Hierarchical Path-Finding for Navigation Meshes (HNA*)

Nuria Pelechano, Carlos Fuentes
Universitat Politècnica de Catalunya

Abstract

Path-finding can become an important bottleneck as both the size of the virtual environments and the number of agents navigating them increase. It is important to develop techniques that can be efficiently applied to any environment independently of its abstract representation. In this paper we present a hierarchical NavMesh representation to speed up path-finding. Hierarchical path-finding (HPA*) has been successfully applied to regular grids, but there is a need to extend the benefits of this method to polygonal navigation meshes. As opposed to regular grids, navigation meshes offer representations with higher accuracy regarding the underlying geometry, while containing a smaller number of cells. Therefore, we present a bottom-up method to create a hierarchical representation based on a multilevel k-way partitioning algorithm (MLKP), annotated with sub-paths that can be accessed online by our Hierarchical NavMesh Path-finding algorithm (HNA*). The algorithm benefits from searching in graphs with a much smaller number of cells, thus performing up to 7.7 times faster than traditional A* over the initial NavMesh. We present results of HNA* over a variety of scenarios and discuss the benefits of the algorithm together with areas for improvement.

Keywords: path-finding, hierarchical representations, navigation meshes

1. Introduction

Most video games are required to simulate thousands or millions of agents who interact and navigate in a 3D world and show capabilities such as chasing, seeking or intercepting other agents. Path-finding provides characters with the ability to navigate autonomously in a virtual environment. The most well known path-finding algorithm is A*, which explores the nodes of a graph while balancing the accumulated cost with a heuristic to find an optimal path quickly. Throughout the years many algorithms have been proposed to further speed up the basic A* algorithm, but the cost of these algorithms is still strongly dependent on the size of the graph. Hierarchical path-finding aims to reduce the number of nodes that need to be explored when computing paths in large terrains. The reduction in the number of nodes for higher levels of the hierarchy significantly decreases the execution time and memory footprint when calculating paths.

Current hierarchical techniques may result in unbalanced abstractions. For example, top-down hierarchies are created by splitting the environment into large square clusters, where all the clusters contain the exact same number of lower level grid cells. The main disadvantages of such constructions are that the resulting higher level of the hierarchy may have an uneven number of edges between nodes and also an uneven number of walkable cells (since there may be some clusters with a large percentage of the grid cells being occupied by obstacles).

Navigation meshes represented by polygons provide closer representation of the geometry with a lower number of cells than regular grids. Since having a smaller number of cells can greatly accelerate path-finding, it is therefore necessary to extend the concept of hierarchical path-finding to a more general representation of navigation meshes with polygon based cells. Moreover it would also be beneficial to have a hierarchical representation with a balanced number of polygons per node and portals between nodes.

In this paper we present a new hierarchical path-finding solution for large 3D environments represented with polygonal navigation meshes. The presented solution works with navigation meshes where cells are convex polygons, and thus it also includes triangular representations. Our hierarchical graph representation is based on a multilevel k-way partitioning algorithm annotated with sub-path information. Our method presents a flexible approach in terms of both the number of levels used in the hierarchy and the number of polygons to merge between levels of the hierarchy. We evaluate the gains in performance when using our hierarchical path-finding, and discuss the trade-offs between the number of merged polygons and the number of levels employed for the search. We present a number of benchmarks that can help during the parameter fitting process to achieve the best speedups, as well as a quantitative analysis of the bounds on sub-optimality of the paths found with HNA*.

We also present an evaluation of the bottleneck that appears for certain configurations when inserting the start and goal positions in the hierarchical representation.

2. Related Work

A large amount of work to speed up path-finding focuses on enhancing the A* algorithm to reduce the computational time needed to calculate a path. This comes at the cost of finding sub-optimal paths or allowing a certain degree of error when searching for the optimal path and then allows the algorithm to