Conclusion & Future Scope

6.1 Introduction
6.2 Conclusion
6.3 Comparison
6.4 Application
6.5 Performance
6.6 Limitations
6.7 Further Scope
6.8 Summary
Chapter 6: Conclusion & Future Scope

Chapter - 6
Conclusion & Future Scope

6.1 Introduction
This part of the chapter describes conclusion about newly designed and developed pathfinding algorithm and implementation of steer behavior which uses this algorithm. In middle it describes limitation and comparison of new algorithm with existing pathfinding algorithm Dijkstra. In later part it describes performance, limitation and future scope of this algorithm.

6.2 Conclusion
As a part of research work the new pathfinding algorithm has been designed and implemented. Using this algorithm AI Demo game has been created.

- The AiDemo game uses proposed pathfinding algorithm which is faster and more accurate than existing ones (BFS and Dijkstra) in Game Artificial Intelligent world. This newly designed 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 an agent's ability to handle pathfinding in real-time.

- A focus of pedestrians behavior in 3D games and simulations is increased now a days in the game. AI game demo which is made for steer behavior, implementation of it also contains existing efficient collision avoidance algorithm that can be used in
simulation of crowd behavior in 3D games and simulation project. It demonstrates flock AI, Combat system AI, NPC behavior, and Crowd behavior. Hence, the research emphasis on steer behavior and implementation of it using proposed pathfinding algorithm.

- As a part of the research, the classification of the factors are also made 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.

- The game demo developed for this research with proposed pathfinding (seeking for a way from one point to another) algorithm and Steer Behavior is an ACTION game which is developed using Unity3D game engine. Due to the wide range of platforms the Unity3D game engine supports, the developed model also supports cross-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.

- The newly designed and developed algorithm is faster because it use heuristics to make more educated guesses about which route is the best to case, something which Dijkstra's algorithm does not do.
The novel algorithm which is nothing but the extended form of Dijkstra’s algorithm and works exactly the same except it adds an approximate distance to the end node as a part of the weight system. This approximation is done by heuristic (a method where trading optimality, accuracy or completeness for speed when the tradition problem solving methods do not work). Use of this method allows the algorithm to eliminate longer paths based off this approximation, in turn speeding up the resolution of the shortest path. Use of this heuristic approach makes this algorithm faster than the Dijkstra’s algorithm.

The novel algorithm not only intends to take shortest step among each movement, but also cares about the choosing step whether on the direction which is just from source to target.

6.3 Comparison

Let us compare newly designed pathfinding algorithm with well-known pathfinding algorithm Dijkstra.

1. Novel pathfinding search uses heuristic. This heuristic helps to reach nearer. Efficiency of this algorithm depends on heuristic function, which can be either complete or optimal. This algorithm is faster than Dijkstra. Dijkstra is helping to find a shortest path.

2. By comparing figure 6.1 and 6.2 we can say that Dijkstra scans all nodes available on the terrain. More over it also visits some node again during the search operation, where as the new algorithm requires less node to scan and it never visits the same node during the search operation, Hence the new algorithm is more efficient and faster.
Figure 6.1: Execution of Dijkstra algorithm scans all nodes
3. Dijkstra has one cost functions for reaching to each node from the source. So $F(x) = G(x)$.

Where as in case of this new algorithm, it has two functions. One is the same as Dijkstra ($G(x)$) for reaching to node $x$ and second one is, heuristic function $H(x)$. Here, we have to note that $H(x)$ should never overestimate the cost, which means real cost to reach goal node from node $x$ should be always greater than or equal to heuristic function. i. e. the $F(x) = H(x) + G(x)$.
Chapter 6: Conclusion & Future Scope

6.4 Application
This newly designed and developed algorithm which is improved form of Dijkstra algorithm, examines fewer node for shortest path than Dijkstra. For games specially for third person shooter genre this algorithm improves frame rate than Dijkstra pathfinding algorithm.

6.5 Performance
As already stated, the creation of the field only took less than 10% of CPU time. Important to know is that this was achieved with every cell storing the references to its neighbors.

The novel algorithm is superior to tradition Dijkstra in time efficiency.

6.6 Limitations
The limitation of this algorithm is that it examines a fewer path than Dijkstra but it is not always guaranteed that it is a optimal path. For games this compromise is not an issue as far as it improves frame rate than Dijkstra algorithm. Where ever guaranteed shortest path is needed for each and every time than this algorithm can’t be used.

6.7 Further Scope
- To overcome limitations of this new algorithm, more efficient Heuristic function can be developed.
- This newly designed algorithm can be modified so that it always returns the shortest path.
Chapter 6 : Conclusion & Future Scope

6.8 Summary

This chapter concludes the research work done and explains features of newly designed and developed pathfinding algorithm. In middle of the chapter it has shown limitation, comparison with existing pathfinding algorithm Dijkstra. In later part it describes limitation and future scope of this algorithm. Also it describes limitations of this algorithm and makes it comparison with existing pathfinding algorithm Dijkstra. It is observed that it is an efficient than Dijkstra algorithm.