

# 基于 A\* 算法的游戏寻径的设计与实现

王豫峰<sup>1</sup>, 韩璞<sup>1</sup>, 王华彬<sup>2</sup>

(1.南阳理工学院 软件学院,河南 南阳 473000;2. 武汉天和技术股份有限公司,湖北 武汉 430000)

**摘要:**在游戏中,寻径算法占有很重要的作用,A\* 算法广泛应用于游戏寻径,是目前比较流行的启发式搜索算法之一。该文首先介绍 A\* 算法的思想,并通过矩形方格模拟游戏地图,应用 A\* 算法实现最优路径的搜索。

**关键词:**A\* 算法;地图寻径;启发式搜索

中图分类号:TP313 文献标识码:A 文章编号:1009-3044(2011)30-7450-02

## Design and Implementation of Game Map Path Finding Based on A\* Algorithm

WANG Yu-feng<sup>1</sup>, HAN Pu<sup>1</sup>, WANG Hua-bin<sup>2</sup>

(1.Nanyang Institute of Technology, Nanyang 473000, China; 2.Wuhan Tianhe Technology Co., Ltd., Wuhan 430000, China)

**Abstract:**Path-finding algorithm is the most important issues in the developing of game. A\*Algorithm is one of the most popular Heuristic searching,that is the most widely used algorithm in the field of game map path-finding. This paper analyzes the principle of A\* algorithms and realize the best path-finding by using rectangular grids to simulate the game map.

**Key words:**A\* algorithm; map path-finding; heuristic searching

在游戏开发中,智能寻径是非常重要的一个模块。智能寻径就是角色在当前游戏场景中,能够绕过不同的障碍,以最快的方式或者代价最低的方式到达目的地。目前,路径搜索方法按照搜索方式来划分,可以分为盲目搜索和启发式搜索两种。启发式搜索算法在进行搜索的时候,对每一个搜索的节点进行评估,得到一个最佳的节点。然后,再从这个节点进行搜索直到搜索到目标节点[1]。

启发式搜索可以省略大量无谓的搜索路径,极大提高搜索效率,在一些复杂问题的解决中得到了广泛应用。A\* 算法是目前比较流行的启发式搜索算法之一,被广泛应用于路径优化领域。

## 1 A\* 算法原理及算法思路

A\* 算法是启发式搜索算法,对于搜索空间中需要查找的节点,通过一个启发函数对当前节点与目标节点之间的路径情况进行评估。由于搜索空间的不确定性,所以评估函数只是一个相对来说比较合适的评估函数。不同的评估函数将会对 A\* 算法的效率产生不同的影响。

A\* 算法的评估函数可表示为: $f(n)=g(n)+h(n)$

$g(n)$ 函数表示从起始节点到节点  $n$  所需要的时间;

$h(n)$ 函数表示从节点  $n$  到目标节点所需要的估计时间;

$f(n)$ 函数表示从起始节点路过节点  $n$  到达目标节点的最合适路径的时间;

通过比较  $f(n)$ 值的大小,可以所需时间最少的节点作为下一路径的节点<sup>[2]</sup>。

在 A\* 算法中用到了两个链表,一个是 Open 表,用来存放所有搜索的节点。另一个是 Closed 表,用来存放所有已经检索到最合适的节点。

算法过程如下:

1)创建 Open 表,初始化时将起始节点放入 Open 表中,此节点  $f(n)=h(n)$ ,因为该节点为起始节点。

2)创建 Closed 表,初始化清空;

3)寻找起点周围所有可达到的节点,跳过有障碍的节点,把他们加入 Open 表。

4)如果 Open 表为空,则失败退出,无解。否则,从 Open 表中选择  $f(n)$ 值最小的一个节点  $n$ 。

5)把节点  $n$  从 Open 表中删除,然后添加到 Closed 表中。

6)如果节点  $n$  是目标节点,则成功找到路径退出循环,得出最佳路径。否则进入第 7 步。

7)继续对节点  $n$  寻找周围所有可达到的节点  $m$ 。

① 如果节点  $m$  不在 Open 表和 Closed 表中,把它添入 Open 表。将  $n$  标记为  $m$  的父节点,同时记录  $m$  节点的  $f(m)$ 值。

② 如果节点  $m$  已经在 Open 表中,则需要比较刚才算出的  $f(m)$ 的新值和该节点在 Open 表中的旧值。如果  $f(m)$ 新值比较小,将表中  $f(m)$ 旧值替换掉,同时将表中该节点的父节点更新为  $n$ 。

③ 如果  $m$  为障碍的节点或者和已经在 Closed 表中的节点,则忽略。

收稿日期:2011-08-15

作者简介:王豫峰(1982-),男,山东平度人,硕士,主要研究方向为计算机应用;韩璞(1985-),男,河南南阳人,硕士,主要研究方向为计算机应用。

8)如果目标节点已经进入 Closed 表中,则路径已经被找到。否则,跳转到第4步。

## 2 A\* 算法在游戏寻径中的应用

在游戏中,地图一般是由一些小网格拼接而成的。游戏运行过程中,当切换到某个场景时,首先需要读取地图网格信息,然后按照数据在不同的网格中放置障碍物,最后将 UI 效果叠加在网格上,在屏幕上显示。游戏角色在地图上行走时,通过判断角色当前网格周边的是否可通,从而计算出下一步的行走路线。

本文采用 12\*12 的矩形方格来模拟游戏地图,其中黑色矩形块表示障碍,0 为起点单元格,1 为目标单元格如图 1。

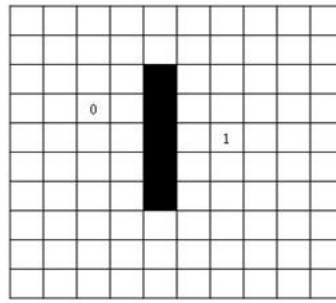


图1 12\*12 的网格地图

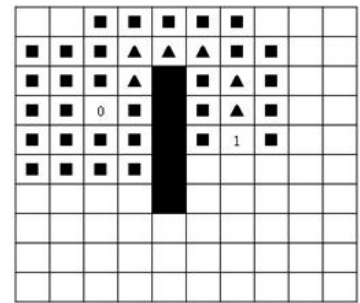


图2 A\* 搜索后的地图

A\* 算法从一个网格的起始位置开始,先进行广度优先搜索,再进行启发式搜索。对网格地图中当前节点进行一轮八个方向的邻域搜索后,利用估价函数对搜索到的节点进行估价,从而选出最好的节点。然后,再从这个最好的节点开始进行下一轮搜索直到搜索到目标节点为止。

A\* 算法的评估函数为  $f(n)=g(n)+h(n)$ 。在本次搜索中, $g(n)$ 函数表示从起点移动到节点  $n$  所需要的时间, $h(n)$ 函数表示从节点  $n$  移动到终点的估计时间。这种搜索方法称为启发式的,因为整个路径是未知的,我们通过一个估价函数对每个节点进行评估,有可能路上存在各种障碍(墙,花,山堆等等)。在本例中我们令水平和垂直移动的时间为 10,对角线方向移动时间 14。

在第一次搜索时会将起点节点周围八个节点的  $f$  值算出来。从左上方节点开始,顺时针旋转,各节点的  $f$  值分别为:74、60、54、40、74、60、74、60。在第一轮搜索中起始节点右边的节点  $f$  值最小,对这个节点再进行搜索的时候,发现右边是障碍物过不去,只能往上。如果往上又到了起始节点的右上方。所以最开始应该将起始节点的右上方节点放到 Closed 表中。如图 2 所示,正方形表示 A\* 算法在当前网格内需要搜索的节点,三角形表示通过 A\* 算法搜索后得到的最佳节点。

## 3 结论

在游戏开发中,人工智能是一个非常重要的模块,游戏角色的自动寻径问题是人工智能在游戏应用中的体现之一。在目前的游戏开发中,大部分游戏的寻径算法都是使用的 A\* 算法,也是目前最有效的一种智能寻径算法。

A\* 算法在搜索计算过程中通过引入估价函数来实现对最佳节点的选择,从而不用遍历整张地图的每一个节点,使得计算复杂度减少很多,实现了快速、方便、高效的搜索路径。一般游戏地图越大,A\* 算法搜索效率也就越高越明显[4]。

## 参考文献:

- [1] 赵明茹,郭键,孙媛.A\* 算法在地图寻径中的应用[J].科技信息,2009(31).
- [2] 漆阳华,杨战平,黄清华.A\* 的改进路径规划算法[J].信息与电子工程,2009,7(4).
- [3] 陈刚,付少锋,周利华.A\* 算法在游戏地图寻径中的几种改进策略研究[J].科学技术与工程,2007(8).
- [4] 詹海波.人工智能寻路算法在电子游戏中的研究和应用[D].华中科技大学,2006.