**搜索大作业实验报告**

兰国兴 自11 2011011464

1. 实验目的
2. 加深对搜索算法的理解，学会利用搜索算法解决实际问题。
3. 锻炼编程能力、软件界面设计能力。
4. 实验要求

自己设计一个利用搜索解决的问题，自己实现搜索算法以及必要的界面。

1. 实验内容

利用C#程序设计语言设计一个与“十滴水”类似的单机游戏，并利用搜索算法实现自动求最优解。

“十滴水”游戏规则如下：

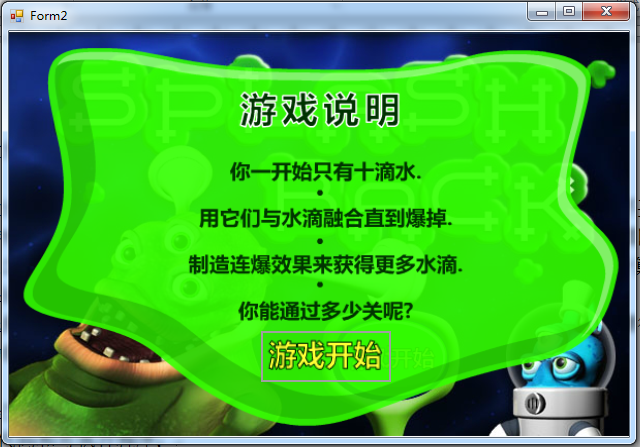
在一个6×6方格中，分布着一些水球，水球的含水量有4个等级，由低到高依次标为1、2、3、4。玩家可以随便点击某一水球，如果点击前水球的含水量为1、2、3，则点击后该水球的含水量增加一个等级；如果点击前已经为4，则水球爆破，朝上下左右四个方向飞溅出水滴，飞溅出的水滴遇到静止的水球则被吸收，同时该水球的含水量也加1（类似地，如果该水球吸收水滴之前已经为4，则爆破并朝四个方向飞溅出水滴）；如果点击处没有水球，则会该位置凭空产生一个含水量为1的水球。每点击一次会使玩家所拥有的水滴减1（游戏开始前有10滴），点击后如果产生连锁爆破，每连续爆破3个水球会使玩家的水滴数加1。当所有水球都爆破后，本轮游戏结束。

1. 软件设计说明
2. UI设计

双击可执行文件后首先弹出如下欢迎界面：



点击“游戏开始”图标，进入游戏说明界面：

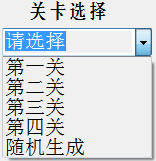


点击“游戏开始”图标进入正式游戏的界面：

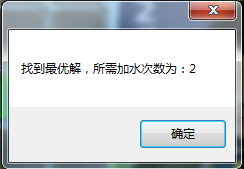


初始时6×6方格中无水球，剩余水滴10滴，得分（也即剩余水滴数）为10分，游戏模式默认为玩家来玩。

点击“关卡选择”的选择键，可以进行关卡设置。（如下图所示）本程序固定关卡有四关，难度由易到难。此外，玩家可以让程序在6×6方格中随机生成水球分步图。

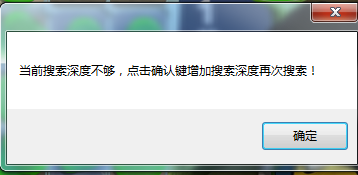


选择好关卡后，玩家可以选择“我来玩”或“电脑来玩”。选择完毕后，点击“确定”按钮即可开始游戏。如果选择的是“我来玩”，则由玩家自行点击6×6方格的任意一个位置进行加水；如果选择的是“电脑来玩“，则程序会自动搜索使当前关卡结束后得分最高的的解法，搜索成功后弹出如下对话框：



上述对话框显示此最优解需要加水的总次数。点击确定后程序会自动执行加水操作，直至本轮游戏结束结束。

需要注意的是，在“电脑来玩”的模式下，程序的搜索算法是在一定的深度范围内进行的。所谓深度，就是程序设定的最多加水次数。初始深度为4，如果在此范围内没有找到解，则会弹出如下对话框：



点击“确定”，程序会将搜索深度加1再次进行搜索，如此重复进行直至找出最优解。

1. 搜索算法说明

本程序利用一个6×6矩阵存储每个位置上的水球信息（水球的含水量等级，用0、1、2、3、4表示，0表示没有水球）。

本搜索算法的状态空间为每次加水后的水球分布状态，也即6×6矩阵的状态。搜索算法结合了BFS和DFS两者的优点，即限定DFS的搜索深度，如果到达该限定的深度或者在该深度之内找到了一个解，则对当前结点n的兄弟结点进行搜索。如果n没有兄弟结点或兄弟结点都搜索完毕，则转向n的父结点n-1的兄弟结点进行搜索，如此往复，直至找到最优解。

算法流程图如下：

 在实际编写程序时，正如在UI设计说明里面讲的，初始时限定深度值为4。如果在此限定深度内没有找到解，则限定深度加1并再次进行搜索。之所以这样设计，是因为随着搜索深度的增加时间开销会迅速增加，而有相当一部分的布局情况下最优解的加水次数不超过4。故对于具体的一个问题，可以先用限定深度为4来进行搜索，如果搜索不到解再逐渐增加限定深度的值。体现出算法的一定“自适应性”。

此外，在实际扩展一个节点的后裔时，不必一次性把其后裔全部扩展，而是先扩展一个，对该节点搜索完毕后，再扩展其兄弟结点，这样可以大大节省空间开销，因为每增加一层，只需增加一个结点的空间开销。

1. 总结与感想

实际测试时发现，随着水球个数的增加以及布局复杂度的增加，搜索时间可能需要几十秒甚至几分钟，故需要耐心等待。

本次大作业让我对于搜索算法有了更深的理解，并初步学会如何利用搜索算法来解决实际问题。此外，本次大作业对我的编程能力和自学能力（自学C#）也是一次很好的锻炼，相比以前，这次编程过程中我遇到的程序语法错误大大减少，故有比较充裕的时间来解决逻辑与算法方面的问题。

不足与遗憾：由于时间比较紧迫，算法基本上可以算是蛮力的，采用遍历的方法尝试各种情况，希望在人工智能课程的后续学习中能够利用所学对当前算法进行进一步优化或者提出更好的算法。此外，在界面设计中没有把水滴飞溅的效果展示出来，也是一大遗憾；界面友好性也有待提高。