Journal of Computing and Communication Vol.2 , No.2 , PP. 38-51 , 2023

VR Computer Role-Playing Game Using Reinforcement Learning

Bilal Ghareeb*[a], Mohamed Doqmak [a], shihab Mostafa Kandil [a] , Farah Sherif [a] , Verina Adel, walaa hassan [a]

[b]Department of Computer Science, Misr International University, Cairo, Egypt

*Corresponding Author: Bilal Ghareeb [***bilal1908384@miuegypt.edu.eg***]

ARTICLE DATA

Article history:
Received 18 June 2023
Revised 05 July 2023
Accepted 05 July 2023
Available online

Keywords:
Virtual Reality
Hand tracking
Reinforcement learning
Deep learning
Video game

ABSTRACT

This paper shall discuss the process of making one of the first Computer Role-Playing Virtual Reality Games that take advantage of the underutilized hand-tracking technology provided by the Oculus Quest 2 headset. The game takes place in two separate fictional dystopian worlds. The player will have to navigate and explore to discover its secrets while trying to survive against enemies that employ a reinforcement learning algorithm to stop the player. Through extensive experimentation and fine-tuning of various implementation aspects, including positive and negative rewards, hyper parameter values related to Proximal Policy Optimization (PPO), and the number of environment observations, our agent achieved highly favorable outcomes in terms of general machine learning model efficiency indicators. As explained in the methodology section, the model's behavior aligns with the desired behavior, achieved by implementing three rays at different angles to determine some of the rewards. These results validate the effectiveness and success of our approach in training the agent to exhibit desired behaviors and achieve desirable outcomes.

1. Introduction

Virtual reality is the future and the next step in technological advancement [1]; it can be used for educational purposes, tourism, or entertainment. Role-playing games (RPGs) are one of the most loved genres with countless titles, but developers have rarely made role-playing games for virtual reality headsets. Moreover, even though a large amount of the VR applications out there are games, only a very small number utilize hand tracking technology even though it has large potential in addition to providing the player with a more immersive experience by removing the need for a controller to interact with objects or move around and navigate through the environment. Other papers have been written on the potential of virtual reality for different purposes as well as how to improve the experience of users, such as the paper by Jonatan Hvass, Oliver Larsen, Kasper Vendelbo, and their colleagues [2], where they explored whether the quality of the graphics and textures affected the player's overall experience. To do so, they recruited 50 participants, ages ranging from 16 years to 55 years old, both female and male. The participants played a horror VR game where they were tasked with escaping from an abandoned apartment by searching for and getting four keys hidden around the apartment to be able to unlock the front door, while the second door of the apartment is rattling and scratched by unknown menace, as well as the players being subjected to random scares during exploring to add to the level of stress. The researchers made two versions of the game, one with hyperrealistic graphics and one with low-quality graphics, then gathered information on the participants' experience using Slater-Usoh-Steed (SUS). Self-assessment Manikin (SAM) questionnaires and some physiological measures such as the blood volume pulse showed that the more realistic and higher the general quality of the environment is, the more immersed the players felt.

This paper discusses a proposed VR game that uses the hand tracking integrated within the Oculus Quest 2 headset, the player shall be free to do multiple things, such as choosing a character class, exploring the environment, solving puzzles by interacting with objects and teleporting to a second dimension to gather clues, and battling enemies that use reinforcement learning as an added challenge. First, similar papers shall be mentioned to shed more light on virtual reality and reinforcement learning in games in section 2. The gameplay and the mechanics of the proposed game shall be explained in further detail in section 3; afterward, the results and progress made so far, as well as what we plan to add in the future, shall be in section 4, and lastly, the paper shall wrap up with the conclusion which is section 5.

2. Related Work

Many attempts to discuss VR Computer Role-Playing Games using Reinforcement Learning can be classified into different categories as follows:

2.1. Virtual reality games for rehabilitation of people with stroke: perspectives from the users[3] :

The paper discusses the potential of using virtual reality games to rehabilitate stroke patients, and they aimed to design a VR game that improves upper limb movement and measure how much the movement improves and the patient's experience during the game. Six participants suffered from chronic hemiparesis; their forearms and wrists were in a fiberglass cast that rested on a 6-degree-of-freedom load cell that controlled the submarine in the game environment. The game consisted of eight games, each with a higher complexity than the one before, ranging from movement on a single axis to movement on multiple axes. The study conducted 1-hour sessions for each participant per week for six weeks, and the arm function was measured in the first and last sessions to see the improvement. Their rate of progression measured the practicality of the games through the games. Their opinions on this recovery method were gathered using a questionnaire and interviews with a researcher in the last two weeks. Only three out of the six managed to complete all the games even though all of them attended every session, while the other three couldn't complete the more complex games; however, they managed to get through a good level of difficulty. All participants showed improvement after the sessions; nonetheless, the results weren't medically significant. Despite that, they seemed to enjoy the rehabilitation experience through games and reported improvement in their arms, having had an overall positive experience. They seemed to encourage the idea of using VR games as a means of recovery as it provided them with a fun challenge to complete and improve their health [3].

2.2. Evaluating Performance and Gameplay of Virtual Reality Sickness Techniques in a First-Person Shooter Game [4] :

In virtual reality, one of the main and most important aspects is conveying the feeling of being in a different world or environment; however, many people experience Virtual Reality Sickness (VRS), especially in games or fast-paced environments. Even though it's a widespread issue that has plagued many VR users, the issue of VRS hasn't been given a foolproof solution yet, even with the advances in VR technology made in recent years. One way to mitigate the VRS is for VR applications or games to use teleportation as a way of movement instead of continuous movement. Another solution was to decrease the Field-of-View (FoV) of the user or to take a 3D environment and display it as a 2D Frame (2DF) to decrease the amount of sickness experienced, but as the teleport method hasn't been compared to the 2DF method, the paper aimed to discover that. They performed an experiment in which they implemented the 2DF technique using Unity and the teleport method in a normal FPS game. The players were required to traverse a maze and defeat enemies without being hit by the energy balls thrown at them by the said enemies. To test whether the methods decrease VRS, they designed the game to force the players to rotate, which is one factor that induces VRS. The researchers used the Simulator Sickness Questionnaire (SSQ) and Immersive Experience Questionnaire (IEQ), which measured the player's level of Nausea, Disorientation, and

Oculomotor. They had eighteen participants play each version of the game and then fill out a questionnaire about how sick they felt. The results showed that the 2D experience was better in terms of Nausea and Oculomotor, but it had the highest scores when it came to disorientation; second to 2D was the teleport method, it didn't have bad scores overall, yet it seemed to prove a bit hard to use in an FPS environment which required quick and frequent movement, in last place was the 3D game with no VRS mitigation techniques. Their results suggested that 2DF had the most potential for decreasing the effects of VRS over the 3D environment and the teleport method [4].

2.3. Deep Reinforcement Learning in Immersive Virtual Reality Exergame for Agent Movement Guidance [5] :

Exercise game applications of immersive VR can direct and inspire people to engage in physical activity. Machine learning developments offer the potential for more intelligent gameplay in these types of games. This study examined how to teach a virtual robot arm to direct, challenge, and test actions during physical activities. They showed a unique artificial intelligence-powered game mechanism that uses the Unity Game Engine, Unity Ml-Agents, and the HTC Vive Head-Mounted Display to aid users visually. Deep reinforcement learning and generative adversarial imitation learning are utilized to finish the virtual reality game workouts. The findings imply that deep learning agents efficiently pick up on game mechanics and offer special insights to players. Two training sessions were held. Parallel agent training utilizing solely Proximal Policy Optimisation (PPO) was the first session's focus. PPO and Generative Adversarial Imitation Learning (GAIL) were integrated into the second session. While the GAIL + PPO model investigated the influence of user demonstrations and personalized agents depending on user movement preferences, the PPO-only model sought to maximize incentives. Each butterfly exercise movement was demonstrated by humans, who then recorded the performances. Elbow and shoulder joint movement dynamics were recorded with Vive Trackers. A total of sixteen parallel agents were used for training. For the specific arrangement of each trainer, model parameters were adjusted. To learn particular workout movements, each training model completed one million steps. During the one million training steps with 16 parallel agents, the "PPO Only" and "GAIL + PPO" models showed encouraging learning rates. The fact that the "PPO Only" model received a greater reward shows how well it teaches workout movements. Despite obtaining lesser incentives, the "GAIL + PPO" model showed the potential for identifying user movement biases and flaws. Customizing agents could change exercise difficulty based on user demonstrations. Additional testing with a broader user sample is required. The "PPO Only" model in Project IB showed a notable improvement in cumulative reward compared to the Reacher Agent. The trained double-jointed arm delivered Visual cues for immersive virtual reality (iVR) exercises. A pilot study compared the "PPO Only" model to human agents. Four individuals in the pilot user study beat the Project IB agent successfully. In the horizontal shoulder rotation movements, the agent was on par with or slightly superior to the users. However, users outperformed the agent in the Forward Arm Raise and Side Arm Raise activities, with mixed results seen in the Side Arm Raise. Users said they got exhausted while doing the exercises, emphasizing the value of deliberate, slow motion. Although Project IB's initial results were encouraging, some drawbacks still exist. Additional users must be included to assess the efficacy of the "PPO Only" and "PPO + GAIL" models and investigate unlearned exercises.[5]

2.4. Reinforcement learning 101 with a virtual reality game[6] :

In their paper, Youri Coppens, Eugenio Bargiacchi, and Ann Nowe demonstrated how virtual reality might instruct students in the core ideas of reinforcement learning. They depict the Watkins' Q() learning process through an interactive treasure hunt game, a crucial algorithm in this area. Players take on the role of an autonomous entity in the game, learning the quickest way to a hidden prize via trial and error. The application also provides an external display so spectators can watch the game. They used Watkins' Q()

algorithm because of its simplicity and crucial place in modern RL. For each state-action pair, this method keeps Q-values and eligibility traces. Temporal difference (TD) mistakes weighted by eligibility traces and a learning rate are used to update the Q-values. The eligibility traces give credit for prior interactions while the learning rate permits incremental updates, hastening the learning process. Based on a parameter, the eligibility traces degrade rapidly as interactions move apart. So they may efficiently learn the value function in a tabular manner and enable better approximations in stochastic contexts by using Watkins' Q() approach. To introduce reinforcement learning (RL) ideas, they created a virtual reality (VR) treasure hunt game. The game aims to uncover a hidden treasure while navigating a foggy maze. Instead of conducting their research, they rely on the information provided by an autonomous agent. A Q() learning agent that updates Q-values for state-action pairings is present in the game. Players can change real-time algorithm parameters. Finding treasure chests yields benefits, with various chests yielding various rewards. On an external monitor, onlookers can see the player's point of view and understand the difficulties RL agents face. Their VR game provides an engaging and instructive experience to comprehend RL principles. The concepts and dynamics of tabular Reinforcement Learning (RL) are illustrated in their virtual reality (VR) game. They intended to demonstrate how RL may address various issues by casting the player as a learning agent with limited information. Despite favorable comments from public demos, they have not yet done a formal review of the efficacy of their tool; this is something they want to do in the future. While the typical grid-world problem served as the inspiration for their labyrinth environment, different goal-based scenarios might be investigated to show off the adaptability of RL. They can also use more core RL algorithms like SARSA or REINFORCE to emphasize various strategies. Additionally, the emergence of multiplayer VR games could broaden our example to incorporate multi-agent issues [6]. Also, other related work in [7-26] has been proposed in recent years to address machine learning and its application in different fields.

3. Proposed Approach

This section demonstrates the proposed VR Computer Role-Playing Game method using Reinforcement Learning. The following are the specifics for each step.

3.1. System Overview

Figure 1 shows the System overview. The game begins with the player being introduced to hand tracking, as seen in the figure. A tutorial will begin that instructs the player to select a character, melee, and ranged class, where each has a different play style and stats. The player will then be taught the movement mechanic to be able to explore the environment, and the teleportation mechanic that allows the player to switch to the second dimension and back, following that the use of the inventory system will be explained after its completion takes the player to a simple combat encounter with an enemy that utilizes reinforcement learning to teach him about battling enemies, after the enemy's defeat the player will gain experience points which the player can then use to upgrade his stats such as strength and defense, with that the tutorial then ends. The player is free to wander around the world, solve quests, defeat enemies and switch dimensions to uncover the story of the world.

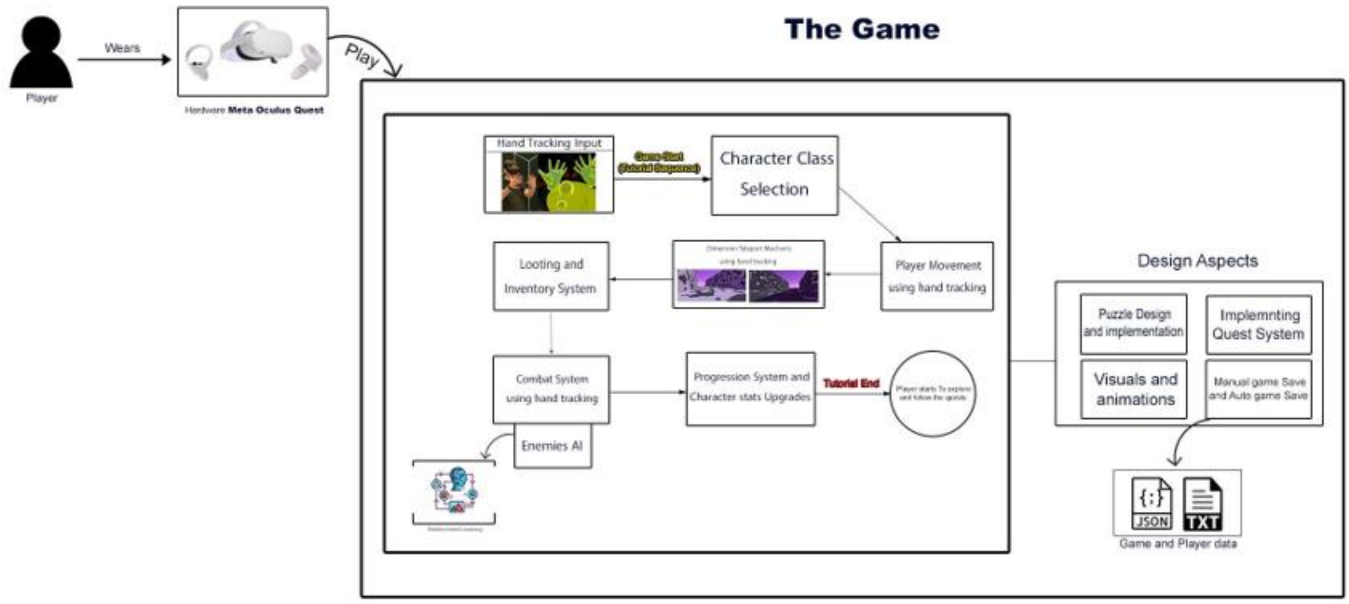

FIGURE 1. System Overview

3.2. Game Mechanics

This study utilized the usage of Oculus Interaction SDK to enable hand interactions and heuristics on Meta Quest headsets, as well as the Oculus XR plugin to provide head tracking, hand tracking, spatial audio, and passthrough to improve the user experience in a VR game. The algorithm used to enable hand gestures in the game involves measuring the distance between the player's hands using the OVRSkeleton entity and comparing the resulting distances to a predefined list of gestures. The gesture corresponding to the smallest distance is then used to execute a specific operation.

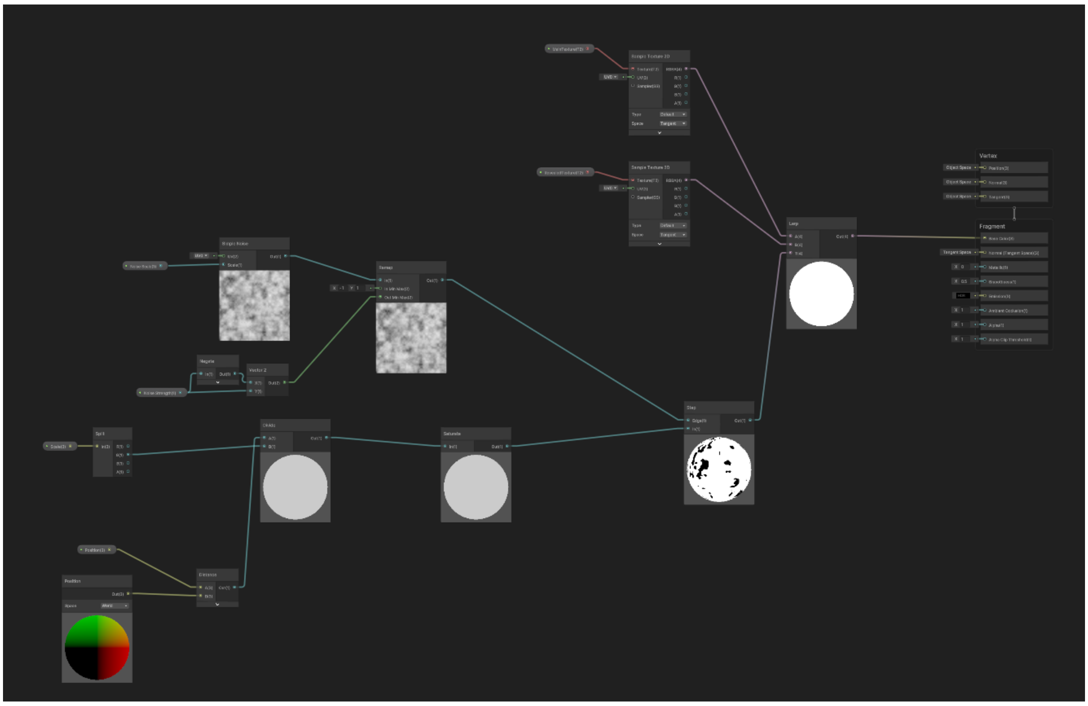

FIGURE 2. Blending Textures Shader Graph

In the VR game, the player can teleport between dimensions within the current scene using a core mechanic called dimension teleportation. This allows the player to explore different aspects of the game world and discover new paths and opportunities for exploration. When the game recognizes the appropriate gesture, it executes the necessary code to initiate the dimension teleportation, allowing the player to transition between different dimensions seamlessly. Using hand gestures as the primary input method for the dimension teleportation mechanic adds a layer of immersion and interactivity to the game, allowing players to feel like they are truly interacting with and manipulating the virtual environment. When developing the dimension teleportation mechanic for the VR game, the team initially considered replacing the models at the transition point between the two worlds. However, due to the hardware limitations of the Oculus device, they had to use a different strategy. The team opted to replace a smaller number of models that distinguish each world, leaving the remaining ones intact. They implemented this using a shader graph, as shown in Figure 2, that blends between two different textures to combine the landscape and environment and change its texture. This approach was found to be appropriate for the capabilities of the Oculus device.

When the player wears an Oculus VR headset and uses their hands to interact with a virtual environment, the OVRSkeleton SDK is used to track the movements of their hands and render them in the virtual environment. However, the team chose to have their custom hand models represent the player's hands in the virtual environment. In that case, they mapped the movements of the custom hands to the movements of the OVRSkeleton hands, allowing the player's gestures and movements to be accurately represented in the virtual environment. They choose to have a full-body model to allow more natural and fluid movement for the player in the game. They had to prepare the body model with inverse kinematics (IK) animations. IK animations is a technique used in 3D animation to control the movements of a character's limbs by manipulating their bones rather than controlling each joint. Using the Animation Rigging package in Unity, the team could define the range of motion for each bone and prepare it for IK animations, as shown in Figure 3. This allows for a high degree of control over the model's movements and interactions with the virtual environment depending on the player's movements and interactions.

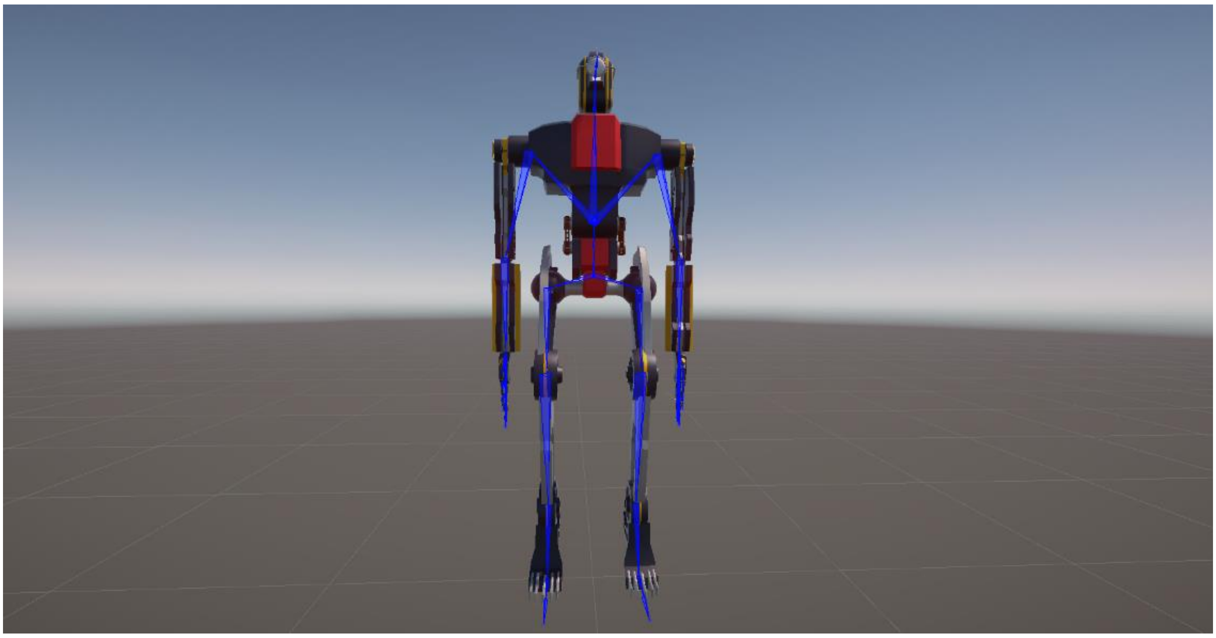

FIGURE 3. Custom Model

The player can move and explore by teleportation using hand tracking, where the player uses a certain hand gesture to trigger the teleport. The player's inventory system consists of a belt from which the player can place or remove objects and use them to switch weapons. When fighting the enemies, the player may get killed, and if he does, he will re-spawn at a safe location; on the other hand, if the player wins, he will gain experience points that can be used at certain locations to upgrade his stats. To progress in the game, the player must finish quests involving solving puzzles or switching dimensions.

3.3. Reinforcement Learning

The enemies use deep reinforcement learning, as mentioned previously, and they specifically use Proximal Policy Optimization [7]; there is currently one type of enemy: a melee type that uses a baton to attack. The enemy training process had several steps, which shall be discussed in more detail.

The first step was creating an MDP (Markov Decision Process) to visualize how the enemies should behave, and it was used to showcase the possible actions the enemies might take in a given state. The state of the environment that the enemy agent will observe determines the agent's actions and the consequent reward it may receive for that action. The states are represented as circles, actions are the arrows, and rewards are written above the arrows, as seen in Figure 4.

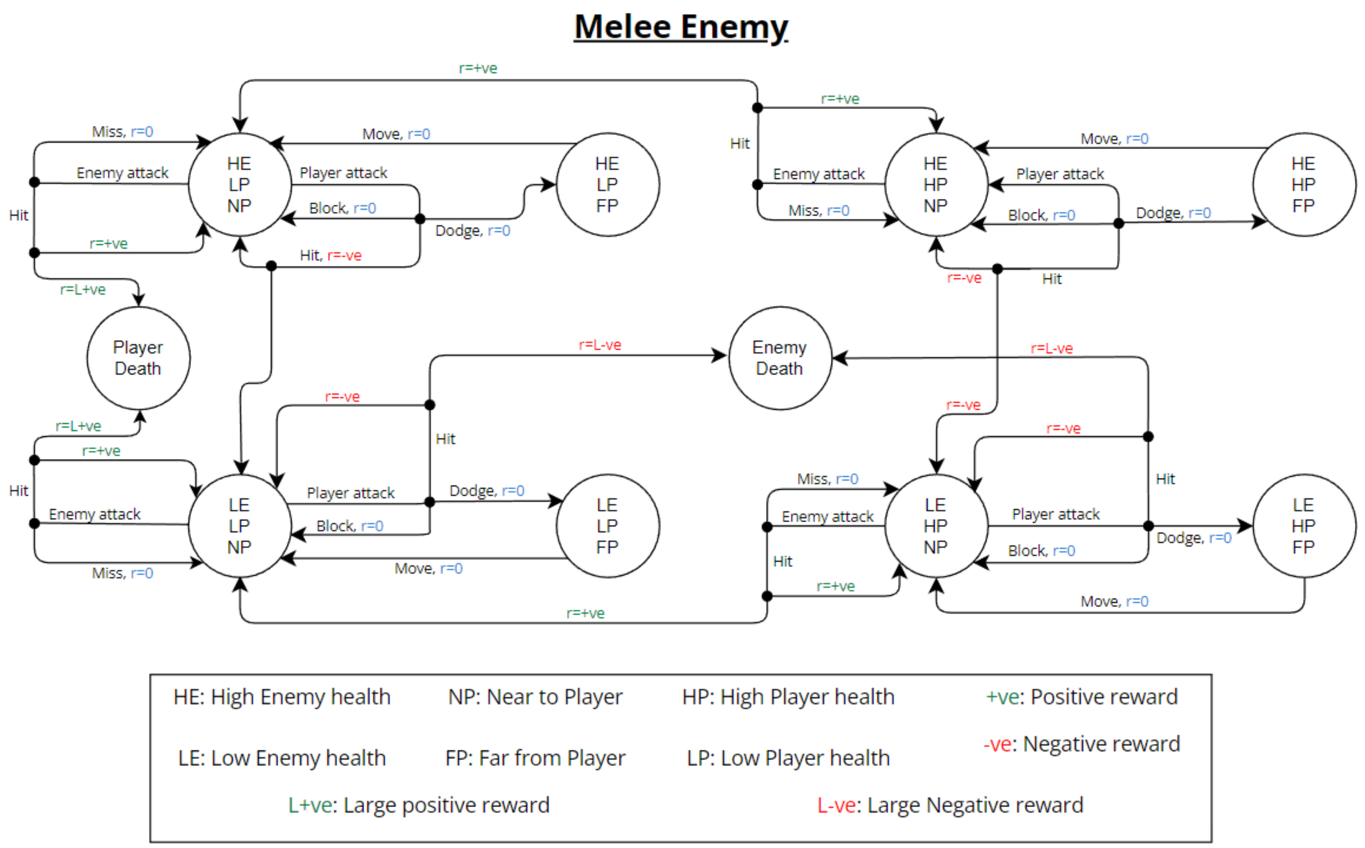

FIGURE 4. Melee Enemy MDP

The agents were created using the MLAgents package from Unity, which allows the development and integration of different agents in Unity games; thus, the second step was learning how to use the package and create a complex agent with it. To that end, a simple agent was made then the behavior's complexity was incrementally increased until the desired behavior was reached. The initial agent's main goal was chasing a target which spawned in a different position each episode, as in Figure 5, while the second agent had the added challenge of avoiding obstacles, as seen in Figure 6; for all the agents, the environment was duplicated 9 times to speed up training

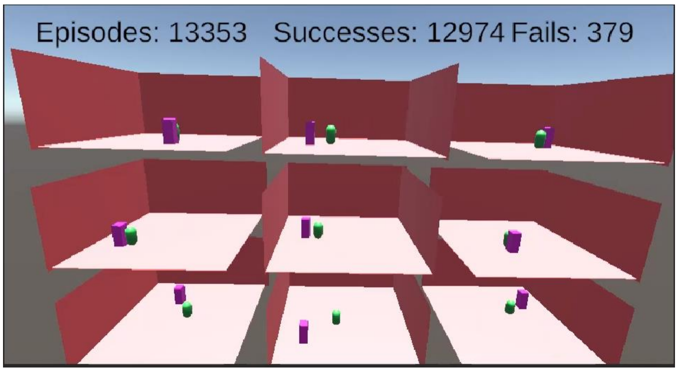

FIGURE 5. Chaser Agent

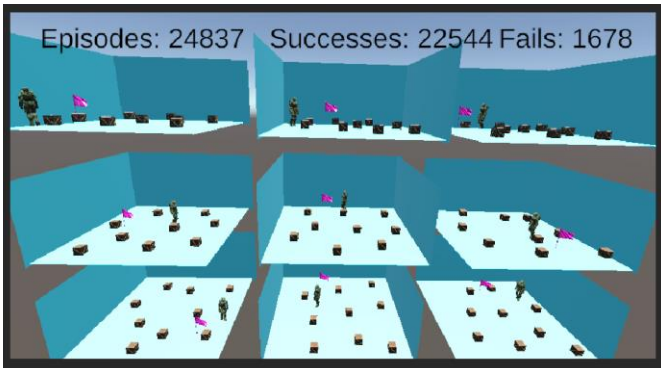

FIGURE 6. Obstacle Agent

The final iteration of the agent was trained using self-play [8], where it fought against an older version of itself to teach it how to behave while playing against the player later on. It was trained in an environment similar to where it would be placed in the actual game with random obstacles around the environment. The agent was fitted with three short rays emanating from the center of the agent; one was in front of the agent, while the other two were at a 30-degree angle to both sides. They were used to "see" its environment, as they detected whether the agent was close to an obstacle or the player. The agent could take a few select actions at any point: attack, block, walk forward, idle, rotate left, rotate right, or no rotation. Each action could result in a positive or negative reward, and the main rewards are represented in Table 1; as observed, all the rewards are in the range of -1 to 1 since machine learning models train the best in that range. The agent determines which action to take by observing the environment as mentioned previously; specifically, the agent observes its position, the player's position, what each ray is detecting, its rotation around the Y-axis, its own health points, and the player's health points.

TABLE 1: Hyper-parameters for Rewards

Rewards	Reason
-0.0002	A ray detects an obstacle
0.0002	The middle ray detects the player
-0.5	Hitting a wall
1	Decreasing player HP to zero
-1	Its own HP decreased to zero
(old distance - new distance) / 8	Getting closer/further from the player
-0.5	Taking too long to win

The final hyper-parameters used to train the agent are displayed in Tables 2 and 3, where Table 2 shows the main general hyper-parameters, while Table 3 displays the self-play-specific parameters. These values have been adjusted according to the graphs of each training run and the agent's behavior; thus, these values offered the best results after many tries.

TABLE 2: The main general hyper-parameters used to train the agent

Parameter	Value
Batch size	320
Buffer size	20480
Learning rate	0.0005

Beta	0.0005
Epsilon	0.2
Lambda	0.94
Num Epoch	5

TABLE 3 : The self-play specific parameters

Parameter	**Value**
Window	10
Save steps	25000
Team change	125000
Swap steps	25000

4. Experimental Results

4.1. Game Mechanics

The game's two main aspects are the RPG elements integrated with the hand-tracking technology and the reinforcement learning agent, which will be discussed in more detail.

The game starts with the main menu screen, as shown in Figure 7, which allows the player to start the game by picking a character class; currently, there are two classes; the first is a warrior class, figure 8, which uses a melee weapon while the second is a ranged class that uses a bow.

FIGURE 7. Main Menu

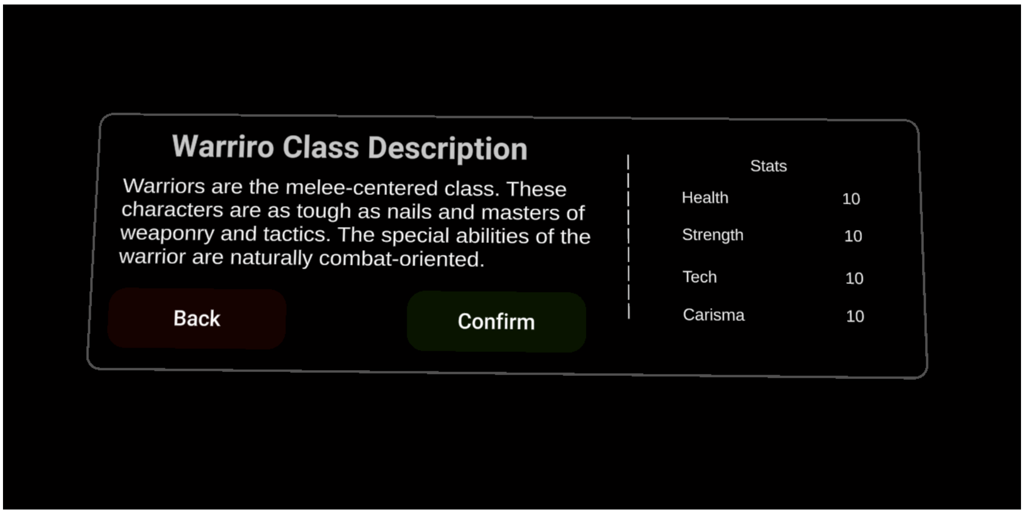

FIGURE 8. Warrior Class Description

After picking the class, the player can start the game; in the beginning, there is a small tutorial that instructs the player on how to move around using the teleportation triggered by the button on his hand, figure 9. The player can begin interacting with the world.

FIGURE 9. Movement Button Trigger

There are multiple rooms in the hallway, one of which contains the enemy that uses reinforcement learning, which shall be discussed later, and a room with a puzzle. The puzzle requires the player to find a code to open a chest containing a fuse required to continue. During the player's exploration, he may come across a friendly NPC, which shall guide him through the game, shown in Figure 10.

FIGURE 10: E-99 : The first NPC, the player, will encounter

4.2. Reinforcement Learning

In the final analysis and by experimenting with different implementations for our agent, testing with different both positive and negative rewards, tweaking the values of the hyperparameters that are related to the PPO (Proximal Policy Optimization), and also changing the number of environments observations which are represented as the "space size" and accordingly changing the code implementation for the agent to match these changes, It was determined that the agent achieved the most favorable outcomes in terms of general machine learning model efficiency indicators, and also that the model behavior is congruent with the desired behavior for the model by implementing three rays at three different angles that are cast out of the agent which is used to determine some of the rewards as discussed in the methodology section. Some of the machine learning model efficiency indicators are Cumulative Reward (figure 11), Episode Length (figure 12), and Entropy (figure 13).

In all of these figures, the x-axis is the same, representing the total number of steps where each step is a unity frame update.

In Figure 11, the y-axis represents the Cumulative Positive Reward, which refers to the total number of rewarding experiences an agent has across all episodes or encounters with an environment. Positive reinforcement and encouragement are frequently given to reinforce or promote desired behaviors and actions that result in positive results or accomplishing goals. The agent acts in the environment while learning and gets input in the form of rewards. Whether the agent's behaviors align with the desired aims will determine whether these rewards are positive, negative, or neutral. The agent's positive rewards from all events or encounters are added up to determine the cumulative positive. The cumulative positive reward of this figure peaks at around 0.7.

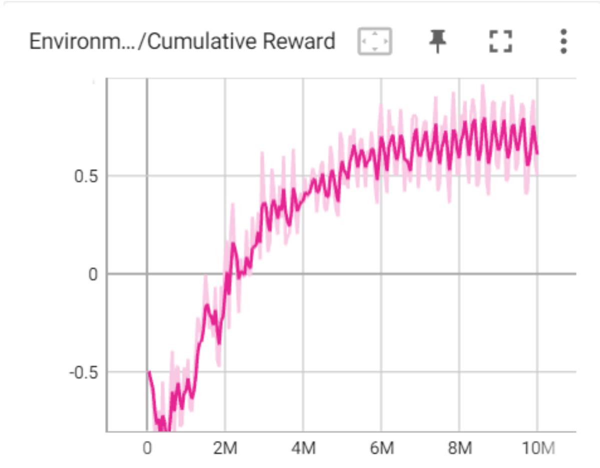

FIGURE 11: Cumulative Reward

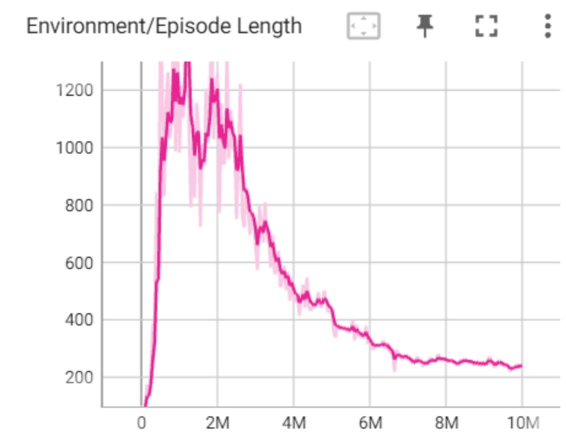

FIGURE 12: Episode Length

In Figure 12, the y-axis represents Episode Length which describes how many time steps or actions an agent completes during a single interaction episode with an environment. A self-contained series of actions and observations that the agent goes through to complete a particular job or aim is represented by an episode.

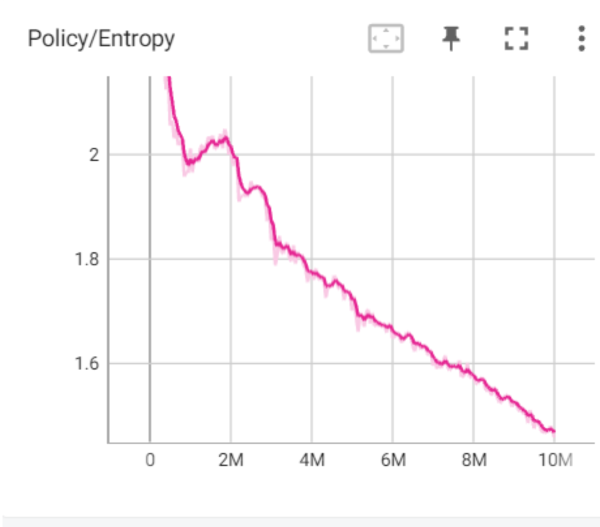

FIGURE 12: Entropy

In Figure 13, the y-axis represents Entropy which refers to a measurement of randomness or uncertainty in the agent's distribution of policies. The policy distribution represents the probability distribution over potential courses of action that an agent might pursue in a specific environmental state. Entropy is determined based on the odds that the agent's policy has allocated to certain acts. A more varied and experimental approach, where the agent is unsure which course of action to take, is indicated by a greater entropy value. In contrast, a lower entropy number denotes a more exploitative and deterministic behavior, where the agent is more certain in choosing a particular course of action.

5. Conclusion

In conclusion, creating and developing a virtual world with a narrative-driven experience with RPG elements using the new hand tracking technology provided by the Oculus 2 headset was a challenge but it succeeded overall. One of the hardest aspects was the integration of unique gestures used for interaction with the world while trying to make it comfortable for the player, as well as the development of the reinforcement learning agent. However, as mentioned previously, the reinforcement learning agent reached satisfactory results with very stable training using the MLAgents package provided by Unity and could perform well in the game environment.

References

[1] Ludlow, Barbara L. "Virtual reality: Emerging applications and future directions." Rural Special Education Quarterly 34, no. 3 (2015): 3-10.

[2] Hvass, Jonatan, Oliver Larsen, Kasper Vendelbo, Niels Nilsson, Rolf Nordahl, and Stefania Serafin. "Visual realism and presence in a virtual reality game." In 2017 3DTV conference: The true vision-capture, Transmission and Display of 3D video (3DTV-CON), pp. 1-4. IEEE, 2017.

[3] Lewis, Gwyn N., Claire Woods, Juliet A. Rosie, and Kathryn M. Mcpherson. "Virtual reality games for rehabilitation of people with stroke: perspectives from the users." Disability and Rehabilitation: Assistive Technology 6, no. 5 (2011): 453-463.

[4] Monteiro, Diego, Hao Chen, Hai-Ning Liang, Huawei Tu, and Henry Dub. "Evaluating performance and gameplay of virtual reality sickness techniques in a first-person shooter game." In 2021 IEEE Conference on Games (CoG), pp. 1-8. IEEE, 2021.

[5] Elor, Aviv, and Sri Kurniawan. "Deep reinforcement learning in immersive virtual reality exergame for agent movement guidance." In 2020 IEEE 8th International Conference on Serious Games and Applications for Health (SeGAH), pp. 1-7. IEEE, 2020.

[6] Coppens, Youri, Eugenio Bargiacchi, and Ann Nowé. "Reinforcement learning 101 with a virtual reality game." In Proceedings of the 1st international workshop on education in artificial intelligence K-12. 2019.

[7] Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. "Proximal policy optimization algorithms." arXiv preprint arXiv:1707.06347 (2017).

[8] Silver, David, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot et al. "Mastering chess and shogi by self-play with a general reinforcement learning algorithm." arXiv preprint arXiv:1712.01815 (2017).

[9] Balducci, Fabrizio, Costantino Grana, and Rita Cucchiara. "Affective level design for a role-playing videogame evaluated by a brain–computer interface and machine learning methods." The Visual Computer 33 (2017): 413-427.

[10] AbdElminaam, Diaa Salama, Essam H. Houssein, Mokhtar Said, Diego Oliva, and Ayman Nabil. "An efficient heap-based optimizer for parameters identification of modified photovoltaic models." Ain Shams Engineering Journal 13, no. 5 (2022): 101728.

[11] Abdelminaam, Diaa Salama, Nabil Neggaz, Ibrahim Abd Elatif Gomaa, Fatma Helmy Ismail, and Ahmed A. Elsawy. "Arabicdialects: An efficient framework for arabic dialects opinion mining on twitter using optimized deep neural networks." Ieee Access 9 (2021): 97079-97099.

[12] Rajeena PP, Fathimathul, Rasha Orban, Kogilavani Shanmuga Vadivel, Malliga Subramanian, Suresh Muthusamy, Diaa Salam Abd Elminaam, Ayman Nabil, Laith Abulaigh, Mohsen Ahmadi, and Mona AS Ali. "A novel method for the classification of butterfly species using pre-trained CNN models." Electronics 11, no. 13 (2022): 2016.

[13] Abd Elminaam, Diaa Salama, Walaa Hassan Elashmawi, and Shaimaa Abdallah Ibraheem. "HMFC: Hybrid MODLEM-Fuzzy Classifier for Liver Diseases Diagnose." *Int. Arab. J. e Technol.* 5, no. 3 (2019): 100-109.

[14] Ai, Mona AS, Anitha Shanmugam, Suresh Muthusamy, Chandrasekaran Viswanathan, Hitesh Panchal, Mahendran Krishnamoorthy, Diaa Salama Abd Elminaam, and Rasha Orban. "Real-time facemask detection for preventing COVID-19 spread using transfer learning based deep neural network." *Electronics* 11, no. 14 (2022): 2250.

[15] Elminaam, Diaa Salam Abd, Nabil Neggaz, Ibrahim Abdulatief Ahmed, and Ahmed El Sawy Abouelyazed. "Swarming Behavior of Harris Hawks Optimizer for Arabic Opinion Mining." *Computers, Materials & Continua* 69, no. 3 (2021).

[16] AbdElminaam, Diaa Salama, Nabil Neggaz, Ibrahim Abd Elatif Gomaa, Fatma Helmy Ismail, and Ahmed Elsawy. "Aom-mpa: Arabic opinion mining using marine predators algorithm based feature selection." In *2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, pp. 395-402. IEEE, 2021.

[17] Abd-Elmiaam, Diaa SALAMA, and Shaimaa ABDALLAH Ibrahim. "Building a robust heart diseases diagnose intelligent model based on RST using lem2 and modlem2." In *Proc. of the 32nd IBIMA Conference: Seville*. 2018.

[18] Ali, Mona AS, Fathimathul Rajeena PP, and Diaa Salama Abd Elminaam. "A Feature Selection Based on Improved Artificial Hummingbird Algorithm Using Random Opposition-Based Learning for Solving Waste Classification Problem." *Mathematics* 10, no. 15 (2022): 2675.

[19] Mahmoud, Eman, HM Abdul Kader, and DS Abdul Minaam. "Fuzzy knowledge base system for floating car data on sumo." In *2019 29th International Conference on Computer Theory and Applications (ICCTA)*, pp. 38-42. IEEE, 2019.

[20] Ong, Song-Quan, Gomesh Nair, Ragheed Duraid Al Dabbagh, Nur Farihah Aminuddin, Putra Sumari, Laith Abualigah, Heming Jia, Shubham Mahajan, Abdelazim G. Hussien, and Diaa Salama Abd Elminaam. "Comparison of pre-trained and convolutional neural networks for classification of jackfruit artocarpus integer and artocarpus heterophyllus." In *Classification applications with deep learning and machine learning technologies*, pp. 129-141. Cham: Springer International Publishing, 2022.

[21] Ali, Mona AS, Rasha Orban, Rajalaxmi Rajammal Ramasamy, Suresh Muthusamy, Saanthoshkumar Subramani, Kavithra Sekar, Fathimathul Rajeena PP, Ibrahim Abd Elatif Gomaa, Laith Abulaigh, and Diaa Salam Abd Elminaam. "A Novel Method for Survival Prediction of Hepatocellular Carcinoma Using Feature-Selection Techniques." *Applied Sciences* 12, no. 13 (2022): 6427.

[22] Neggaz, Nabil, and Diaa Salama AbdElminaam. "Automatic sport video mining using a novel fusion of handcrafted descriptors." In *2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, pp. 387-394. IEEE, 2021.

[23] Abd Elminaam, Diaa Salama, Asmaa El Tanany, Mustafa Abdul Salam, and Mohamed Abd El Fattah. "CPSMP_ML: Closing price Prediction of Stock Market using Machine Learning Models." In *2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, pp. 251-255. IEEE, 2022.

[24] Ali, M.A., Balasubramanian, K., Krishnamoorthy, G.D., Muthusamy, S., Pandiyan, S., Panchal, H., Mann, S., Thangaraj, K., El-Attar, N.E., Abualigah, L. and Abd Elminaam, D.S., 2022. Classification of glaucoma based on elephant-herding optimization algorithm and deep belief network. *Electronics*, *11*(11), p.1763.

[25] Csereoka, Petra, Bogdan-Ionuţ Roman, Mihai Victor Micea, and Călin-Adrian Popa. "Novel Reinforcement Learning Research Platform for Role-Playing Games." *Mathematics* 10, no. 22 (2022): 4363.

[26] Wu, Yuechen, Yingfeng Chen, Xiaofei Xie, Bing Yu, Changjie Fan, and Lei Ma. "Regression testing of massively multiplayer online role-playing games." In 2020 IEEE international conference on software maintenance and evolution (ICSME), pp. 692-696. IEEE, 2020.