Journal of Innovations in Computer Science and Engineering
Shahid Beheshti Unversity

January 2025, Special Issue on AI 4 All- 2

MQL-NPC: A Modified Q-Learning-based Approach to Design Intelligent Non-Player Character in a Survival Game

Morteza Nalbandi✉, Code ORCID: 0009-0008-5056-971X
Faculty of Computer Engineering K.N.Toosi University of Technology, Tehran, Iran mortazana8@gmail.com
Athena Abdi, Code ORCID: 0000-0002-5598-5762
Faculty of Computer Engineering K.N.Toosi University of Technology Tehran, Iran, a_abdi@kntu.ac.ir

Abstract—This paper presents an intelligent non-player char- acter(NPC) for a survival game with a modified Q-learning-based scheme. Due to the dynamics of the computer game environment, reinforcement learning is employed to make this agent smart. This leads the agent to react appropriately based on the game's scenario by choosing an action that provides a higher reward in the current situation. This is like a brain for the target NPC that processes different situations and reacts appropriately. Our intelligent agent is applied to a sample survival game with different complexity levels. In this game, multiple characters and objects alongside win-and-lose scenarios are considered. Our designed intelligent NPC is equipped with modified Q-learning to interact and try different actions on objects and learn about them. This learning process leads to an experience saved in the designed agent to react best to the environment. The efficiency of our proposed approach is evaluated through multiple scenarios and the appropriate reaction of the NPC is verified.

keywords: Non-Player Character(NPC), Intelligent Agent, Survival Game, Reinforcement Learning, Dynamic Environment.

I. Introduction

The gaming industry has been rapidly growing in recent years and games are improving in all aspects. The unpredictability of the game players and their intelligence, provided by artificial intelligence, challenges the users and makes the designed game more interesting. In recent game development projects, various aspects of non-person characters and their intelligence are focused. The intelligence of the enemies and NPCs, the dynamics of the game environment, and the appropriate reaction to the events and changes of the char- acters are some factors that make the game more enjoyable. These factors are mostly achieved by learning the agents to make the game live and unpredictable. This learning process makes the game and its characters more similar to real-world scenarios [11], [2], [9].

Artificial intelligence techniques can be employed in different aspects of computer games. One approach is adding a search system to stealth games. In this scenario, the NPC selects the action from predefined paths based on its last location. One of the best-implemented samples of this scheme is the "Last of Us-Part II" game. Another example has happened in shooting games like "Call of Duty" or "Battlefield" when the enemy NPC could not shoot directly at the player. In this case, they used other tools like grenades which would overpass barriers and explode to damage the target. This appropriate reaction based on the sequence of live and unpredictable events is one of the main requirements of the mod games. These reactions should also be real-time and robust in new scenarios. Adding the recognition capability to the game makes it capable of reconstructing appropriate actions in undefined scenarios, but it will take more time [11], [10], [8].

Interacting with objects is a sequential task and various actions' sets lead to different results. Selecting these actions based on the best local outcome does not lead to the optimal action. For instance, after the agent reaches a locked resource object in a game, small resources are received by collecting it. Hitting this object breaks the lock releasing more resources while double-hit destroys the resource thoroughly. Thus, various actions in a trajectory induce different effects on the agent and the environment based on the system's state. So, an appropriate interaction is planned step by step by looking over the result through the effects of performed actions. In this context, related studies focus on applying intelligence to the NPCs, game levels, and player experiences [1], [4], [5].

Our proposed method considers a reinforcement learning- based scheme to control the agent's interaction. In reinforcement learning, one of the main goals of an agent is to learn how to interact with its environment. To this aim, it interacts with the environment and chooses an action based on the learned policy and the environment's state. This interaction is one of the main challenges of this system due to its dynamic. Reinforcement Learning presents many advantages in sequential problems in which the agent needs to do some set of predefined actions the perform a specified task. The agent learns to interact with the environment better by receiving rewards based on the action it does in the environment. The agent interacts with the

Submit Date: 2024-12-01
Accept Date: 2025-05-05
✉ Corresponding author

environment multiple times to learn about different possible sets of actions and update its knowledge. After this training, the agent will be tested to see how well it performs the task in the environment. There are multiple versions of reinforcement learning algorithms designed for different scenarios like deep Q-learning, actor- critic, and so on.

In this context, the interaction is derived based on the saved knowledge and past actions. This action is evaluated by a learning policy guided by the environment and its repetition teaches the agent to have appropriate reactions. We employ a quasi-Q-learning scheme for the designed intelligent agent of a survival game. The learning process starts with design of a state-oriented system to interact with different objects and save the actions' results. This result is a dynamic reward system that is proposed to evaluate the appropriateness of the selected object considering the situation and time step. These steps are implemented through a modified Q-learning algorithm for each agent of the game. The main contributions of our proposed method could be summarized as follows:

- Designing a survival computer game with various objects and functionalities in the environment;
- Proposing a modified Q-learning-based scheme for the NPC to have appropriate reactions to objects during the game;
- Proposing a dynamic reward system based on the game's conditions and time step to well-adapt the reactions.

The rest of the paper is organized as follows: Section 2 presents the literature review, and the details of the proposed task scheduling approach are explained in Section 3. Section 4 represents the experimental results and the conclusion remarks and future trends are summarized in Section 5.

II. Related Work

In recent years, different aspects of artificial intelligence (AI) development in games have been studied in the gaming industry. Depending on the game genre, different approaches are employed to improve the intelligence of games. Sweetser et al briefly talked about AI in games at his time and gave numerous ideas on how to use the full power of AI in Games[3]. These improvements were focused mainly on three aspects of games: "Believable NPC", "Procedural Level Generation", and "Player Experience Modeling". Xia and colleagues did complete research on the matter and examples in game developments of AI[2].

A. Believable NPC

Reinforcement learning has been commonly used in the design of game NPCs in different games, different approaches were taken to use the power of RL in different game aspects. Arzate et al tried a new approach to use the power Of RL in designing believable agents in games [4]. in this method, it is tried to solve two main problems in NPC Behavior, huge state space and having behavior diversity. These problems need to be solved while keeping human-like behavior and being able to adapt to different play styles which directly affects game difficulty. Tests on Street Fighter IV show a 0.6 human- likeness ratio in the Turing test which is a good result. Zhao et al also proposed a method of training human-like agents, the goal was to perform a human-like performance in a team

sports game based on hierarchical learning by using both imitation learning and RL[5]. The proposed method achieved good results by performing human-like strategies and tactics.

B. Procedural Level Generation

Procedural Level Generation (PLG) means generating game levels randomly, which leads to a series of unpredictable possible game levels for different missions in games. Super Mario Bros. is one of the best benchmarks used in PLG. Guzdial et al. proposed combining two different inputs to create a new output for the PLG Problem. It outperformed three existing approaches which proves the power of the method.[6]. Angry Birds Games are another type of commonly seen game in the research of PLG. Their level-dependent design which every level combines different basic objects to create the levels makes this game a good benchmark [7].

C. player Experience Modeling

Player experience modeling aims to model how the player acts and plays during game-playing. The main motivation of AI games is to provide players with a more immersive and interesting game experience and unpredictable events. Hence having a general model of player behavior is very important. Luo et al. proposed a model player experience using gameplay videos and image processing techniques [8]. Their method is based on convolutional neural networks (CNN) and transfer learning and gathers information about the overall model of character behavior. Evaluation of a Super Mario Bros. style game showed their method outperformed several existing ones. Makantasis et al, also explored three different deep CNNs to learn the player's interest in a game through gameplay videos [9]. The proposed method achieved an accuracy of 75 percent.

III. Proposed Method

In this section, the details of our proposed learning scheme to find the agent's best action set for interacting with objects are described. First, the architecture of the problem in the form of a state machine for each specific object of the game is defined. Then the proposed reinforcement learning-based scheme and its components are presented. Last the proposed modified Q-learning and the designed reward are explained.

A. Architecture of the Game's Objects and Agents

As it has been explained, our target survival game consists of various objects and the appropriate interaction with them is the goal of the designed agent. Thus, to derive the object's state during the game we design a state machine for each object to inform the result of the action's result on it to the players. These state machines are exclusive to objects and the agent informs about the effectiveness of its performed action by the achieved reward. This reward has dynamic weights that are modified

based on the objects' state.
Our designed agents have the learning capability to have appropriate interactions. Due to the dynamics of the environ- ment and its step-wise manner, a modified Q-learning scheme

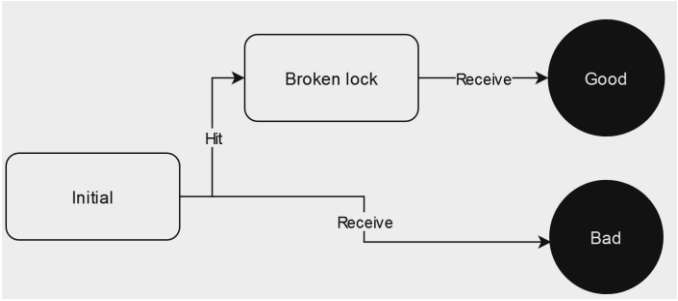

Fig. 1. State machine of "Power-up" object

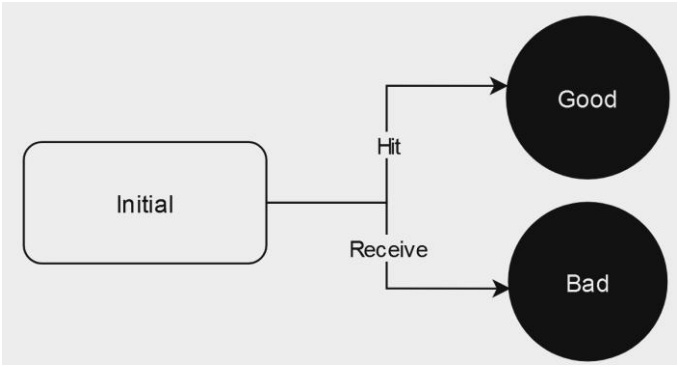

Fig. 2. State machine of "Block" object

is employed. This model-free approach assigns a value to the agent based on its action. Our proposed learning method has three important factors that differentiate it from basic Q- learning: considering different sizes for action sets of each object and lack of information about an Object's state.
In our designed platform, three objects are considered "Power-up", "Block", and "Iron". It is possible to extend these objects by considering their corresponding state machines. The "Power-up" object is available for the player and enemy NPCs to increase health. This object has a double effect with a lock embedded in it. By breaking this lock, this object improves the health of the agent that attains it twice but this object is destroyed. Thus the best trajectory of action for this object is hitting it once and collecting its reward while keeping it in the game for further use. The state machine of this object based on its valid actions and their correlated results is presented in Fig. 1.
The second object is the "Block" that prevents the enemies' advancement and slows them down. While facing this object, two actions are available: picking up or hitting it. Picking the block removes it from the NPC's way but slows it down. on the other hand, hitting the block decreases its durability and multiple hits lead to its break. Thus the best action when reaching this object is to hit it. The state machine of this object based on its valid actions and their correlated results is presented in Fig. 2.
The third object of the game is the "Iron" which is used to improve the state of the player and enemy NPC. Two actions are described for this object. Picking "Iron" adds it to the agent's storage and is a good action for both the player and NPC. While hitting the "Iron" decreases its durability and breaks it without any gain. Thus the best action when reaching this object is to pick it. The state machine of this object based on its valid actions and results is presented in Fig. 3.

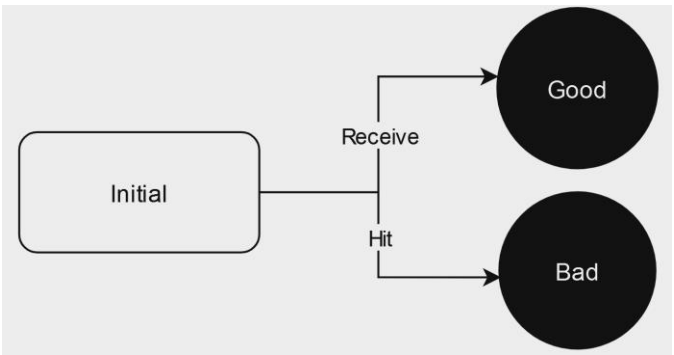

Fig. 3. State machine of "Iron" object

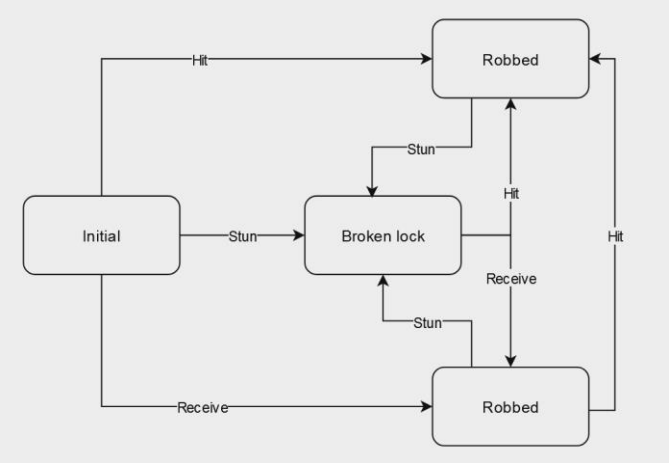

Fig. 4. State machine of ”Player”

The main character is modeled with a player unit in the game. The player has different abilities which it can use against the Enemy NPC. NPC can hit the player and damage it. NPC can also stun the player and Rob him from Irons. the best possible action set when colliding with the player is to Rob the player, stun him, and then deal the final blow by hitting the player. The state machine of the player based on its valid actions and results is presented in Fig. 4.

B. The Proposed Learning Process

As mentioned, the game’s agents have the learning capa- bility to interact appropriately with objects and achieve the best result. A modified version of Q-learning provides this learning ability in our proposed method. The value of the learning process is the corresponding reward of the action related to the state. In our proposed learning scheme, we utilize the state systems of different intractable objects to choose appropriate actions while facing various objects. This teaches each object’s desired action set independent of the overall state

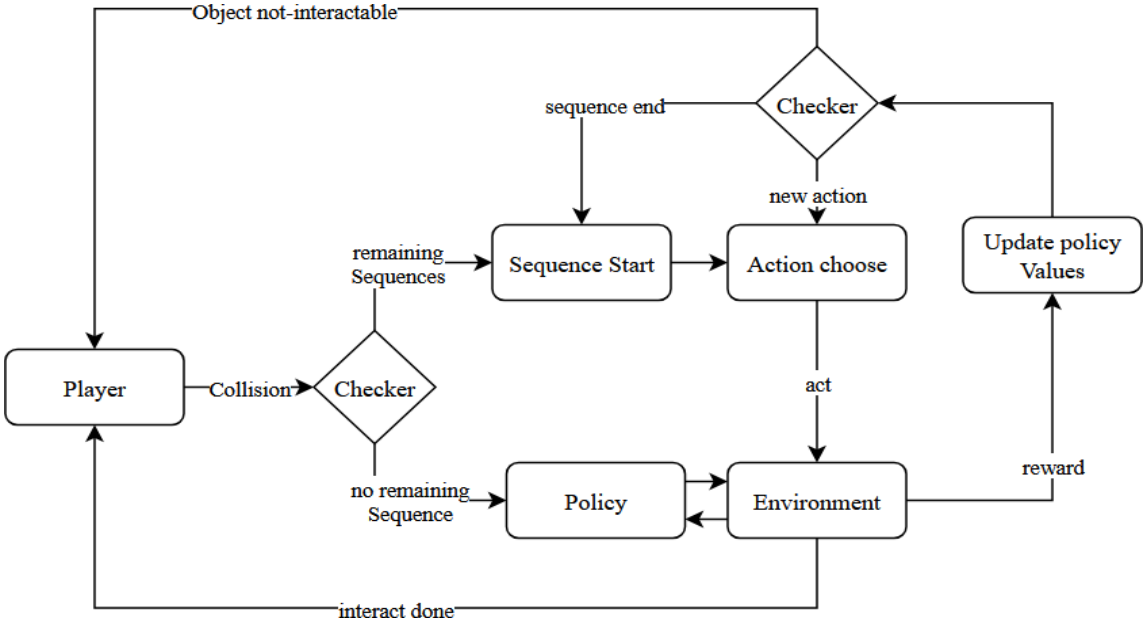

Fig. 5. Architecture of the Proposed Learning Method

of the environment to the agents. Moreover, in our proposed learning process instead of finding an action, we focus on the most effective sequence of actions to get the best result. To this aim, rewards of a set of actions from start to finish are reviewed and summed to get a numerical value to evaluate the sequence. Afterward, the action set with the most cumulative reward is selected.

The mentioned learning process is split into multiple phases to have distributed learning, memory management, and server-client connection costs in a master manager platform. The architecture of the proposed learning process of the agents is presented in Fig. 5. As this figure shows, when the target agent faces an object it checks the database to see whether the best sequence exists. If not it checks possible action sequences on the object to find the best and most rewarding action sequence. The reward of each action is set based on its appropriateness in the current state. Then at the end of the sequence, the sum of the rewards is employed to evaluate it compared to the others. These values are stored in a memory table to be used during the future decision-making process. The training process terminated when all possible sets had been checked on a

state. In this case, the best result for the action when facing the desired Object is saved.
During the learning process, the effects of the actions are computed through a reward. Our designed reward is dynamic

and depends on the state of the objects. It means each action on an object does not lead to a unique reward. The rewards are defined based on the object's attributes that shape its behavior. As well the rewards are set to be positive or negative to determine the importance of the path and not the instant step. For instance, if the NPC hits a "Power-up", it receives a -5 reward. But if it picks up the reward, a 10 reward is achieved. However, after hitting a "Power-up", picking it up leads to 20 rewards due to its more efficiency. The design of the algorithm is optimized to cope with different sequence sizes and actions possible, also when the system is introduced to new mechanics and actions ,the system can easily adapt to the new setting by using the knowledge it has obtained in the past interactions. at the beginning of the learning phase, we focus on trying all possible sequences to update the policy. after reviewing all possible sequences, we only choose the sequence with the most estimated reward. because the main agent is a NPC which we can have multiple of in a game, we can use the brute method to speed up the learning process in more complex and hard environments. as explained, at first we focus in exploring all possible paths and after learning is complete, we only focus on exploiting the learned rewards.

Fig. 6. Environment of the considered game and its agents and objects items

IV. Experimental Results

To implement our proposed method and the gaming plat- form, we used Unity Game Engine and C# to add extra capabilities to the game. The output of this environment is a playable game that can be played on any system. Fig 6 shows the environment of our considered game and its agents and objects items. In this figure, the log of the learning process and the outcomes of the steps are recorded along with the game advancement which will be shown in the left part of the screen. the ability and weapon of the player are also shown in the HUD of the game which can be viewed in the top part. The enemy NPC chases the player until it reaches it and after completing the learning steps, chooses the best set of actions to perform on the player. for the parameter setting, we chose a limit of at most 3 actions for each sequence. we also have 3 different action for each agent NPC.
To evaluate the learning process and the effectiveness of the selected actions, two scenarios are considered. First, the accuracy of the proposed learning method in a baseline scenario where the NPC has to overcome a bunch of "Blocks" to reach the player and defeat it is examined. In the second scenario, we put different objects between the player and the NPC. In this case, the method's ability to switch between learning different scenarios without affecting the overall learning process is verified.
1) Evaluating the proposed Learning Method: In this ex- periment, the NPC starts to head toward the player, after colliding with an object. In this scenario, the NPC checks whether it has found the best solution to the action set by reviewing all possible scenarios. In case of lacking the best solution, it starts to test the remaining action sets. If the object is destroyed, it continues its path until reaching another object and continues the same until finding the best action set. After finding the best action set, when colliding with an object, it chooses the action set. the overall learning phase can be seen in Fig. 8. As we see it gets different rewards for each set which indicates the path's goodness. the same scenario was tested on the player when colliding with an NPC. As we can see in Fig 9 the NPC tried multiple possible rewards and in the end chose the path with the best reward overall. In Fig 7

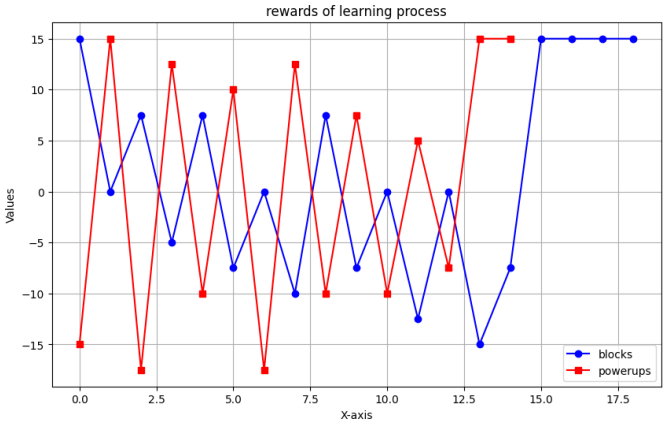

Fig. 7. Learning process rewards

Fig. 8. Learning process of the target NPC colliding with Block object in the game

the curve of rewards in the learning process is demonstrated. As this figure shows, the agent tries different possible sequences of action and get different rewards to try and learn the best possible action. The NPC learns the appropriate action for its confronting object and after that, kills. To this aim, various action sequences on the ahead object are applied and their corresponding rewards are computed. The best reward represents the best solution and is saved for further cases.

Fig. 9. Learning process of the target NPC colliding with Player in the game

Fig. 10. Learning capability of the target NPC colliding with various objects in the game

2) Evaluating of the proposed Learning Method in complex Scanrio: The previous experiment focuses on a single type of object. The ability of the proposed NPC to react to different objects simultaneously is also important. In this case, some ”Power-ups” objects are put in the path of NPC. Before NPC completely learns the ”Power-up” action-set, we collide it with some ”Irons” and more ”Power-up”s. As presented in Fig. 10, the proposed NPC starts to learn ”Iron” after colliding with and exactly after colliding with ”Power-up”, it continues to learn from the last checkpoint. This shows the proposed learning process is capable of handling different put-on situations and generating appropriate sequences.

V. Conclusion

In this paper, a learning-based approach to make the NPC of a survival game intelligent is proposed. This intelligence leads to making the games more real and unpredictable. To this aim, our proposed method uses reinforcement learning to deal with the dynamics of the environment. Our method is based on modifying the Q-learning and applying it to the target NPC of the game. The intelligence of NPC is defined in its appropriate reaction while confronting the various objects of the games and the players. The proposed modified Q-learning scheme is lightweight and capable of evaluating various scenarios and scoring them to determine the best action set based on the NPC’s experience. another important capability of the method is to cope with scenarios where the agent’s knowledge about the state of the environment and objects is limited to reward. in the said scenario the agent can learn the best sequence of action to be used in the environment The efficiency of the proposed method in a sample survival game is studied and the results verify the efficiency of the learning process. As a future trend, implementing a master-worker system that uses a centralized manager to update the workers based on the new knowledge of different workers is proposed. Moreover, improving the method for employing more complex games is suggested as another trend. NPC systems are not always simple and may need complex control and action selection to acheive the best result, the proposed method can work as a base model to improve upon and address such challenges.

References

[1] K. Arulkumaran, M. Deisenroth, M. Brundage, A. Bharath, ”Deep rein- forcement learning: A brief survey,” IEEE Signal Processing Magazine, 34(6), 26-38, 2017.

[2] Xia, Boming, Xiaozhen Ye, and Adnan OM Abuassba. ”Recent research on ai in games.” 2020 International Wireless Communications and Mobile Computing (IWCMC) (2020): 505-510.

[3] Sweetser, Penelope, and Janet Wiles. ”Current AI in games: A review.” Australian Journal of Intelligent Information Processing Systems 8.1 (2002): 24-42.

[4] C. Arzate Cruz and J. A. Ramirez Uresti, “Hrlb2: A reinforcement learning based framework for believable bots,” Applied Sciences, vol. 8, no. 12, p. 2453, 2018.

[5] Y. Zhao, I. Borovikov, J. Rupert, C. Somers, and A. Beirami, “On multi-agent learning in team sports games,” arXiv preprint arXiv:1906.10124,2019.

[6] M. J. Guzdial and M. O. Riedl, “Combinatorial creativity for procedural content generation via machine learning,” in Workshops at the Thirty- Second AAAI Conference on Artificial Intelligence, 2018.

[7] “Aibirds cog 2019 level generation competition,” https://aibirds.org/level-generation-competition.html, accessed January 30, 2020.

[8] Z. Luo, M. Guzdial, N. Liao, and M. Riedl, “Player experience ex- traction from gameplay video,” in Fourteenth Artificial Intelligence and Interactive Digital Entertainment Conference, 2018.

[9] K. Makantasis, A. Liapis, and G. N. Yannakakis, “From pixels to affect A study on games and player experience,” in 2019 8th International Con- ference on Affective Computing and Intelligent Interaction (ACII).IEEE, 2019, pp. 1–7.

[10] Z. Zhan, Y. Tong, X. Lan, B. Zhong, ”A systematic literature review of game-based learning in Artificial Intelligence education,” Interactive Learning Environments, vol. 32, pp. 1137-1158, 2024.

[11] L. Hammond, J. Fox, T. Everitt, R. Carey, A. Abate, M. Wooldridge, ”Reasoning about causality in games. Artificial Intelligence,” vol. 320, 2023.