image below as reference. Ultimately the agent through reinforcement learning, was able to reach the treasure after 20 episodes and a total of 170 wins (100% rate), at epoch 198 after 33.90 minutes.

if len(win\_history) > hsize:

win\_rate = sum(win\_history[-hsize:]) / hsize

A screenshot of a crossword puzzle

AI-generated content may be incorrect.

Sources

* *Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. Machine Learning, 8(3–4), 279–292.*
* *Humans vs. Machines: Untangling the Tasks AI Can (and Can’t) Handle*. (2023, November 22). Harvard Business School. https://www.library.hbs.edu/working-knowledge/humans-vs-machines-untangling-the-tasks-ai-can-and-cant-handle
* CS 370 Pirate Intelligent Agent Specifications SNHU <https://learn.snhu.edu/content/enforced/1798923-CS-370-19863.202511-1/course_documents/CS%20370%20Pirate%20Intelligent%20Agent%20Specifications.pdf?ou=1798923>
* Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. [Deep Learning](https://www.deeplearningbook.org/)