火柴棍实验报告

一.任务描述与问题建模

将本次大作业主要分解为以下几个任务：

1.给出一个式子，搜索如何可以移动一根或两根火柴棒将其变为等式的核心搜索算法

2.根据选择的难度，随机进行出题

3.玩家可以拿走放下火柴的 UI 设计

4.给式子用火柴棒显示，和将当前火柴棒组成的式子读出来的函数。

（相当于搜索和图像界面的一个对接）

二.算法设计和实现

2.1 搜索部分

数字到数字的变换建立了几个 10×10 矩阵：move1, add1, sub1, move2, add2, sub2, move1_add1, move1_sub1。比如 move2[i,j]=0 表示数字 i 自身移动两根火柴无法变为数字 j， move1_sub1[i, j]=1 表示数字 i 移走一根火柴棍后自身再移动一根火柴棍可以变成数字 j。

移动一根火柴时变成等式有以下几种可能：

1.自身移动 1 根

2.一个数给另一个数一根火柴

3.加减号变化，还有一个数增加或减少一根火柴

移动两根火柴变成等式时有以下几种可能：

1.自身变两根

2.一个数给另一个数两根火柴

3.一个数加两根火柴，来自两个不同的数（也可以是来自+变-）

4.一个数减两根火柴，给两给不同的数（也可以让-变+）

5.一个数加一根自己变一根，来自一个数（也可以是来自+变-）

6. 一个数减一根自己变一根，来自一个数（可以来自-变+）

7.加乘变换

用 for 循环遍历这些情况即可。

移两根之所以不能用移一根再调用移动一根使成立的原因是：一个数动两下到另一个数的过程中间可能会出现不是数的火柴摆放，而没有对不是数的拜访建立可以查的表。

可以写一个 addmatch1 和一个 submatch1（加一根或减一根使等式成立）在移动两根使成立的循环中调用，可大大减少 for 循环的层数！这个和建立的转换表配合使用使得搜索部分的代码行数不会很长，debug 也方便。

2.2 出题部分

随机生成前两个数和符号，计算第三个数，超过 100 或是负数就重新生成，得到了一个等式。出题在等式的基础上根据相应难度变化就好。再生成一个随机数决定变哪个数或是符号。

简单模式：将随到的数看看加一根减一根能变成什么，再遍历其它数做相应改变

中等模式：将一个数给另一个数两根或是加减变换

困难模式：有一个数加（减）一根变一根，另一个数或符号做相应变化

等式模式：直接对改等式用 movematch2（搜索移动两根火柴棒使等式成立的函数）进行搜索，找到不等于原式的解就加入题目。

2.3 显示与读取显示

首先对 button 进行改名，eg 第二个数的第三根火柴就叫 button23

给式子去显示：对所有 button 遍历，根据 button 后第一个数和要求显示的等式看这个数是什么，设为 a，建一个 10×7 的矩阵为 10 个数对应七个火柴是否为 1. 查这个矩阵，看数 a 对应的七根火柴哪些是 1，在看这个 button 的最一个数，就知道这个 button 要不要是 1 了。

读取火柴得到当前式子：建一个 6×7 的矩阵，遍历所有 button 后知道这六个数的七根火柴的为 1 情况，对每个数那一行查七根火柴为 1 情况就知道这个数是什么了。对于符号，加减乘图片的 size 是不一样的，可根据这个来读取符号是什么。

三.UI 设计和使用说明

鼠标拖动火柴移动和改变角度其实都没什么问题，加上一个根据位置判断是想放到哪个位置让火柴停那儿就行，但是符号的位置加减乘太密集了，考虑到这样移动不准且设计的移动两根火柴还没成等式就对玩家判断游戏失败，这样游戏体验

就会很差，最后决定还是写成点一下式子里的火柴就相当于把它拿下来，然后放在雪人的手上，点击式子里空的位置就把火柴放上去，然后雪人手上火柴少一根。

每一根火柴都是一个 button，把 Backcolor，Forecolor，Flatsytle 都进行相应更改并把边线宽度调成 0 就能使 button 在不放背景图时真正的透明，这样点一下就把背景图换成火柴，再点一下就把背景图变成 null，视觉效果还是很好的。为了让游戏体验更好，编程使鼠标放在 button 上时，button 显示边线，鼠标移开时边线消失。

符号由于放单根火柴太密了，最后就把符号集成成一张图，再加两个 button 用于使符号变化，符号变化就放不同的图。

为了提示玩家移动了几根火柴了，或者说拿起来几根火柴了，在右下角的小雪人的手上做了点工作，就模仿我们自己玩游戏，拿起来的火柴棍就拿在手上，使游戏体验更好。

使用说明：点击火柴棍将其拿起来，点击空位将其放下去，符号通过点其下方两个按钮来改变。选题库的题就先选难度再选题目，难度有“简单”“中等”“困难”“等式模式”，只有简单模式是移动一根火柴棒，其它都是移两根，等式模式初始式子就是成立的等式，要求移成另一个成立的式子。想自定义题目的话就在对应窗口输入式子，只要正常输入注意不加空格就行，输入完点击定义完成即可。自己做题时注意只能移动对应根数的次数，请想好了再移，否则移两（或一）次就会提示游戏失败。可以随视按右下角的答案按钮查看答案，若是自定义的题目无解时会提醒此题无解。

四.实验总结

其实思路清楚了写这个还是很快的，最好先设计好任务的分解，可以分开写分开调试，大大加快完成速度。而且多把一些可以重复利用的部分封成函数，调试和做改动都会变快，而且代码的思路会更加清楚。

数据结构真的很重要，想清楚这个问题用什么数据结构可以让代码量大大减少，思路也清楚也就更不容易错了，这次通过查矩阵表来知道数字间的转换关系就很方便。

按钮处理透明了之后，视觉效果一下子就上去了，真的好用！以前把 backcolor 调透明后发现没用就算了，这次好好查了下，调那 4 个地方才能做到真正的 buttton 透明，以后都可以用，学到了。

和七巧板比起来火柴棍真的很简单，我写七巧板写了三十左右个小时，但是算法不太好，最后调通后只能拼很简单的图形，对一个节点扩展 open 表就要快 1 分钟，题目难了就会超运行内存，最后只好在这周二放弃转而写火柴棍。我七巧板用 C#写的，按说应该是运行比较快的，但是和其它用 python 写的同学比，同样的功能他用 opencv 里的函数实现我自己写，却是他的运行更快，不得不承认成熟的库还是厉害啊，而且对于内存管理也更好，我自己写代码时对 bitmap 和 graphics 建了很多却没有资源释放所以内存就炸了，以后还是从简单的任务开始加强对内存管理的意识。以及以后有时间的想研究一下库函数是怎么写的，加强自己代码能力。没写出来七巧板还是很不甘心哎。