Abstract

Artificial Intelligence (AI) in games has taken the backseat in development for a long time for many reasons but the future of game is definitely going to be weighted heavily with increasingly detailed game AI. If your game’s AI is not up to the current level that game player’s expectations demand then your game will feel dated and suffer for it in their opinions [3].

Game AI is not just neural networks, learning systems and complex mathematical structures, although it can be, about creating an environment and the appearance of thought from units. Game AI is behavioral, not scientific [3].

The key to understanding how to create game AI is understanding what you want your final results to be and then building the system to provide those results. It all comes down to what the player can see; if they can’t tell it’s happening, then it might as well not be.

Since game AI is centered on appearance of intelligence and good gameplay, its approach is very different from that of traditional AI. A common goal today is to make the game AI more human, or at least appear so [3].

Pathfinding is one of the important factor in game AI. Pathfinding in the games involves solving a planning problem with agents seeking optimal paths from a start state to a goal state. The pathfinding process involves utilizing the full state space information available to agents to find the least expensive route to the goal.

Two well-known and efficient existing pathfinding algorithms for 3D games and simulation projects are Breadth-First Search (BFS) and Dijkstra Algorithm. Among them Dijkstra is one of the most famous path finding algorithm being used in 3d Games and Simulation project now a days.
By doing literature survey, following limitations in 3D games development have been identified with existing pathfinding algorithms [5]:

1. Breadth First Search (BFS) is a non-informed search for trees. It involves visiting the nodes closest to the root before spreading out [11,12].
2. BFS uses memory like a hog. Since we want to visit the nodes closest to the root before we visit their children, we must store the children in a queue. At each iteration, we take out a child, check to see if she is the solution. If she is, exit quickly. If not, put her children in the queue and repeat [11,12,13,14,15,16].
3. BFS must put all the children of all the nodes it traverses, the queue can get quite big, especially if the branching factor is high [11,12,13,14,15,16].
4. The major disadvantage of the existing pathfinding algorithm like Dijkstra is the fact that it does a blind search there by consuming a lot of time waste of necessary resources.
5. Another disadvantage is that Dijkstra cannot handle negative edges. This leads to acyclic graphs and most often cannot obtain the right shortest path in the game [4,11,12,13,14,15,16].

For a more efficient and safer design of public facilities, it is important to be able to estimate the behavior of the crowd. The existing model available for the game to describe the steer behavior of the crowd usually deals with microscopic variables like the average speed or the flow. They are not designed on an individual-based model of the crowd which results in not refined steer behavior in the game system[4,11,12]. Also, there is need of the model which presents classification of the factors that influences the performance of path finding techniques [4,11].
As a part of research, a new pathfinding algorithm for 3D game has been designed and developed to overcome the limitations of existing pathfinding algorithms (BFS and Dijkstra). Using this algorithm a prototype model of AI game (third person shooter 3D game) is designed and developed for steer behavior.

**Features of the Prototype model of AI game System (AI Demo):**

1. The AiDemo game uses proposed new pathfinding algorithm which is faster and more accurate pathfinding than existing once (BFS and Dijkstra) in Game Artificial Intelligent world. Proposed pathFinding technique for Steer Behavior combines the pieces of information that Dijkstra’s algorithm uses (favoring vertices that are close to the starting point) and information that BFS uses (favoring vertices that are close to the goal). This research will focus on how machine learning techniques like BFS and Dijkstra Algorithms can be used to enhance agents ability to find a path in real-time.

2. The research also includes classification of the factors that influences the performance of path finding techniques. This factors includes the dynamics of the game, the geometry of the players and the environment, the (un) predictability of movement, kinematic and temporal restrictions, interaction rules, and real-time performance. The purpose of this classification is to help the developers to identify the complexity of the task before choosing a certain approach.

3. A focus of pedestrian’s behavior in 3D games and simulations is increased now a days in the game industry. AI Demo also contains efficient collision avoidance algorithm that can be used in simulation of crowd behavior in 3d game and simulation project. It demonstrates flock AI, Combat system AI, NPC behavior, and Crowd behavior.
4. The demo is an ACTION game which is developed using Unity3D game engine. It is supported by wide range of platforms, since the Unity3D game engine supports many platforms, i.e. a player with an Android device, iphone with windows desktop PC can run this game by doing minor modification in the code.

Hence, the research emphasis on steer behavior and implementation of it using newly designed and developed pathfinding algorithm.

The research work is divided into six chapters.

This first chapter: Introduction covers, overview about the AI, application areas of AI and technology used in the 3D game development. It gives overview of existing pathfinding techniques with their limitations. It also gives idea about needs of research work in this area. In the later part of the chapter objectives, features of the proposed model are listed. The last part of the chapter gives overview of the structure of this thesis.

The second chapter: is Literature survey and Scope of Research, gives introduction of graph theory and search algorithm like DFS, BFS and Dijkstra for 3D pathfinding has been discussed. In the middle part of chapter, types of games and game platforms have been discussed. Later part of this chapter discusses steps for game development and limitation of existing pathfinding algorithms. In the last scope of research in this area has been identified.

The third chapter: Study of related technology, gives overview of the study of the technologies used to design and develop the Artificial Intelligent Game System for Steer behavior. It gives introduction of windows 7 operating system, Visual Studio 2008, .NET framework, C#, 3D Max, Windows API, Photoshop CS, Unity 3D engine and JAVA Script. The main purpose of this chapter is to introduce the reader of this document to the above mentioned technologies.

The forth chapter: Overall System, describes various factors that influences the complexity, and hence the performance of the pathfinding. These factors
have been categorized according to issues such as the type of the path, the soundness for planning, the dynamics of the game, the geometry of the environment and the players, the uncertainty of information and various other factors. It also describes new developed faster, accurate Path finding algorithm in Artificial Intelligent world. In later stage it explains concept of steer behavior.

The **fifth** chapter: **System Outcome**, explains in detail about the developed gamed AI system. It demonstrates combat AI and NPCs. For each selected item, it shows screenshots and some important coding. Code shows how the newly pathfinding algorithm is used for steer behavior.

The **Sixth** chapter: **Conclusion and Future scope**, concludes the research work done and explains features of newly designed and developed pathfinding algorithm. In middle of the chapter it discusses limitations of existing pathfinding algorithm, Dijkstra and it shows comparison of this algorithm with it. In later part it describes future scope of this new algorithm. It is observed that the new algorithm is more efficient than Dijkstra algorithm.