International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2019 IJSRCSEIT | Volume 5 | Issue 2 | ISSN : 2456-3307
DOI : https://doi.org/10.32628/CSEIT1952177

Local Ride

Raparthi Aruna*, Tammisetty Praneetha, Potti Sai Sandhya
Computer Science and Engineering, Vasireddy Venkatadri Institute of Technology, Guntur, Andhra Pradesh, India

ABSTRACT

Games plays an important role, as we go through life we succumb to both physical and mental decline. To stave off mental decay one must maintain an active brain. Playing brain games or video games (as long as they're not entirely mindless) all help to curb the loss and also increase your problem solving skills. At present, Artificial Intelligence has been an integral part in developing games and expanded greatly since its introduction. Experiencing AI in gaming will be quite exciting. Our application, Local Ride is a game in which the player can change the vehicle of his interest. The game uses AI for creating AI humans, AI cars, AI traffic signals, AI buildings. The player tries to complete the target with the vehicle moving on the given track. The player can experience the traffic which looks similar in the real environment. The game loads by selecting the vehicle and is completed when the player finishes the target within a given time limit.

Keywords : Artificial Intelligence, digital immigrants, electronic games, MonoDevelop, Unity

I. INTRODUCTION

Games have become an integral part of everyday life for many people. A traditional game often presents a situation where "players engage in an artificial conflict, defined by rules and results in a quantifiable outcome". Such artificial conflicts are often represented as a puzzle or a challenge, and having the puzzle solved or the challenge resolved provides a real world purpose to the game players. This type of games is sometimes referred to as "serious" games. However, this kind of traditional or "serious games" has been increasingly replaced by electronic games, especially for the so-called game generation. This generation typically consists of "digital natives," who, in contrast to "digital immigrants" of the older generation, grew up playing a lot of games and who are trained in skills such as "dealing with large amounts of information quickly even at the early ages, using alternative ways to get information, and finding solutions to their own problems through new communication paths".The artificial intelligence(AI) community has witnessed a similar transition from the "classical AI games" such as Samuel's Checker Player and Waterman's Poker Player to the contemporary AI techniques adopted in electronic games. The objective of the game is no longer a quantifiable outcome of beating the opponent in a checker or poker game.

Instead, a contemporary game contains changing environments, multiple objectives, and dynamic aspects of the game that are revealed to the game player as the game unfolds. The objective is to offer the human player an enjoyable experience through his or her interaction with the game, and this does not involve any specific quantifiable outcome. The game is a contemporary video game and its objective is to offer the player a challenging and enjoyable experience in a car race against a game-controlled car.

Although the player's goal within the game is to reach the target. The AI techniques adopted in the game are primarily designed to give the player an enjoyable time racing his or her car.

II. METHODS AND MATERIAL

1. UNITY GAME ENGINE

The game's 3D environment was built using the Unity game engine developed by Unity Technologies. Unity is a game development engine combined with an integrated development environment (IDE) that is capable of creating applications for Windows, Mac OS X, Linux, and other mobile platforms .

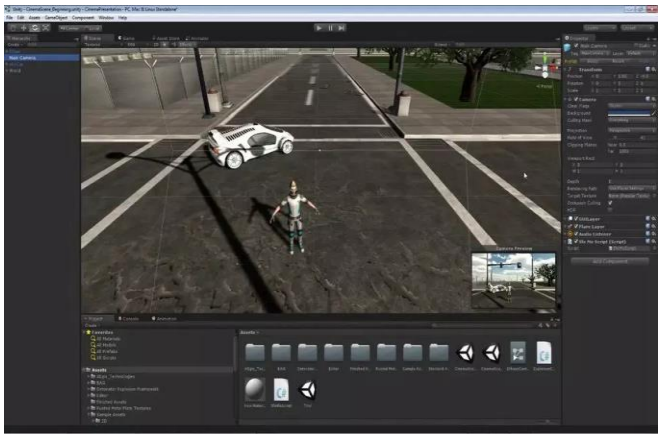

Fig 1: Unity game development

The Unity game engine can perform graphical rendering and physics calculations, thereby enabling game developers to focus on the creation and development of the game software. Using the Unity engine, developers can position, scale, and add realistic physics to objects in the game system. The below figure shows the Unity editor's interface. The left panel shows a list of defined game objects, the middle visualization shows the game objects in the 3D space of the racetrack, the right panel shows the attributes of a selected game object, and the bottom panel consists of icons of the scripts, each of which defines the behavior of a game object.

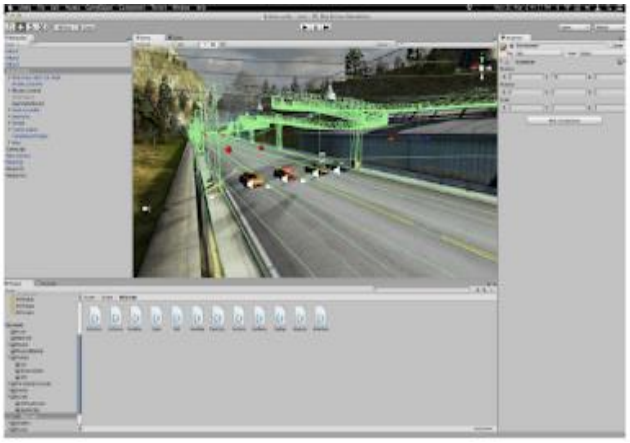

Fig 2 : Chasing the vehicle image

ENVIRONMENT MODULE

The environment module consists of two submodules: (i) the racetrack and (ii) the waypoints system. The racetrack is the road within the game environment on which the vehicle moves, and the waypoints system represents the key positions around the racetrack.The latter enables the game i.e.,game-controlled cars to navigate around the racetrack. The Unity platform provides built-in components that can satisfy these requirements of a smooth path along the track and its fast determination.

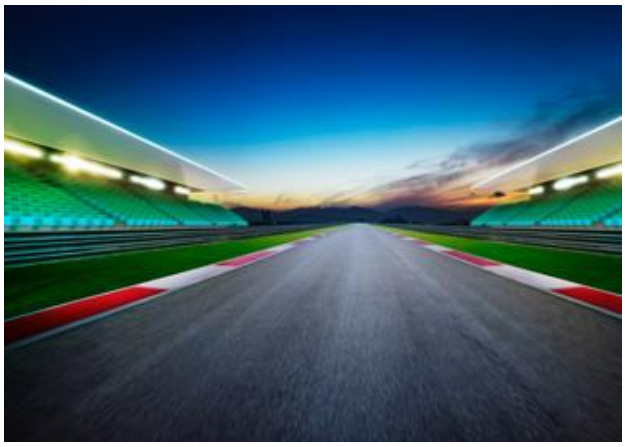

Fig 3 : Race track image

2.MONO DEVELOP

The Unity game engine works in conjunction with Mono Develop for controlling the behavior of objects.

MonoDevelop is an open source Integrated Development Environment or IDE developed by Xamarin and the Mono community, which is primarily used for development in the C# programming language. With MonoDevelop, developers are able to write scripts in C#, JavaScript, or Boo and attach them to objects in the Unity engine; these scripts enable developers to control the logic and behaviors of objects within the Unity environment. In this way, the combined tools of Unity and MonoDevelop enable developers to focus on the development of the AI components of the game rather than on issues related to 3D graphics rendering and physics calculations. The script inside one of the icons shown on the bottom panel of the Unity editor; development of the script was supported by the MonoDevelop IDE.

PROGRAMMING

The primary programming language used for writing the scripts to control the behaviors of objects was C#. C# was chosen because it is object-oriented and class-based. Also, it has a variety of libraries and frameworks that are built specifically for the development with Unity, which contain many generic classes and interfaces such as the List, Stack, and Dictionary. These built-in classes have been optimized and are accessible with online documentation.

III. RESULTS AND DISCUSSION

Development of the game system was made easier because of the implementation tools. The Unity game engine supports effective development of the game system with its high-level abstraction programming tools and intuitive user interface. These features that support the developers in the implementation of AI concepts so that they can focus on the game logic and ignore lower level development details such as graphics rendering and physics calculations. The combined tools of MonoDevelop and Unity support the implementation processes because MonoDevelop consists of auto correction features for many libraries and SDKs used in Unity.

The Unity platform provides built-in components that can satisfy these requirements of a smooth path along the track and its fast determination. Other features of Unity also support the development effort.

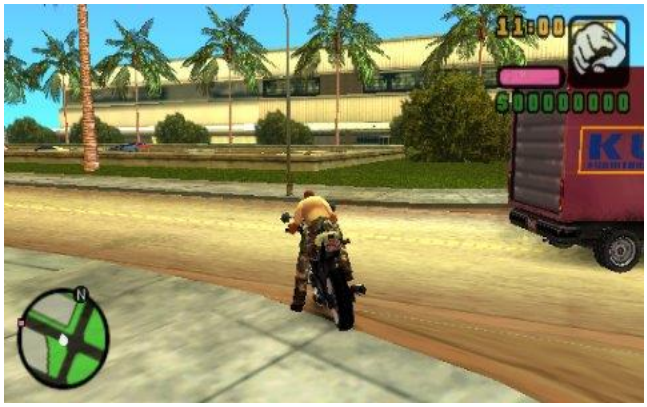

Fig 4 : Player with vehicle

The game play contains a player who can change his vehicle on his interest and completes the game by moving to the target within a specific time that is provided.The player can shift to any of the vehicle on the road track.

IV. CONCLUSION

Racer was tested by who all reported that they thoroughly enjoyed the game. All of which are not available if the implementation was done using traditional AI search techniques. With these Unity components, the developer was able to implement the vehicle on the track with less effort and more efficiently, and the vehicle can successfully mimic human driving behavior. The player can experience the traffic which looks similar in the real

environment. The game loads by selecting the vehicle and is completed when the player finishes the target within a given time limit.The game is developed using the unity engine with which it can be modified again to make any modifications to the game.

V. REFERENCES

[1]. K. Salen and E. Zimmerman, Rules of Play: Game Design Fundamentals, vol. 1, MIT Press, 2004.

[2]. Y. Liu, T. Alexandrova, and T. Nakajima, "Gamifying Intelligent Environments," http://www.dcl.cs.waseda.ac.jp/~yefeng/yefeng / pubs/2011/ubimui11 yefeng.pdf.

[3]. M. Prensky, "Digital natives, digital immigrants part 1," On the Horizon, vol. 9, no. 5, pp. 1–6, 2001.

[4]. A. L. Samuel, "Some studies in machine learning using the game of checkers. II—recent progress," IBM Journal of Research and Development, vol. 11, no. 6, pp. 601–617, 1967.

[5]. A.Khoo and R. Zubek, "Applying inexpensive AI techniques to computer games," IEEE Intelligent Systems, vol. 17, no. 4, pp. 48– 53, 2002.

[6]. Unity—GameEngine, http://unity3d.com.

[7]. Microsoft Developer Network, http://msdn.microsoft.com/enUS.

[8]. https://unity3d.com/get-unity/download.

Cite this article as :

Raparthi Aruna, Tammisetty Praneetha, Potti Sai Sandhya, "Local Ride", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 5 Issue 2, pp. 796-799, March-April 2019. Available at doi : https://doi.org/10.32628/CSEIT1952177

Journal URL : http://ijsrcseit.com/CSEIT1952177