**杭州电子科技大学**

**毕业设计（论文）文献综述**

|  |  |
| --- | --- |
| **毕业设计（论文）题目** | 基于Godot引擎的第三人称视角冒险游戏设计与实现 |
| **文献综述题目** | 游戏中的寻路算法 |
| **学 院** | 计算机学院 |
| **专 业** | 计算机科学与技术 |
| **姓 名** | 金嘉浩 |
| **班 级** | 14052313 |
| **学 号** | 14051616 |
| **指导教师** | 陆剑锋 |

**游戏中的寻路算法**

1. **前言**

人工智能是游戏设计过程中的一个重要组成部分，寻路是人工智能应用于游戏中最基本的问题之一。在当今的游戏工业界，A\*算法是被大家承认和广泛使用的人工智能寻路算法，也是最有效的最短路径搜索算法之一[12]。A\*算法实际上是一种基于广度优先搜索基础上的启发式搜索算法。在实际应用的过程中A\*算法也存在许多局限性。

近年来，路径查找的研究取得了重大的进步。研究人员根据需要解决的具体问题提出了一些新的寻路算法，如势场法、群集算法（Flocking Behavior）；也有的对A\*算法进行拓展和改进，提出了分层A\*算法、D\*算法、HPLPA\*算法等[1]。尽管如此，仍然有许多问题困扰着研究人员。

1. **主题**
   1. 基准寻路算法：A\*算法

A\*算法是一种标准的路径搜索算法。它由斯坦福研究中心的Peter Hart, Nils Nilsson和Bertram Raphael在1968年首次提出。

在一个基准路径搜索算法中有三个主要的元素：地图的图表示；搜索算法；引导搜索算法的启发式函数[1]。地图网格化是将地图转化为搜索图的一个普遍方案，地图被分割为一个个方块，通常称为瓷砖。移动单元一次可以移动到一个瓷砖上。A\*算法采用启发式函数h(n)，它估计移动单元距离目标位置的距离。曼哈顿距离是被普遍接受的启发式函数。在四方向连接地图中，给定两个节点坐标，，曼哈顿距离被定义为。A\*采用估价函数f(n)=h(n)+g(n)对当前的搜索位置进行评估，从可移动的瓷砖中选择估价最小的瓷砖加入路径[15]。

A\*算法作为基准寻路算法存在着一些缺点。第一，当地图很大时，地图网格化后的瓷砖数量会很多，A\*算法所需要的时间将增加。第二，单位移动的方向有限，只能是上下左右和斜行，八个方向。第三，单位体积过大时会超出瓷砖的大小。第四，很难处理动态地图的情形。除此之外还有其他一些缺点。

为了解决以上一些问题，研究人员对A\*算法进行了改进，也提出了一些新的寻路算法。

* 1. 效率更高的启发式函数

曼哈顿距离是一个在网格地图上简单的，计算快速并相对准确的启发式函数。随着研究的进展，研究人员提出了一些新的启发式函数，这些函数能够减少在搜索中可能出现的状态，加快路径搜索速度。

Goldberg和Harrelson将可接受启发式描述为他们ALT搜索算法的一部分，使用预处理的方式生成一个查找表。他们的算法可以用于非均匀的地图，如路线图和网图。

Björnsson和Halldórsson提出了dead-end启发式和gateway启发式[5]。这两种方法将地图分解为不相交的区域。Dead-end启发式会识别解决给定实例时无关的区域，并将这些区域中所有的节点启发值设为无限大。Gateway启发式识别分区中相邻区域之间的节点，并预先计算一张入口对之间最佳距离的表格。

高效率的启发式函数普遍使用了预计算和额外的内存去存储预计算的结果，从而获得更高的准确度和效率。

* 1. 分层路径搜索算法

Holte等在1996年提出了将地图进行分层描述的思想Hierarchical A\*，目的是为了减少整体搜索耗费。顾名思义，分层方法是将游戏整体世界进行抽象，创建一个层次结构，在路径搜索时首先搜索层次较高较简单的一层，然后逐层深入。这种方式可以限制搜索空间。对比于非分层搜索，分层搜索的优点是搜索时间和内存使用相对较小。

后来，A Botea,M Müller等人提出了分层路径搜索算法HPA\*，使得搜索空间大大减小，有效降低了基于网格地图路径搜索问题的复杂性[6]。

* 1. 动态路径搜索算法

在一些游戏如即时战略游戏中，游戏环境往往是动态的，地图中可通行的节点可能发生变化，节点之间连接的边的权重也可能发生变化。因此，随着时间的推移，游戏角色需要实时更新其路径，对变化的地形做出感知和反应，这就是动态路径搜索。

S. Koenig和M. Likhachev发展了A\*的增量版本LPA\*方法，用于解决动态环境中有确定起点和重点的相似路径搜索问题[9]。LPA\*的第一次搜索与A\*相同，后续的搜索重新利用了与之前搜索树相同的部分，利用之前搜索存储的信息快速更新找到最短路径。

* 1. 基于三角形划分的寻路算法

基于网格的地图寻路算法在地图基础元素形状不规则时就不是那么有用了，同时如果地图的大部分区域是空的，根据网格划分会浪费很多空间。为了解决这些缺点，Demyen和Buro提出了两种寻路算法TA\*(Triangulation A\*)和TRA\*(Triangulation Reduction A\*)[7]。

TA\*使用DCDT(dynamic constrained Delaunay triangulation)表示障碍物是多边形的地图，并通过在包括已经划分的三角形的图上运行一个类似A\*的算法来为一个圆形物体找到一条可能是任何角度的最优路径。

随着用几何体型代表地图的方法出现，基于网格方法的种种限制被解除。

星际争霸2的游戏引擎使用的就是TA\*算法的一个变体。

* 1. 势场法

势场是一个来自机器人领域的概念，它是由Khatib首次提出的，用于机器人和移动机器人的实时避障。它通过在虚拟世界中的重要位置放置吸引或者排斥电荷来工作。

需要移动的单位带有排斥电荷，当计算该单位移动到目标点的路径时，在目标点放置一个吸引电荷，以不同的方式辐射开来（圆形、四边形等）。这些电势的数据会以一定的形式存储，将地图中所有元素的电势相加可以得到一个新的反应当前世界电势的矩阵，单位选择周围电势的最高点移动。

在Hagelback和Johansson的一系列论文中，他们讨论了在开源即时战略游戏ORTS中使用势场法来寻路。他们建议为游戏中每一个有趣的对象，友善的、中立的或者敌对的都根据当前的代理类型和影响对象建立一个势场[3]。

Johan Hagelback的系列论文中将势场法应用在星际争霸的AI当中，并提出了势场法和A\*算法混合的寻路方法[4]。

* 1. 分层动态路径搜索

国内由河北大学李艳副教授领导的研究小组在2012年提出了一种分层动态路径搜索算法HPLPA\*[13]。该算法结合了增量路径搜索算法LPA\*和分层路径搜索HPA\*算法。该算法能有效避免原始LPA\*对于动态变化的节点位置的过度敏感，且能够明显提高搜索的效率。

HPLPA\*比LPA\*和HPA\*更适用于大型游戏地图环境中的动态路径搜索，其不足之处在于某些情况下更新抽象图的时间较长。

1. **总结**

本文首先介绍了基础的路径搜索算法A\*算法，以及A\*算法的优缺点和可以改进的方向。随后，以A\*算法为基准，介绍了多种游戏中应用的路径搜索算法，包括他们的部分发展历史，简单的原理和优缺点。这些算法都在某个或者某些方面优于A\*算法，但也存在一定的缺陷，需要继续研究和改进。

其中，通过预计算改进启发式函数可以让算法效率更高，更准确，但是需要额外的空间；分层路径搜索方法通过将地图分层描述，分层计算的方法降低了搜索的复杂性；动态搜索算法让单位在动态的游戏环境中寻找路径成为了可能；基于三角形划分的寻路算法解除了基于网格划分地图方法的限制，被应用于即时战略游戏中；基于势场的寻路算法能够让单位根据势场大小移动到一个相对正确的位置，已经被用到AI开发中。

路径搜索的研究并没有冷却的迹象，随着研究的深入，路径搜索算法的局限会越来越小，商业游戏和学术研究之前的差距也会越来越小[1] 。在将来，我们会看到越来越多更优化的寻路算法出现在游戏当中。

1. **参考文献**
2. Botea A, Bouzy B, Buro M, et al. Pathfinding in Games[J]. 2013.
3. Cui X, Shi H. A\*-based Pathfinding in Modern Computer Games[J]. International Journal of Computer Science & Network Security, 2011, 11.
4. Hagelback J. Potential-field based navigation in StarCraft[C]. Computational Intelligence and Games. IEEE, 2012:388-393.
5. Hagelback J. Hybrid Pathfinding in StarCraft[J]. 2016, 8(4):319-324.
6. Björnsson Y, Enzenberger M, Holte R C, et al. Fringe Search: Beating A\* at Pathfinding on Game Maps[C]. IEEE Symposium on Computational Intelligence and Games. DBLP, 2005:125--132.
7. Botea A, Müller M, Schaeffer J. Near optimal hierarchical path-finding[J]. Journal of Game Development, 2004, 1(7):7--28.
8. Demyen D, Buro M. Efficient triangulation-based pathfinding[C]. National Conference on Artificial Intelligence. AAAI Press, 2006:942-947.
9. Korf R E. Real-time heuristic search[J]. Artificial Intelligence, 1990, 42(2):189-211.
10. Koenig S, Likhachev M. Real-time adaptive A\*[C]. International Joint Conference on Autonomous Agents and Multiagent Systems. DBLP, 2006:281-288.
11. Qiu L, Zhang H. Implementation of Path Finding on 2D Game Maps[J]. Journal of Hunan University of Technology, 2012.
12. 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.
13. 杨科选. 人工智能寻路算法及其在游戏中的应用研究[D]. 中南大学, 2009.
14. 李艳, 陈彩, 李铁松,等. 游戏地图中的分层动态路径搜索算法[J]. 计算机工程, 2012, 38(2):288-289.
15. 余帅. 即时战略游戏中基于势场的交互寻路方法[D]. 河北大学, 2014.
16. 吴润方, 王鲁. A\*寻路算法在即时战略游戏中的应用[J]. 科技广场, 2016(4):164-166.
17. 邱磊, 张辉. 2D游戏地图的寻路实现[J]. 湖南工业大学学报, 2012, 26(1):66-69.

**文献综述考核表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **指导教师对文献综述的评语：**  **指导教师 （签名）**    **2018年 3 月 16 日** | | | | |
|  |  |  | **建议成绩** |  |
| **评阅小组或评阅人对文献综述的评语：**  **评阅小组负责人或评阅人 （签名）**    **2018 年 3月 16 日** | | | | |
|  |  |  | **建议成绩** |  |