Pathfinding in partially explored games environments

The application of the A* Algorithm with Occupancy Grids in Unity3D

John Stamford, Arjab Singh Khuman, Jenny Carter & Samad Ahmadi
Centre for Computational Intelligence
School of Computing
De Montfort University
Leicester, United Kingdom

***Abstract*—One of the key aspects of games development is making Non-Playable Characters (NPC) behave more realistically in the environment. One of the main challenges is creating an NPC that is aware of its surroundings and acts accordingly. The A* Algorithm is widely used in the games development community to allow AI based characters to move around the environment however unlike real characters they are often given information about the environment without having to explore. This paper combines the use of the A* Algorithm with the occupancy grid technique to allow Non-Playable Characters to build their own representation of the environment and plan paths based on this information. The paper demonstrates the application of the approach and shows a range of testing and its limitations.**

Keywords*—A* Algorithm, AI, Games Development, Mobile Robots, Occupancy Grids, Unity3D

I. Introduction

The paper aims to discuss the application of the A* Algorithm [1] in partially explored environments with the aim of making pathfinding for NPCs in a 3D games engine. The proposed pathfinding system is aimed to simulate more realistic planning and decision making processes based on the information known about the environment. This will result in the proposed system making assumptions about the best path and then having to recalculate the path as it learns more about the environment.

One of the distinct differences between a human player and a NPC has is that often the NPC is given information about the whole environment prior to solving the pathfinding process [2]. This gives the NPC an unnatural awareness of its environment which can result in an unrealistic behaviour when navigating. The proposed system aims to address this issue as the NPC has to plan the path to a given target through an fully unknown environment. As the NPC moves along the path it builds an internal map of the environment and recalculates the best path based on this internal map. The proposed system will build on the work of [3] which discusses the application of occupancy grids in robotic systems. As a result the NPC will store information about the environment and will apply pathfinding to its internal knowledge rather than the environment.

The system was developed in Unity3D and simulates real world interaction through the application of its physics engine. The NPC simulates sensors such as Sonar and LIDAR through the use of a Raycast function. The NPCs destination is set through the use of a placeable target object. The environment can be constructed from any objects which have a 3D collider that allows the Raycast function to interact with.

978-1-4799-5538-1/14/$31.00

II. Literature Review

A. Pathfinding algorithms in games

The A* algorithm is widely used in game development as a way of controlling NPC movement [2][4][5][6]. The A* algorithm is similar to that of Dijkstra's algorithm [7] however it uses a heuristic to determine the optimal search area. The most common implementation of the A* algorithm uses the Manhattan approach where the heuristic is calculated by the number of nodes from the current position to the target. The use of an appropriate heuristic allows the algorithm to find the shortest path to the target and has a distinct advantage over the Dijkstra algorithm.

Typically pathfinding in games is achieved through the use of either predefined nodes [8] or the use of a tile based system [2]. When using a node based system the developer is usually required to position path nodes throughout the environment as shown in Fig. 1. The limitation of a node based approach is that the behaviour of the NPC is predictable and that in some cases the NPC will not have full access to the environment. When using a tile based approach as shown in Fig. 2 the NPC is often given information about the whole environment, again making the behaviour unnatural.

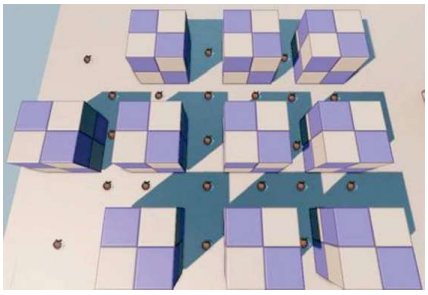

Fig. 1. Node based methods [8]

In addition to the A* algorithm further pathfinding approaches include various D* alternatives such as [9][10] and further work such as [1][11] which are more widely used in mobile robotics.