Robust Navigation Meshes from Polygonal Obstacles

Computer Graphics Lab

Marcelo Kallmann

University of California, Merced
http://graphics.ucmerced.edu
mkallmann@ucmerced.edu

i3D 2007 Symposium on Interactive 3D Graphics and Games
Posters & Demos

Introduction

Navigation meshes are commonly used for representing the free space characters are allowed to navigate in computer games. This work proposes the use of **Dynamic Constrained Delaunay Triangulations** [1] for robustly constructing navigation meshes from any set of given polygonal obstacles.

Implementation

The presented implementation of Dynamic Constrained Delaunay Triangulations has unique features for robustly handling polygonal obstacles:

(1) It automatically detects and handles polygonal obstacles that overlap, intersect with each other, and that have self-intersections or duplicated points, always generating a correctly triangulated navigation mesh as output.

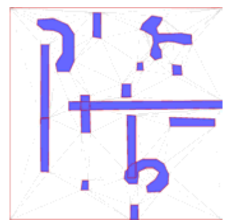
(2) Each polygonal obstacle receives a unique *id* that can be later on used for obstacle removal from the triangulation with local operations. Therefore obstacle displacement can be achieved by successive removals and insertions in the triangulation.

Furthermore, efficient algorithms for path planning in the maintained triangulation can be employed. The size of the adjacency graph used for searching paths is often much smaller than in grid-based methods, and the triangulated domain precisely describes polygonal regions (no approximation as in grids). Extensive tests have demonstrated that path planning in triangulations greatly outperforms grid-based methods [2].

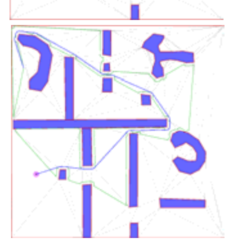
Availability

The software has been integrated in the Graphsim toolkit of the UCM Graphics Lab, and it is now **freely available for noncommercial use**.

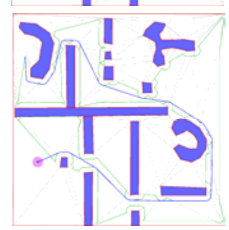
Examples



A navigation mesh is obtained by computing the Constrained Delaunay Triangulation of given polygonal obstacles. Only intersections between obstacles are added as additional vertices to the triangulation.



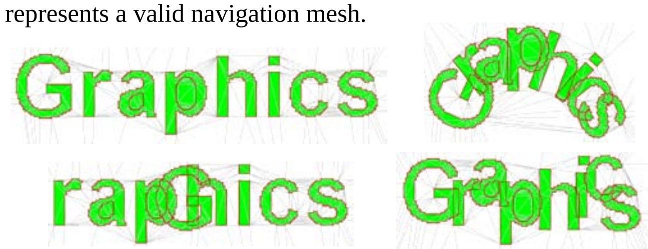
Modifications and rearrangements of obstacles are dynamically handled, for example to form walls and rooms. While the triangulation is being maintained, collision-free paths can be computed at anytime.



Paths can also be computed according to the agent size*. Here the obtained path shown is different in order to accommodate an agent of larger size. If agents have equal sizes, obstacles can be dilated before triangulation.

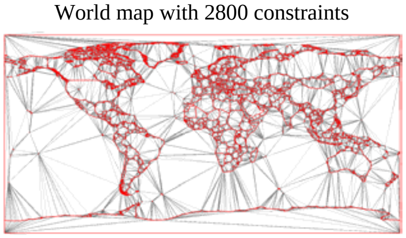
*under development

The examples below demonstrate the result of dynamically displacing several polygonal obstacles (or constraints). The triangulation is correctly managed and always represents a valid navigation mesh.

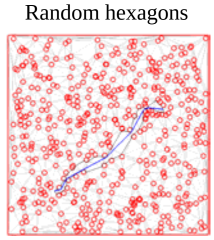


Further Examples

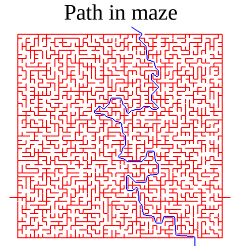
World map with 2800 constraints



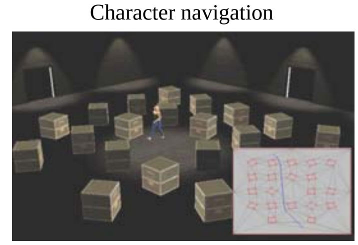
Random hexagons



Path in maze



Character navigation



References

[1] M. Kallmann, H. Bieri, and D. Thalmann, "Fully Dynamic Constrained Delaunay Triangulations", In Geometric Modeling for Scientific Visualization, G. Brunnett, B. Hamann, H. Mueller, L. Linsen (Eds.), Springer-Verlag, Heidelberg, Germany, pp. 241-257, 2003.

[2] D. Demyen and M. Buro, "Efficient Triangulation-Based Pathfinding", Proceedings of the AAAI conference, Boston 2006, pp.942-947.