A Virtual Maze Game to Explain Reinforcement Learning

Youri Coppens[1,2][0000−0003−1124−0731], Eugenio Bargiacchi[1], and Ann Nowé[1]

[1] Vrije Universiteit Brussel, Brussels, Belgium
[2] Université Libre de Bruxelles, Brussels, Belgium
yocoppen@ai.vub.ac.be

Abstract. We demonstrate how Virtual Reality can explain the basic concepts of Reinforcement Learning through an interactive maze game. A player takes the role of an autonomous learning agent and must learn the shortest path to a hidden treasure through experience. This application visualises the learning process of Watkins' Q(λ), one of the fundamental algorithms in the field. A video can be found at https://youtu.be/sLJRiUBhQqM.

Keywords: Reinforcement Learning · Education · Virtual Reality

We present a Virtual Reality (VR) treasure hunt game, teaching the basic concepts behind Reinforcement Learning (RL) in an engaging way, without the necessity for mathematical formulas or hands-on programming sessions. RL tackles the problem of sequential decision-making within an environment, where an agent must act in order to maximise collected reward over time. Immersive VR allows us to put the playing user in the shoes of an RL agent, demonstrating through *direct experience* how new knowledge is acquired and processed. The user's perspective is aligned with the learning agent as much as possible to create a sense of presence in the RL environment through the head-mounted display.

The game puts the player in a foggy maze, with the task to find a hidden treasure. The fog restricts the player's vision to that of an RL agent, namely its current position (state) and available actions (Figure 1). The treasure allows the player to intuitively grasp the concept of reward in a standard RL process. The user can freely select actions and decide where to explore depending on the available information. All information collected via this exploration is fed to an RL algorithm, Q(λ) [2], which then displays the results of the learning back to the user via colours and numeric values.

The player's task is to find a treasure chest hidden in a grid-world maze (Figure 2). The maze additionally contains multiple empty chests to incentivise exploration. The player is paired with a Q(λ) learning agent which computes Q-values, values associated with each state-action pair that estimate the expected future reward resulting from executing a particular action in a particular state.

Fig. 1. Available actions and respective Q-values in a cell of the maze from the player's point of view.

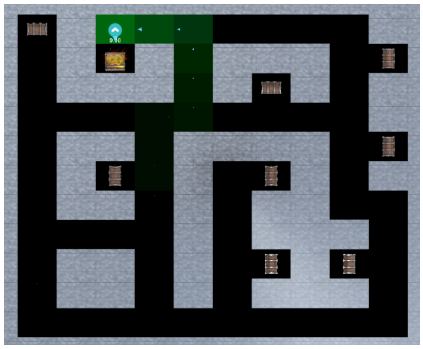

Fig. 2. Top view of the virtual maze. After finding the treasure, the agent receives a reward and updates the cell values, shown here in green shades.

As the player explores, Q(λ) updates the Q-values for each state-action pair and displays them on the ground. The highest Q-value for each state is visualised by shading the floor of each cell in green. Additionally, the eligibility traces of Q(λ), a mechanism that propagates knowledge on experienced reward over Q-values, are seen by the user as a trail of floating arrows. The player obtains a reward of 10 when the hidden treasure chest is found. Afterwards, the user is asked to repeat the task from another starting position, as Q(λ) requires episodic conditioning, under the guise of 'practice makes perfect'. After several trials, the maze will start to show a colour gradient towards the hidden treasure, which in turn helps the user to select the optimal direction to move. This allows the user to understand that reward is discounted over time. As the task is repeated, it will take less time for the player to enter a part of the maze which has already been visited before and thus contains updated Q-values.

Our demonstration has the potential to educate a broad audience on the dynamics of Reinforcement Learning [1]. A moderator directs the demonstration to ensure the game progresses smoothly and to keep the spectating audience involved. The moderator enhances the user experience by explaining the game's purpose and the mechanisms of Q(λ) on a level adapted to the present audience.

The demonstration was developed in C# using the Unity3D engine, the SteamVR plugin and the VRTK software framework. The user plays the game through a HTC Vive VR-system, on a computer containing a VR-ready GPU. The play field requires a minimal surface of 2 by 2 meters.

References

1. Coppens, Y., Bargiacchi, E., Nowé, A.: Reinforcement learning 101 with a virtual reality game. In: Proceedings of the 1st International Workshop on Education in Artificial Intelligence K-12 (August 2019)
2. Watkins, C.J.C.H.: Learning from Delayed Rewards. Ph.D. thesis, King's College, Cambridge, United Kingdom (May 1989)