Implementing Artificial Intelligence Agent Within Connect 4 Using Unity3d and Machine Learning Concepts

Nirmal Baby, Bhargavi Goswami

Abstract:*Nowadays, we come across games that have unbelievably realistic graphics that it usually becomes hard to distinguish between reality and the virtual world when we are exposed to a virtual reality gaming console. Implementing the concepts of Artificial Intelligence (AI) and Machine-Learning (ML) makes the game self-sustainable and way too intelligent on its own, by making use of self-learning methodologies which can give the user a better gaming experience. The use of AI and ML in games can give a better dimension to the gaming experience in general as the virtual world can behave unpredictably, thus improving the overall stigma of the game. In this paper, we have implemented 'Connect-4', a multiplayer game, using ML concepts in Unity3D. The machine learning toolkit 'ML-Agents', which depends on Reinforcement Learning (RL) technique, is provided using Unity3D. This toolkit is used for training the game agent which can distinguish its good moves and mistakes while training, so that the agent will not go for same mistakes over and over during actual game with human player. With this paper, authors have increased intelligence of game agent of Connect 4 using Reinforcement Learning, Unity3D and ML-Agents toolkit.*

Index Terms: *Artificial Intelligence, Machine Learning, Connect four, Game theory, Reinforcement Learning, Unity3D, ML-Agents*

I. INTRODUCTION

Artificial Intelligence (AI) is the study and production of computer frameworks that can see, reason and act [1]. The fundamental purpose of AI is to create smart machines. The intelligence ought to be displayed by considering, deciding and tackling issues significantly by learning methods. In a simple manner, the definition of AI can be said to be the implementation of a computer, a computer controlled automaton or a software which is capable of intelligent thinking. Game can be defined as an activity which can be played according to a certain set of established rules. Usually the player needs to achieve some goals by competing against non-playable characters (NPC). Artificial intelligence has a vital role in game programming. Consider a scenario within a game where the player needs to combat enemy NPCs.To make these NPCs formidable opponents, they need to be able to perform intelligently or in simple terms, adapt to the performance of the player. This can be achieved by using a variety of methods collectively known as AI methods.

Nowadays, AI is implemented in many games. Different AI techniques are used in different games. This is according to the genres of the game. The game can be categorized into different genres such as adventure games, First Person Shooting (FPS) games, board games etc. Here, the implementation is done using the board game "Connect-4". Connect-4 is basically a two player game. The game consists of a board which has several cells that are arranged into a matrix of rows and columns. Usually the board contains six rows and seven columns. The main action involved in the game is the dropping of the coins by each player, vertically into the cells. These moves are performed alternatively by the players and the goal of each player is to get four coins of the same color to be arranged consecutively in a row or column or diagonal manner. The player who manages to get such a set is considered to be the winner [2]. In this game, an AI algorithm is needed to train the system, such that the system can simulate a human in terms of intelligence. By making use of the RL concept and Unity3D, the game's AI agents can be trained for better user experience. The implementation of RL concepts in different games makes those games NPCs more intelligent. The paper [3] shows the successful implementation of RL concepts in the game Othello.Fig. 1 shows the basic structure of Connet-4 game. In which one player represents yellow coin and another player represents red coin.

II. RELATED WORKS

In [4] a neural network is developed which can learn to play the board game Five-in-a-row. An algorithm in Reinforcement Learning is used as the training algorithm for the AI part. First they developed a network topology to learn how to examine a particular position in the board and to find which move should be produced next. The evaluation of all non-occupied board positions and determination of next move is learned by a reinforcement algorithm.

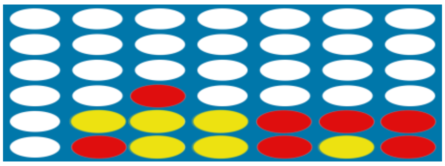

Fig.1 – Connect – 4 game

Revised Manuscript Received on December 22, 2018

Nirmal Baby,Department of Computer Science, Christ (Deemed To Be University) Bangalore, nirmalbaby1113@gmail.com

Bhargavi Goswami,Department of Computer Science, Christ (Deemed To Be University) Bangalore, bhargavigoswami@gmail.com

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

In [5] they introduces a new model which is named as "Neural Connect-4" that can play the board game ‘Connect-4’. It uses one of the common supervised learning algorithm called as "Back propagation algorithm". As the neural network topology, multilayer perception with one hidden layer is used. The training is done using back propagation algorithm and sigmoid is used as the activation function.

In ref. [6] deals with the implementation of reinforcement learning in a Tank battle game as game AI. The reinforcement learning gives each Non-Playing-Character (NPCs) an ability to think like human in this game scenario. Two experiments were done one is the NPC tanks moves by point by point and another is the tank movement is in different ways. These results are taken and plotted a graph. One of the drawbacks of the model is, for achieving the goal many attempts is required.

In [7] theyimplemented multi-stage TD (Temporal-Difference) learning in 2048 game. This method is one of the hierarchical method in reinforcement learning. The learning process is divided into multiple stages where each stage have its own learning agents and sub goals. They used n-tuple network along with multi-stage TD for "2048 game" and for the game "Threes". They also show how the game is improved after implementing multi-stage TD learning.

The paper [8] does the evaluation of Q-learning and sarsa algorithm, on RTS game called battle city. They developed an algorithm which implements the reinforcement algorithm in the game battle city. The two different reward functions which they used are; generalized and conditional. Main advantage of implementing this reinforcement algorithm is that system can quickly switch the strategies with respect to enemies and the learning process can be done through the interaction with opponents, also any type of human traces is not needed. The paper concluded by stating that SARSA algorithm requires less time to learn and win when compared to Q-learning.

The paper [9] introduced a new algorithm to develop the game Tic Tac Toe which is AI based. Soft computing techniques are used to design the algorithm and the game play is developed using different tools like JavaScript, HTML and CSS. Algorithm makes the system to find the efficient move which can result in either win or draw. The algorithm is separated into five sections, here we assume that computer is playing as ‘O’ and player is playing as ‘X’. The AI part of the game works according to the five sections described in the paper [9].

The paper [10] discusses the Tic Tac Toe playing algorithm’s evolving by using Co-Evolution, Interactive Fitness and genetic programming. The different selected tree-structured algorithms are examined using fitness-less game strategy and then plays against a human player. The final evolved algorithms are efficient for playing against human players. Random sampling is done to select a number of different individuals to play against the individual that was to be evaluated. Co-Evolution can be defined as fitness-less evolution of individuals which are being improved by playing against one another. To find the fitness of individuals, they consider the number of wins an individual can obtain by playing against the randomly selected group. Then Interactive fitness evaluation method is used so that they are able to enhance the obtained individuals in order to get a human-competitive algorithm.

In [11] Chen-Huei Chou used Tic Tac Toe for learning data mining, classification, and evaluation. He selected the 3x3 grid board for Tic Tac Toe game. In his paper he deals with two aspects, one is to analyze whether machine leaner’s can classify Tic Tac Toe game successfully and another one is to find out whether novices learning data mining classifications can successfully conduct experiments to evaluate the performance of machine learners using different evaluation methods. In this study seven machine learners and three evaluation methods are used. The final result shows that machine classifiers can easily judge the finished games and novice can correctly conduct evaluations.

The paper [12] describes the work done on the board game of Chung Toi, a little bit complicated version of Tic-Tac-Toe. Reinforcement learning is implemented in this game. They used neural network and temporal difference algorithm for training the network. The trained network won 90% of the game which it plays against a smart random player.

In [13] they introduced a new algorithm which can be implemented using a modified Neural Network (NN) and Genetic Algorithm (GA) is used for training the network. The neurons which is used in this network has two activation functions and those neurons shows a node to node relationship in the hidden layer. This increase the learning ability of the network.

The paper [14] discuss about different learning strategies for text-based games. These type of game’s all communication with the virtual world are through the text. They used deep reinforcement learning with game rewards as the feedback for the model. The text descriptions are mapped to vector descriptions which can get all the game states. The algorithm they proposed outperforms the baselines on two worlds (bag-of-words and bag-of-bigrams) which gave the importance of learning expressive representations.

III. PROBLEM STATEMENT

Presently, the backtracking algorithms of AI is used for “Connect-4” game. The main algorithms used are Min-Max and Alpha-Beta which work by using recursive functions and the game tree concept. The algorithm doesn’t impart any training for AI agents. The focus of this paper is to develop a game which makes use of a trained AI agent that is capable of thinking like a human. In a game scenario, humans are able to compare the present state to past states such that any mistake that they may have previously made, can be skipped for better game result, which is considered as a desirable quality for any trained AI game agent. The agent training takes place using RL (Reinforcement Learning) technique and it is a continuous process such that the agent can learn from its past mistakes and rectify them in its present state, while also tackling a new and different environment at the same time.

IV. IMPLEMENTATION

As discussed earlier, the training of AI agents takes place using the RL technique and the game is developed using Unity3D. There are several concepts in play here which work together to produce an intelligent game, capable of mimicking a human. The section 4.1 discusses all the tools and concepts (such as Unity3D, Visual Studio, ML-Agents etc.), and the section 4.2 explains the procedure of implementing the Ml-Agents in Connect-4 game step by step. The algorithm developed in section 4.2 consist of seven steps in which the development of trained model described from creating the environment to assigning positive and negative reward to the actions done by the agent in the environment for each game state. The Fig.3 represents the flow of training an agent using ML-Agents toolkit in Unity3D.

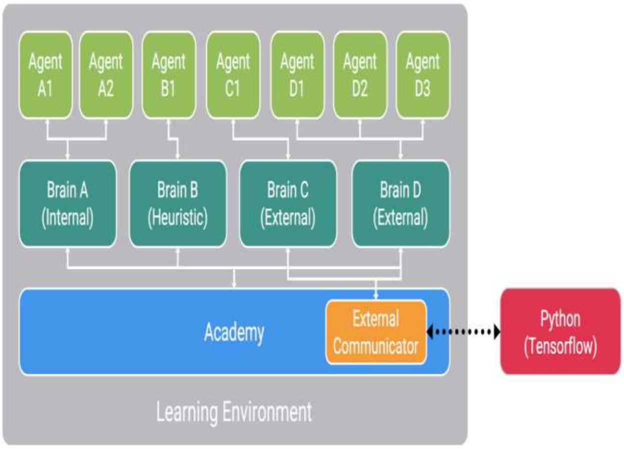

Fig.3 – Flow of training an agent

A. CONCEPTS AND TOOLS

Reinforcement Learning: It is one of the most common concepts in ML which is used to train an AI agent. It works based on an action-reward concept. The Fig.2 shows the basic flow of RL.

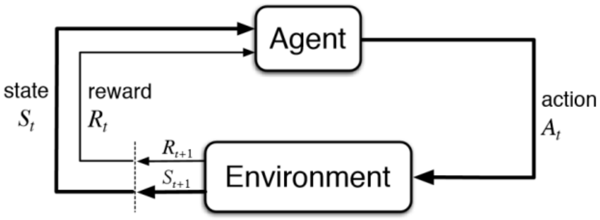

Fig.2 – Flow of Reinforcement Learning

The learning of the agent occurs in accordance with the actions that the agent performs within the gaming environment states. During the training of agents that takes place for each game state, specific rewards are given to the agent that can either be a positive reward (given when agent performs a correct action) or negative reward (given for the mistakes done by the agent). According to this system of rewards, maximizing the positive rewards becomes the primary objective of the agent [15].

Unity 3D: It is a popular game development engine which is used to develop 2D & 3D games. It comes with most of the basic and advanced features that are required for game development. The ML-Agents toolkit is a machine learning toolkit for games which is developed using Unity. Here, the ML-Agent toolkit is used to train the game agent of Connect-4 game [16].

Visual Studio C#: The IDE visual studio is used for C# programming. Unity3D supports C# and JavaScript. In this article C# is used. The rewards, actions and all the agent training related logic is done using C# [17].

ML-Agents: The Unity ML-agents toolkit aid to make games and create simulations that act as an environment for the game agent's training. The toolkit uses different methods for training such as reinforcement learning, neuroevolution, imitation learning etc. The toolkit can be easily used to train the game agents for 2D, 3D and VR (Virtual Reality)/AR (Augmented Reality) games [18].

TensorFlow: It is an open source library which is used to train the agent's brain externally in unity. With respect to ML-Agents toolkit, soon after the training of a game agent, the output is a file which is in tensorflow model (.bytes). This file is then later combined with an internal brain [19].

TensorFlowSharp: TensorFlow doesn't support C# API and this can be considered as one of the main disadvantages of tensorflow. But unity scripts are written in C#. So a third-party library TensorFlowSharp is used to combine the tensorflow model which is in .bytes format with Unity3D for the development of novel model for the Intelligent Gaming [20].

B. METHODOLOGY

Forimplementing the agent in Connect four game along with Unity3D, ML-Agents toolkit and TensorFlow library are used [21].

The training of the agent is implemented as:-

Step 1: Create Environment for ML-Agents to train

using Unity3D. Every environment contains an academy, agents and brains.

Step 2: Implementation of academy: All the changes that happens to the environment is done here. InitializeAcademy(),AcademyStep(),AcademyReset()functions are implemented according to the problem.

Step 3: Implementation of Agents: Here authors have used one agent for training. Agent initialization, collection of observations from the environment, different actions taken by the agent in each step, agent reset is performed in agent implementation.

Step 4: Here, the game board of Connect-4 is used as the environment for the agent to train.

Step 5: Each cell in the game board is considered as an empty game object and at each step the agent takes the X-axis and Y-axis of each empty game objects as the observation from the environment. Agent needs to collect 84 observation at each step (because of 42 cells in the board).

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Step 6: These observations will act as the inputs to the brain which contain neural model. Then the model gives two output values (Proximal Policy Optimization algorithm is used).

Step 7: These values are checked and accordingly the rewards are given in the following manner. a) High positive reward is given if the output values are equal to the X-axis and Y-axis of an empty game object (means cell in the board) and also all the cells below these cells should be filled or not empty. b) Another two situations where high positive reward is given are: i) when the predicted cell results in game win state. ii) When the predicted cell defends the opponents chance to win. c) Less positive reward is given if the predicted cell makes the coin three or two in a row and little less reward is given if the predicted cell block opponents chance to make two or three coins in a row. d) High negative reward is given, if the output values are not equal to the X-axis and Y-axis of an empty game object (means cell in the board). e) Negative reward is given if the opponent wins the game.

So, slowly the agent will began to learn in the environment from its mistakes using the reward – action method which is used as the concept of reinforcement learning. The trained model will be a TensorFlow model which can be integrated in Unity3D using TensorFlow plugin and later the file is combined with internal brain.

V. RESULTS

The screenshots shown in this section, is according to the game flow from the starting to the end of the game until one (either computer or human) is won. Here the game is played between two players. One player is human and another one is the game agent which is trained through the algorithm we discussed in methodology section. The red dots are denoted by the game agent and the yellow dots are denoted by the human players.The game is designed as like each player can make move alternatively. Here the human player makes a move then the trained model will respond to that move using the algorithm which is stated in section 4.2. So the screenshot of entire single gameplay which is implemented using the above algorithm is as follows.

Fig.4 represents the very first screen appears when the game opens. The screen appears at the beginning of the game, when the player clicks on play now from game main menu is shown in Fig.5. The game state after the first move made by both human player and game agent is shown in Fig.6. The Fig 7.shows the state of the game after six successful moves made by both the players. The Fig 8.shows the final game state where the player who plays with red makes the wining move. As shown in Fig 8, the game agent has obtained four consecutive red coins in vertical on third row which makes the game agent as winner of the game. So the game ends here and the game agent wins over the human player. Fig 9 shows final game over screen. In short all the figures from Fig. 4 to Fig. 9 represents the states of the game from the beginning of the game to the end of the game. The screenshot provided here is taken after integrating the trained ML-Agents in Connect-4 game using Unity3D.

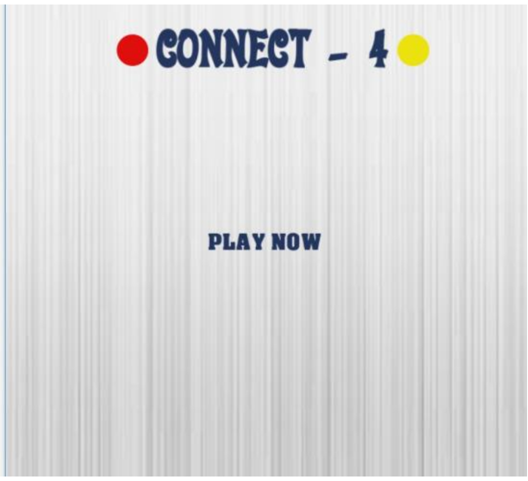

Fig.4 – Game main menu

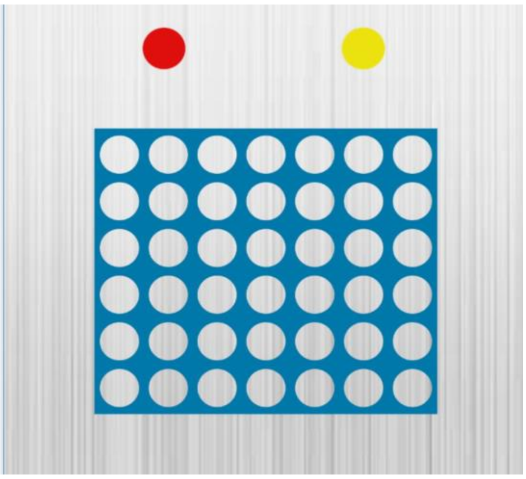

Fig.5 – Initial game state

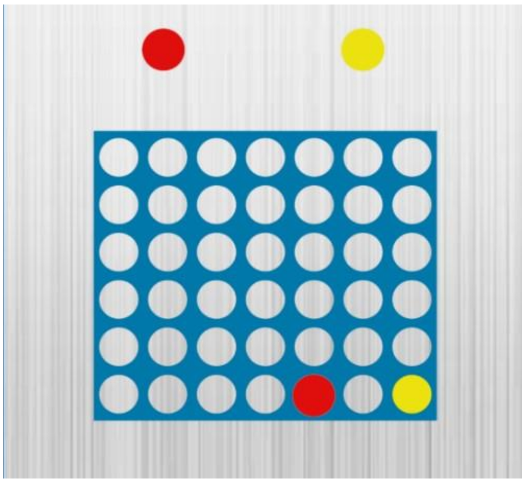

Fig.6 – Gameplay screen

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

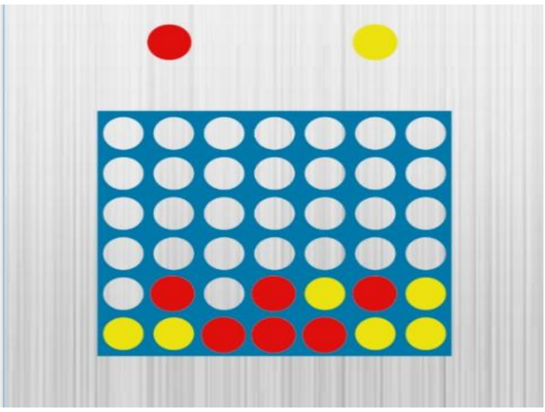

Fig.7 – Gameplay screen

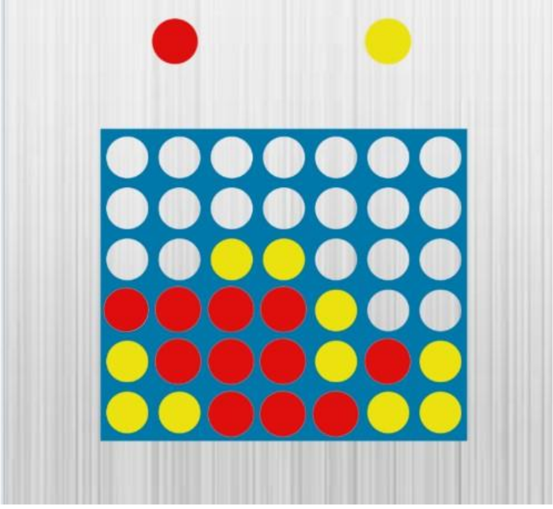

Fig.8 – Gameplay screen

Fig.9 – Game over screen

Thetraining of the agent is done with TensorFlow library using python. PPO algorithm and reinforcement learning are the concepts which is used for training the agent. The progression of training and the training evaluation can be observed from eight different graphs. Fig.10 represents Lesson graph, which is considered only in the case of curriculum training. In this experiment, it doesn't have any impact on performance. Mainly this graph produce outputs when the imitation training is happening.Fig.11 represents Entropy graph where it shows how much randomly the decisions are taken by the model. During a successful training the graph slowly decreases. The graph shows that the developed model slowly starts learning and thus gradually reduces random decision making. Fig.12 represents Cumulative Reward. The mean cumulative reward which is obtained by all the agents at each episode. While training the agent the graph increase if the training progress is successful. The graph shows how gradually the rewards increases. The model maintain the cumulative reward in the range of 60 to 90. The cumulative reward graph goes high from certain point which shows that the agent began to train with good moves. Positive rewards are assigned to good moves and negative rewards are assigned to bad moves or mistakes done by the agent with respect to the current game state. Fig.13 represents Episode Length graph where, for all the agents in the environment it shows the mean length for each episode. It shows the length of each episode. In Fig.14 represents Policy Loss graph which shows the change in the policy while training. The graph should decrease while training. As seen in the graph, it reduces at 4.000K to 15.00K which shows that how much the policy changes during a training session. Fig.15 represents Learning Rate graph.While searching for the optimal policy how large a step the training agents takes, while training. Over the time graph should decrease because, the training algorithm should take less time in each step to find the optimal policy.Here each graph is plotted with respect to the values in X-axis and Y-axis. The horizontal axis or X-axis represents the number of steps that taken by the agent in training and the vertical axis or Y-axis is represents the different values according to each graphs. Entropy graph: the Y-axis represents the randomness of the decision taken by the agent, Cumulative reward graph: the cumulative rewards the agent acquire while training, Learning graph: the number of steps taken by the agent for searching the optimal policy respectively. Episode length graph: length of each episodes, Policy loss graph: the magnitude of policy loss function, Value loss graph: the magnitude of value function, Value estimate graph: shows the value estimate for every states which is visited by the model.

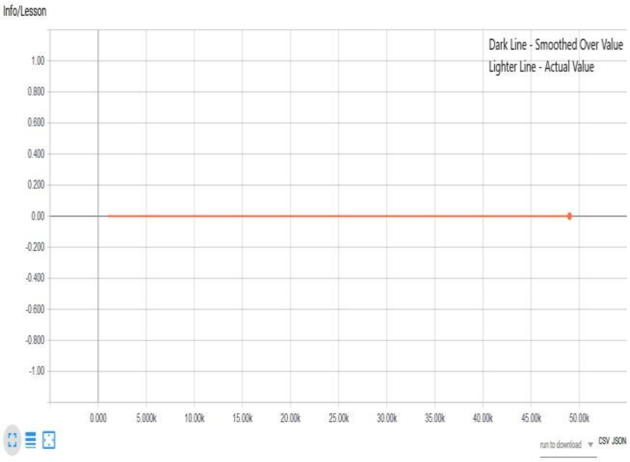

Fig.10 – Lesson graph

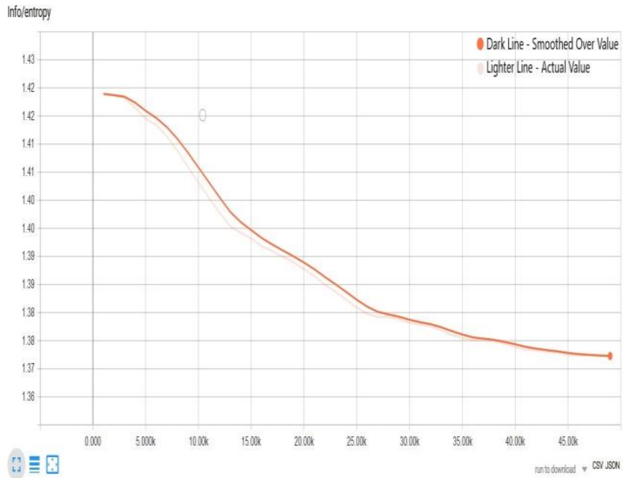

Fig.11 – Entropy graph

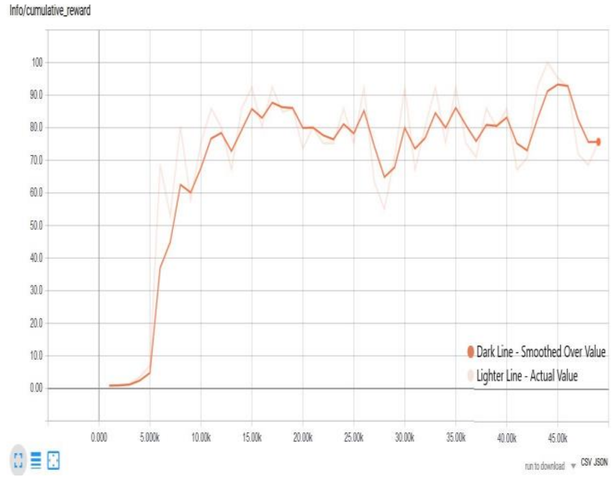

Fig.12 – Cumulative reward graph

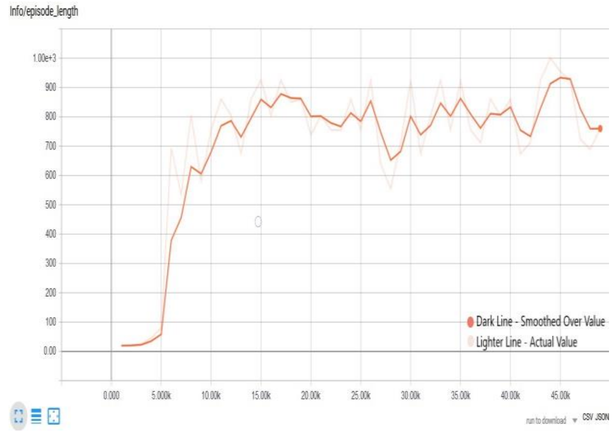

Fig.13 – Episode length graph

The graph shows two different lines, one is darker and another is lighter. The lighter line represents the actual value while training the agent and the darker line represents the smoothed over value which can be adjusted in TensorFlow while training.

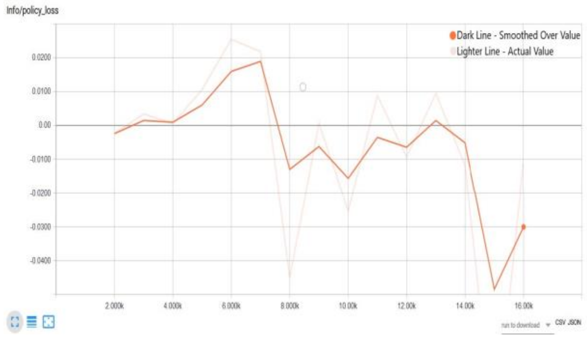

Fig.14 – Policy loss graph

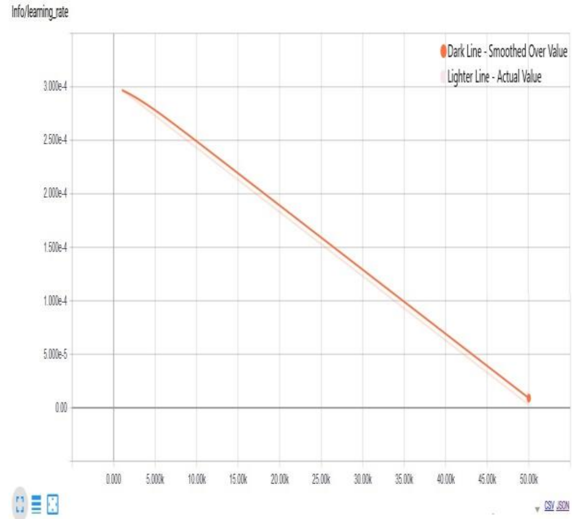

Fig.15 – Learning rate graph

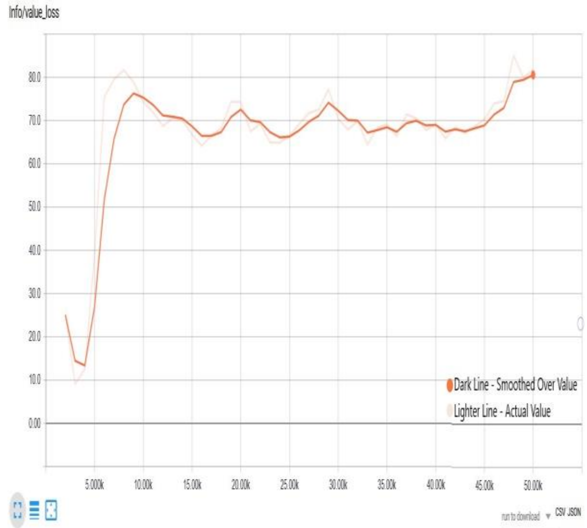

Fig.16 – Value loss graph

Fig.16 represents Value Loss graph which shows at each state how well the prediction of the value by the model occurs.Fig.17 represents Value Estimate which estimates the mean value for all the states that the agents visited. Graph shows constant increase while a successful training happens.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

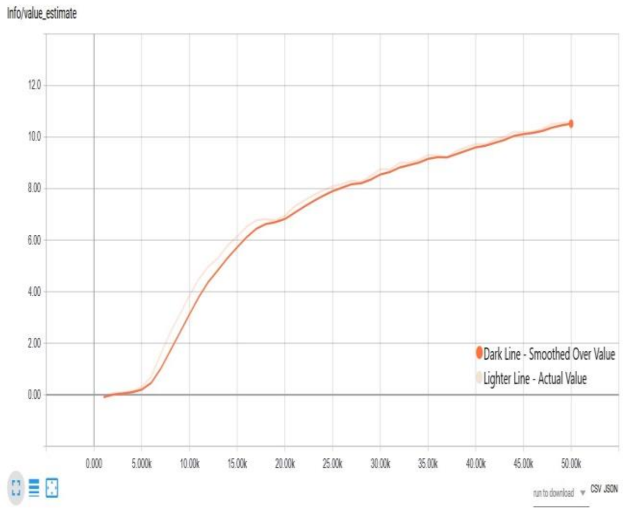

Fig.17 – Value estimate graph

So, these are the graphs which shows the training progress and the agent's training was successful or not.

VI. CONCLUSION

Artificialintelligence in games means to provide intelligence to the Non-Playing-Characters in the game. With respect to board games, this can be done through different training methods. One of the main training techniques used is reinforcement learning methods. Through this technique, the AI agent can learn from their mistakes and can predict the most relevant move in each cases. Here, in this paper the ML-agents toolkit is used to train the agent for Connect-4 game. The agent is trained using reward-action method which is called as Reinforcement learning and neural networks along the PPO algorithm. For each move made by the agent in training section is rewarded. Agent always tries to maximize its reward which makes the agent to learn good moves. Finally, the well learned agent after certain number of episodes is integrated in Unity3D and act as the internal brain of the game (i.e., act as the game AI).

Oneof the main future enhancement for this work can be treated as the training of the agent while playing the game. Here in this research the mistakes appears while training the agents is considered and awarded negative rewards so that the agent tries to avoid those moves when it appears in same game state. So, if the agent can consider mistakes or bad moves done by the agent while playing the game (after training), then by completing each game with the human player, the agent became more efficient.

REFERENCES

1. (ScienceDaily)*Artificial Intelligence*[Online]Available:https;//www.sciencedaily.com/terms/artificial_intelligence
2. Victor Allis, "A knowledge-based approach of Connect Four". *Published by the faculty of general sciences at the University of Limburg, Netherlands. Article in ICGA (International Computer Games Association) journal 11(4), March 1994*
3. Michiel van der Ree and Marco Wiering, "Reinforcement Learning in the Game of Othello:Learning Against a Fixed Opponent and Learning from Self-Play". *2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*
4. Bernd Reisleben, "A Neural Network that Learns to Play Five-in-a-Row", *Proceedings of Second New Zealand International Two Stream Conference on Artificial Neural Networks and Expert Systems, 1995 IEEE*
5. Marvin Oliver Schneider and João Luís Garcia Rosa, "Neural Connect 4 – A Connectionist Approach to the Game", *Proceedings of the VII Brazilian Symposium on Neural Networks (SBRN'02), 2002 IEEE*
6. Yung-Ping Fang and I-Hsien Ting, "Applying Reinforcement Learning for Game AI in a Battle Tank Game". *Fourth International Conference on Innovative Computing, Information and Control, 2009 IEEE*
7. Kun-Hao Yeh1, I-Chen Wu1, Chu-Hsuan Hsueh1, Chia-Chuan Chang1, Chao-Chin Liang1 and Han Chiang1, "Multi-Stage Temporal Difference Learning for 2048-like Games", *TCIAIG • October 2015.*
8. Harshit Sethy, Amit Patel and Vineet Padmanabhan, "Real Time Strategy Games: A Reinforcement Learning Approach", *Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)*
9. Sunil Karamchandani, Parth Gandhi, Omkar Pawar and Shruti Pawaskar, "A Simple Algorithm For Designing An Artificial Intelligence Based Tic Tac Toe Game", *Conference Paper • January 2015.*
10. Helia Mohammadi, Nigel P. A. Browne, Anastasios N. Venetsanopoulos, and Marcus V. dos Santos, "Evolving Tic-Tac-Toe Playing Algorithms Using Co-Evolution, Interactive Fitness and Genetic Programming", *International Journal of Computer Theory and Engineering, Vol. 5, No. 5, Pages 1 – 5, October 2013*
11. Chen-Huei Chou, "Using Tic-Tac-Toe for Learning Data Mining Classifications and Evaluations", *International Journal of Information and Education Technology, Vol. 3, No. 4, Pages 1 – 5, August 2013*
12. Christopher J. Gatti, Jonathan D. Linton, and Mark J. Embrechts "A brief tutorial on reinforcement learning: The game of Chung Toi", *ESANN 2011 proceedings, European Symposium on Artificial Neural Network, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011*
13. H.K. Lam, S.H. Ling, F.H.F. Leung, P.K.S. Tam, Y.S. Lee, "Playing Tic-Tac-Toe Using a Modified Neural Network and an Improved Genetic Algorithm'",*IEEE 2002 28th Annual Conference of the Industrial Electronics Society. IECON 02*
14. Karthik Narasimham, Tejas Kulkarni, Regina Barzilay, "Language Understanding for Text-based Games using Deep Reinforcement Learning".*EMNLP, 2015, 11 September 2015.*
15. (KDnuggets)*Reinforcement Learning* [Online] Available:https;//www.kdnuggets.com/2018/03/5-things-reinforcement-lesrning.
16. Manu Raghaw, Joy Paulose, and Bhargavi Goswami, "Augmented Reality for History Education", *International Journal of Engineering & Technology, Vol 7, Iss 2.6, Pages 1 – 5, February-2018,SCOPUS, India*
17. (Unity Manual)*Unity Documentation* [Online] Available: https://docs.unity3d.com/Manual
18. (Github-open source)*Unity-Technologies/ml-agents* [Online] Available: https://github.com/Unity-Technologies/ml-agents
19. *TensorFlow-Framework*[Online]Available: https://www.tensorflow.org/
20. (Github)*TensorFlowSharp*[Online] Available: https://github.com/migueldeicaza/TensorFlowSharp
21. *TensorFlow Libraries* [Online] Available: https://machinelearningmastery.com/introduction-python-deep-learning-library-tensorflow/

***NIRMAL BABY**, pursuing Master of Computer Application (MCA) in Christ (Deemed To Be university), Bangalore. Interested in the field of gaming with a lot of passion for game development and programming. In game development field he focuses on Artificial Intelligence and Machine Learning concepts in games. Other than game development field he also interested in Android mobile application development.*

***BHARGAVI GOSWAMI**, Short Dr. Bhargavi Goswami received her BCA from Saurashtra University in 2006. In 2009, she received her MCA from Sardar Patel university. She further received UGC funding for research worth 17000 USD during her Ph.D. She was awarded travel grant to Hong Kong by SIGCOMM Conference. She received best Research Proposal Award from Doctoral conference at Udaipur. She has 2 patents on the research conducted and converted to industry projects. She has served bodies like IEEE as reviewer and organizing committee for conducting IEEE conferences. Currently she is serving the research community by imparting her research and development skills to her research students along with teaching in Christ (Deemed To Be University), Bangalore as Assistant Professor working in the research area of Artificial Intelligence, Machine Learning project modeling, Design Analytics and Software Defined Networks.*