Artificial Intelligence in Games

Raj Sudhir Pawar
Department of Information Technology
B.K. Birla College of Arts, Science & Commerce (Autonomous), Kalyan, India

Abstract— Artificial Intelligence is a Vast Scientific Field, with tremendous achievements. Interactive Computer Games have been getting rising popularity during recent years, which has artificial intelligence at its base. Some AI is used for collecting data and studying the human brain while they Play Games. Computer VR-games with decent AI are/can help in other fields through a virtual simulation like Flight stimulation, Medical Purpose, Educational Purpose, etc. But Game AI is an effort of going beyond the scripted Interactions for a better experience. The goal of this research is to realistic user experience. Games appear to an excellent platform for various problems related to the challenging field of artificial intelligence.

Keywords: Artificial Intelligence, Computer Games

I. INTRODUCTION

Artificial Intelligence (AI) takes an important place in games of any genre.AI is used to create an environment for improved gameplay that would help discover the peculiar properties of a game world. Most of the games are based on AI-driven characters and their reflections on the player's actions. Complex behavior rules are applied to such characters. One of the most complex and vast cases of AI development is a game with an open-world experience. The key feature of this type of computer game is a player's ability to interact with any in-game object or Non-Playable Character (NPC) at any time. Usually to provide such ability game developers create a big world populated by many characters. Nowadays game developers make the game a more open world with a bigger size for a more realistic and better experience. However, this raises the required number of characters to inhabit the world, like Towns, Villages, and countries need a large number of agents that implement territory dependent behavior. Moreover, the in-game AI must provide every NPC with unique behavior to make it feel different. The behavior of NPC must look realistic and does not follow a pre-scripted behavior. There could be thousands of such characters in modern games. The problem of the creation of so many characters with different and complicated behavior is quite vast and requires a modern and complex approach to solve it. On the other hand, methods that allow modeling crowd dynamics with high precision are used in a multi-agent simulation to create agents. An example of methods used is experimenting to test an agent behavior inside a building in a dense crowd. Advanced behavior frameworks can process multiple agents with varying in real-time. Group behavior and realistic agent movement are the main interest points in such experiments. Consequently, two actual problems can be stated. The first one is the problem of the creation of a large number of characters with different and convincing/believable behavior to inhabit in-game worlds. The second one is the poor ability to perform experiments in pedestrian dynamics. A fresh approach is needed to solve them. In this paper, a model regulating NPC's in in-game interaction and decision-making is proposed. It is based on an integration of methods of a multi-agent simulation and modern game development methods of an AI creation. The proposed model allows the creation of NPCs with a believable behavior that fits the last game development trends and an in-game environment mapping for the area and interactive object positioning.

II. USES OF AI IN GAMES

The same type of reinforcement learning that is being used in the video game industry is also being successfully applied to other industries. Not only is it safe and realistic, but it is also much faster to collect data in a virtual environment compared to the real world.

A. Playing against an AI:

Recently Elon Musk has warned the world that the fast development of AI with learning capability by Google and Facebook is going to put humanity in danger. These types of statements and arguments have drawn public attention to the topic of AI. AlphaGo is trained by observing many no. of historical Go matches and it is still learning from playing with human players online. However, the term "AI" in the video game context is not limited to this self-teaching AI. Rather than learn how best to beat human players, AI in the games is designed for increasing the user experience and joy of playing the game. The most common role of AI in video games is controlling non-player characters (NPCs).

B. Challenges in Computer Game AI:

While developing Artificial Intelligence for Computer games, the issues arrived are briefly described as follows.

- Knowledge engineering: even assuming that strategies or behaviors are handcrafted, authoring these behavior sets in a game requires a huge human engineering effort. Game developers have to encode all the knowledge they have about a field (either to achieve a strategic behavior or convincing human behavior) in some sort of behavior language.
- Authoring support: handcrafted behaviors are, ultimately, software code in a complex programming language, prone to human errors. The behavior errors could be in the form of program faults (bugs) or not achieving the desired outcome.
- Replayability and variability: a player might get bored of seeing the same strategies and behaviors repetitively. Although simple variability can be achieved by a stochastic selection of behaviors or strategies from a large repository, this increases the authoring burden.
- Unpredicted situations: it is not possible to anticipate all possibilities and player strategies that can be encountered during gameplay. This makes it difficult to craft believable behaviors that react properly to these unforeseen circumstances and player actions.

III. DECISION MAKING

A Decision-making system is a necessary component in any in-game AI. Such systems are used to provide NPCs with the ability to choose an option amongst possible ones and allow them to describe behavior. NPCs can use the provided ability to find a particular object in an in-game world or to make a decision about certain tasks. There are various approaches to the creation of such systems in the game development industry. The most potential and frequently used ones are described below with an analysis of their advantages and disadvantages.

A. Planned Behavior:

The most primitive in terms of internal architecture is the approach that uses a predefined NPCs behavior stored in an in-game logic. This results in a non-flexible, pre-planned behavior that doesn't provide any choices for the NPCs to make. This approach is useful for the creation of everyday behavior when a particular action must occur at a particular time. Action changes in strict order according to the schedule. However, such behavior is preferable only in cases of a fixed scenario. But in some cases, this approach takes more time to implement. It doesn't meet the requirements or expectations of the current work in the creation of dynamic behavior.

B. Finite-State Machine:

The most common character behavior management models are the ones based on a finite-state machine. The finite-state machine is a model used to represent and control execution flow and these models are appropriate for NPCs with a few possible states. An NPC that patrols some area is a good example of such a model. However, the bigger the number of possible states the more complicated the finite-state machine becomes, thus increasing the implementation time. Furthermore, these machines are strongly tied to the generated circumstances and it is hard to use a single one in multiple projects simultaneously. Consequently, the majority of modern computer games don't utilize finite-state machines in NPC development. Although some of them use improved versions of the machines.

C. Behavior Tree:

Behavior trees dominate amongst the technologies for AI creation in computer games. Behavior Trees make it possible to control characters in computer games and to define hierarchies of decisions and actions. Behavior trees are also crucial to implement believable agents and to control groups of characters in an interactive way. A vast variety of nodes that differs byways of management of thread of execution provides a high potential of creation of complex behavior structures. Behavior trees are better than finite-state machines (FSM) due to the lack of problems in character state changes. Moreover, this approach allows the creation of more complex behavior structures and to add new behavior patterns with ease. Behavior trees also have several disadvantages. The Depth-first search (DFS) takes a lot of time to complete in a big tree when the search occurs every frame. Behavior trees can't fully implement a believable character behavior due to the lack of an ability to make random decisions. Complex behavior is described with a huge tree that makes it hard to navigate and debug. There are some extensions to behavior trees that can include parallel computation improvements or integration of deep learning methods. But the game industry tends to use more modern approaches for AI creation.

D. Machine Learning Methods:

Machine learning is a subset of artificial intelligence that focuses on using algorithms and statistical models to make machines act without specific programming. Due to the increase in computing potential, the Artificial Neural Network (ANN) becomes more popular in computer games in the last several years. The main usage of the ANN is to provide a player with a proper challenge. An OpenAI bot for the game DOTA 2 was developed with the usage of neural networks in 2017. Another example is an ANN called "DeepMind" for the game StarCraft 2. In general cases, machine learning methods in AI classifies action parameters by input data depending on a network type and architecture. In most cases, these networks learn in offline mode when a game is in development due to high computational resource requirements and unpredictable results. Reinforcement learning has gained special attention in AI researches. This is like learning without a teacher where a system compares computed results with expected ones on its own and changes weights between them depending on similarities in both results. Such an approach allows ignoring the lack of an appropriate mapped database. However, a strict definition of a network error approximation function is needed for the methods to work. Sometimes it is a hard task to define such a function.

E. Utility AI

Utility AI is a technique based on estimating the profit of executing actions. They provide a flexible and scalable approach for generating NPC decision-making systems. When it becomes necessary to choose between different options, an NPC calculates the benefit (utility) of each option and then goes for the option with the highest utilities. The process of estimating benefits considers not only character parameters, but also the assessment of a game environment state. The estimation of benefits brings the decision-making process to a level of fuzzy logic, and that leads to the ability of agents to make believable decisions even in a game situation that was not anticipated during the design. One of the main benefits of the utility approach is the way of presenting behavior rules. They may be described by a curve representation, that shows the superposition of multiple behavior rules, or by a table representation that provides the general overview on changing of utility value in different game situations. The advantage of the Utility AI is that the behavior developed with it becomes highly reusable and expandable.

IV. CONCLUSION

In this paper, we discussed a set of challenges that state the-art of computer games poses to the artificial intelligence community. Developing AI techniques that can deal with the different situations in computer games can be a bit complex and challenging, but has a great potential to have a great impact on other areas including education, training, and entertainment. The decision-making in this paper provides a way to create complex behavior rules for the game characters

controlled by artificial intelligence. With the help of these rules, it becomes possible to generate characters with believable behavior that meets modern requirements for a game of artificial intelligence. To achieve believability, a combination of the Behavior roles and Behavior characteristics components is used. They allow characters to bring more uniqueness into their decisions and adapt their behavior according to the state of a game environment. In turn, generating unique sets of Behavior Characteristics solves the problem of creating a large amount of agents with different and realistic behavior to inhabit huge game worlds.

ACKNOWLEDGMENT

I wish to express my wholehearted appreciation and a deep sense of heartfelt gratitude to my guide Prof. Swapna Nikale, Department of Information Technology, B.K. Birla College of Arts, Science & Commerce for her generous help, excellent guidance, lucid suggestions, and encouragement throughout this work. Her concrete directions and critical views have impactfully helped me in the successful completion of this work. I am also thankful to those who have directly or indirectly helped.

REFERENCES

[1] Petrovic, V. M. (2018a). Artificial Intelligence and Virtual Worlds – Toward Human-Level AI Agents. IEEE Access, 6, 39976–39988. https://doi.org/10.1109/access.2018.2855970
[2] Martinez-Arellano, G., Cant, R., & Woods, D. (2017). Creating AI Characters for Fighting Games Using Genetic Programming. IEEE Transactions on Computational Intelligence and AI in Games, 9(4), 423–434. https://doi.org/10.1109/tciaig.2016.2642158
[3] Sekhavat, Y. A. (2017). Behavior Trees for Computer Games. International Journal on Artificial Intelligence Tools, 26(02), 1730001. https://doi.org/10.1142/s0218213017300010
[4] J. Togelius and G. N. Yannakakis, "General general game AI," 2016 IEEE Conference on Computational Intelligence and Games (CIG), Santorini, 2016, pp. 1-8, doi:10.1109/CIG.2016.7860385.
[5] U. A. S. Iskandar, N. M. Diah and M. Ismail, "Identifying Artificial Intelligence Pathfinding Algorithms for Platformer Games," 2020 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), Shah Alam, Selangor, Malaysia, 2020, pp. 74-80, doi:10.1109/I2CACIS49202.2020.9140177.

