

Java 大作业报告

计算机学院

姓名：陈少鹏 学号：1110310322

手机号：18686831710 email：lsmcsp@gmail.com

迷宫游戏设计报告

1. 游戏简介

本游戏能够随机的产生迷宫地图，支持用户键盘操作角色在规定时间内到达目的地。并且能够给用户提示信息。

2. 迷宫地图生成算法

迷宫地图的要求是任意两个节点之间存在且只存在一条唯一的路径。也就是说，所有的节点与节点之间边（相邻的节点连通则有边）构成一棵树。于是，问题就可以变成如何生成这棵树。可以采用并查集合并的方法来实现。算法描述如下：

1. 初始化所有集合：

$\{NODE_1\}, \{NODE_2\}, \dots, \{NODE_n\}$

这里 n 为所有节点的个数。如欲生成 40×30 的迷宫，则 n 为 1200。每一个节点都代表地图中的每一格。初始化这样，表示每一个节点都与外界没有连通。相当于 n 棵子树。

2. 随机选取两个集合合并：

用 S_i 表示集合 i ，则随机选取 S_i, S_j 作为将要合并的集合。如果 S_i, S_j 可以合并，即 S_i, S_j 中包含两个节点 $NODE_m, NODE_n$ 是相邻的，则合并这两个集合；否则重新选取。这一个过程相当于在 $NODE_m, NODE_n$ 之间打通了一条通道。如下图所示：

	$NODE_n$		
	$NODE_m$		

3. 重复 2 直到所有集合个数为 1：

也即是所有的子树合并成了一棵完整的树。在迷宫地图生产问题上来讲，就是生成了一个任意两个节点之间存在且只存在一条唯一的路径的地图。

3. 路径计算算法

该游戏需要实现一个提示功能，即给用户提示下一步的位置。一种可行的方法是每次都遍历当前位置的其他路径，如果一条路径（迷宫地图生成算法保证有且只

有一条)可以到达目的地,则选取该路径中的下一步作为提示。这样做实现起来比较简单。即是对图的搜索。但是性能不佳,因为每次都需要重新计算路径,这是一个额外的计算。我们可以有以下的改动,使得只需要计算一次:

1. 在每次生成地图后,计算从出发位置到目的位置的路径。将其存储到栈中。
2. 每次用户按键触发一次操作。如果用户的去向与栈顶保存的一致,则将栈顶弹出;否则将用户的去向的反方向压入栈内。
3. 每次提示信息只需要返回栈顶即可。

4. 倒计时时间计算规则

该游戏要求用户在规定时间内达到目的地。确定限制时间是必须的。因为过短的时间完成不了游戏;而过长的时间没有挑战性。即这个时间的设置必须考虑挑战性与合理性。有两种策略来设置这个时间:

1. 时间都是固定的一个值,可以由大量的统计得出。
2. 时间不是固定的,根据游戏的复杂度动态生成。

策略 1 有一个明显的弊端:每一次生成的迷宫都是不一样的,该次生成的迷宫的复杂程度也不一样。所以使用固定值会使得游戏有些时候太简单,而有些时候又太过于复杂。所以衡量该迷宫的复杂度是一个值得讨论的问题。

我们采用以下公式计算倒计时时间:

$$\text{time} = \frac{\text{Size of Maze}}{\text{Traversal Factor}} + \frac{\text{Length Of Shortest Path}}{\text{Click Factor}} + 1$$

Traversal Factor 代表人的大脑搜索迷宫的速度,我们设置为 16Hz。*Click Factor* 为人的点击频率,设置为 5Hz。后面的常数项 1 是为了避免时间为零出现。

在理想情况下,忽略了思考时间,人完成该比赛只需要 $\frac{\text{Length Of Shortest Path}}{\text{Click Factor}}$ 的时间,即将时间全部花费在了控制角色移动上面。而我们通过 $\frac{\text{Size of Maze}}{\text{Traversal Factor}}$ 来衡量人的思考时间,这是合理的。因为遍历这个迷宫的复杂度就是 *Size of Maze*。经过实际的试验,这个公式确实是可行的。

5. 总结

通过这次实践,学会了如何从设计开始,到付诸行动地完成一个完整的项目。在不断求精的过程中提升了自己对算法,数据结构,已经高级程序设计语言(Java),面向对象设计的认识。