基于A\*算法的《明日之后》模拟生存游戏线路规划

姓名：欧祉辛

学号：20020153

单位：研究生院五大队二十五队

姓名：韩建臣

学号：20020083

单位：研究生院五大队二十五队

摘 要

生存类游戏在近年来的游戏市场中越来越受玩家欢迎，如开放沙盒世界类游戏《Minecraft》、末日求生类游戏《Don't Starve》等。本文通过对网易旗下生存类游戏《明日之后》的游戏设定进行分析和抽象建模，利用A\*（A-Star）算法，在给定的游戏场景中找到最优路线。

对问题一，。

对问题二，。

对问题三，。

最后对本文所建立的模型进行了讨论和分析，综合评价模型。

关键词：A\*（A-Star）算法

1.问题重述

1.1　问题背景

生存类游戏在近年来的游戏市场中越来越受玩家欢迎，如开放沙盒世界类游戏《Minecraft》、末日求生类游戏《Don't Starve》、孤岛求生类游戏《ARK:Survival Evolved》等。生存类游戏蕴含着丰富的哲学思想，如苏格拉底提出的“人生就是一次次无法重复的选择”，又如通过游戏来反思战争所带来的的人性变化等。“生存还是毁灭，这是一个问题”，这句话不仅是对于现实生活有着非凡的哲学意义。对于生存类游戏来说，这也同样是一个永恒的命题。生存类游戏在人为设立的重重“规矩”之下让玩家遵守生存法则之余，让他们通过熟悉这些“规则”去寻找到一些开发者们留下的蛛丝马迹，从而最终逃离生天。

生存类游戏的基本玩法包括采集资源、建造、探索和战斗等，其根本目的在于活下去（即消耗资源）。《明日之后》是网易开发的一款生存类游戏，讲述这样一个游戏情节：病毒肆虐各国，人类文明险些毁灭，为了能够在末世中“活下去”，志同道合的伙伴集结起来，一起在病毒蔓延、感染者遍地、资源有限、天气严酷的世界中求生。在游戏中，人物有两类重要的行为，一类是要通过寻找各种食物来维持自身生存，另一类是通过各种御寒措施来降低自身的寒冷程度从而提高生存能力，我们把这两类行为对应到人物的两个特征，前者称为饱食度，刻画人物饥饿的状态，饱食度为负表示处于饥饿状态，饱食度越小，人物生存越难；后者称为舒适度，刻画人物寒冷的状态，舒适度为负表示处于寒冷状态，舒适度越小，游戏人物生存越难。

1.2　问题重述

以该游戏为背景，设计一个简化的场景。假设游戏人物活动区域为一个空间区域，空间区域中不同位置分布有一些食物和篝火，人物从该区域某一个位置进入，从区域另一个位置出去，如图1-1所示：



图1-1　问题示意图游戏

人物在该区域行走的规则如下：

（1）人物到达食物点吃到食物，其饱食度将提高，提高程度依赖于食物数量；

（2）人物到达篝火位置，其舒适度将提高，提高程度依赖于篝火的大小；

（3）人物在平路（即Z坐标相同）行走100米，饱食度和舒适度均降低5个单位，若走上坡路（Z坐标增加），饱食度和舒适度每走100米均降低6个单位，若走下坡路（Z坐标减少），饱食度和舒适度每走100米均降低4个单位。假设上、下坡已经等效为两点之间直线行走；

（4）当人物到达食物点或篝火点，若饱食度和舒适度中任意一个小于-5，人物将死亡，无法通过食物或篝火提高饱食度或舒适度；

（5）假设人物在开始位置时的饱食度和舒适度均为10；

（6）要求人物到达终点时，饱食度和舒适度均不小于-3。

需建立数学模型和算法解决以下问题：

问题一：规划该人物从起点到终点的路线（用序号表示），使其经过的食物点和篝火点的次数最少；

问题二：在第一问的基础上，进一步考虑人物行走的路径尽可能短，规划其从起点到达终点的路线（用序号表示）；

问题三：在篝火点，游戏人物可以制作火把携带，制作火把将使得饱食度降低0.5个单位，但携带的火把能支持人物行走20米而不降低舒适度。请在第一问和第二问的规划路线基础上，进一步考虑人物可制作火把携带的方案，使得人物到达终点后的饱食度和舒适度尽可能高。

1.3　数据说明

数据文件中给出了食物点和篝火点的信息，第一列为点的编号，第2-4列为食物点或篝火点的位置坐标，第一行为起点信息，最后一行为终点信息；第5列为点的类型，1表示该点为食物点，0表示该点为篝火点，第6列为人物位于该点时可通过补充食物或利用篝火提高其饱食度或舒适度的大小，间接代表了该处食物的数量或篝火的大小。部分数据如图1-2所示。



图1-2　部分数据示例

2.问题分析

2.1　问题一分析

对于问题一

2.2　问题二分析

对于问题二，

2.2　问题三分析

对于问题三，

3.模型假设与约定

（1）人物行走时最小计量单位为1米，即每行进1米时会依据当前地形（平路、上坡或下坡）失去相应的饱食度和舒适度；

（2）；

（3）。

4.符号说明及名词解释

4.1　符号说明

|  |  |
| --- | --- |
| 符号 | 含义 |
|  |  |

4.2　名词解释

4.2.1　A\*（A-Star）算法

A\*（A-Star）算法是一种很常用的路径查找和图形遍历算法，它有较好的性能和准确度。A\*算法于1968年，由Stanford研究院的Peter Hart、Nils Nilsson和Bertram Raphael发表[1]。它可以被认为是Dijkstra算法的扩展。 A\*算法是一种启发式算法，它利用启发信息寻找最优路径。A\*算法需要在地图中搜索节点，并设定适合的启发函数进行指导。通过评价各个节点的代价值，获取下一需要拓展的最佳节点，直至到达最终目标点位置。

A\*算法优点在于对环境反应迅速，搜索路径直接，是一种直接的搜索算法，因此被广泛应用于路径规划问题。其缺点是实时性差，每一节点计算量大、运算时间长，而且随着节点数的增多，算法搜索效率降低，而且A\* 算法并没有完全遍历所有可行解，所得到的结果不一定是最优解。算法描述如图4-1所示。



图4-1　A\*算法完整描述

5.模型建立与求解

5.1　模型设计

5.1.1　数据预处理

首先，

5.1.2　模型建立

。

5.2　问题求解

5.2.1　问题一求解

（1）求解步骤

针对。

（2）求解结果

依据

5.2.2　问题二求解

（1）求解步骤

针对。

（2）求解结果

依据

5.2.3　问题三求解

（1）求解步骤

针对。

（2）求解结果

依据

6.模型评价

6.1　模型优点

考虑。

6.2　模型缺点

从

7.模型推广与改进

对于

8.参考文献

[1] P. E. Hart, N. J. Nilsson and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," in IEEE Transactions on Systems Science and Cybernetics, vol. 4, no. 2, pp. 100-107, July 1968, doi: 10.1109/TSSC.1968.300136.

[2]

9.附录

9.1　问题一

9.1.1　程序附录

|  |
| --- |
|  |
|  |

9.1.2　结果附录

9.2　问题二程序附录

9.2.1　程序附录

|  |
| --- |
|  |
|  |

9.2.2　结果附录

9.3　问题三结果附录

9.3.1　程序附录

|  |
| --- |
|  |
|  |

9.3.2　结果附录