



C语言程序设计

课程设计报告

**设计题目：计算机辅助教学（CAI）软件开发：教小学生学算术**

班 级 电信学院15级信息工程2班

姓 名 刘可儿

**评级： 优秀**

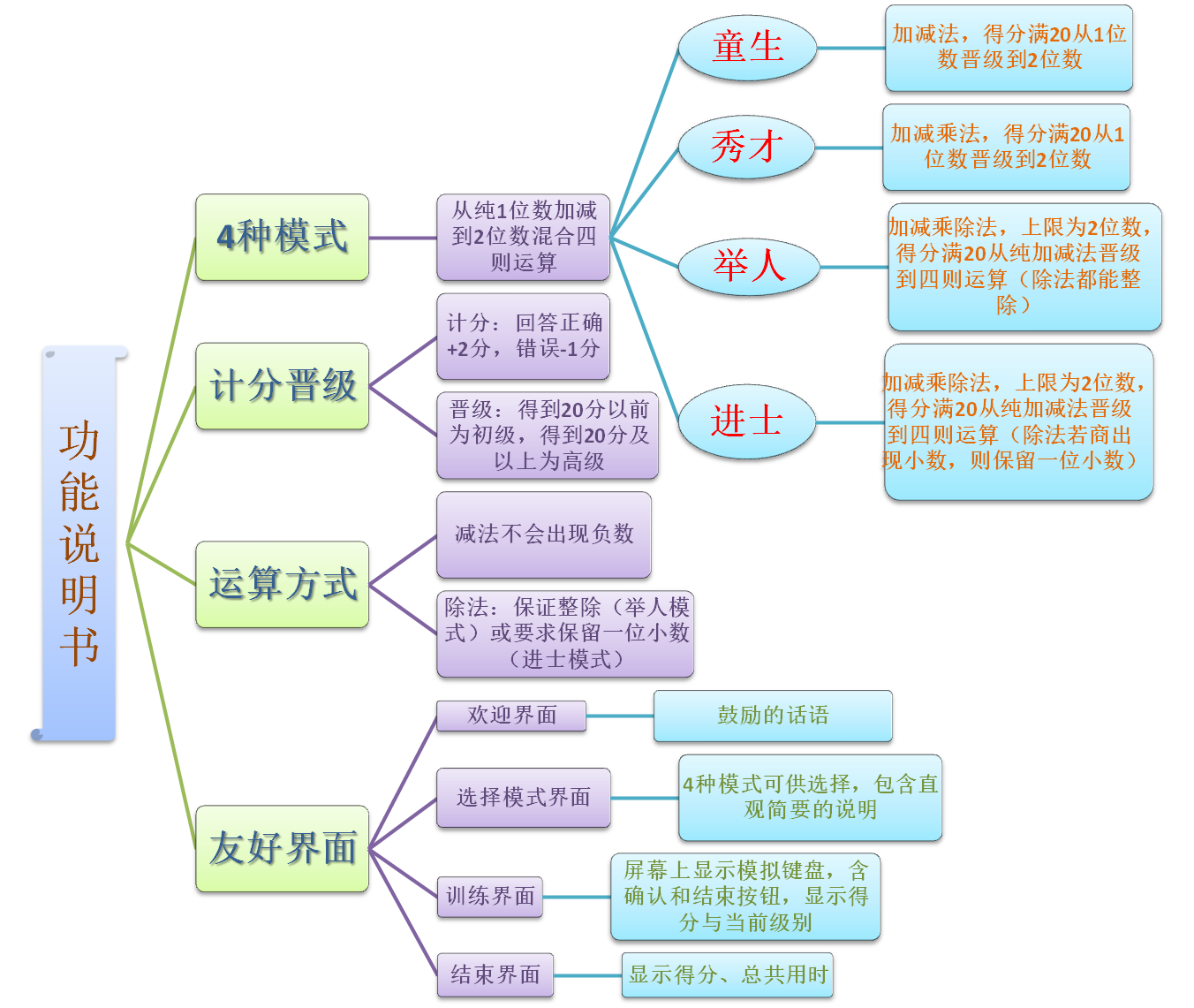
|  |  |
| --- | --- |
| 完成方式：（单选） | 自我评分：（单选） |
| ○ 独立完成 | ○ 优秀 |
| ○ 参考他人工作的基础上完成 | ○ 良好 |
| ○ 在他人的帮助下完成 | ○ 中等 |
| ○ 完整拷贝他人成果 | ○ 及格 |
|  | ○ 不及格 |

2016年 6 月 1 日

|  |
| --- |
| 设计任务：**C语言课程设计9**  题目：题目：计算机辅助教学（CAI）软件开发：教小学生学算术  功能：可做个位数，十位数的加，减，乘和除，减法不能得负数，除法要除尽（注：本软件设置了两种除法模式：保证整除与保留一位小数，选择哪种模式取决于用户），屏幕上随机出现两个数，提示小学生给出答案，答错了要提示重新输入，直到答对为止。要统计给分，且累计，够一定分数后可进级，即从个位进到十位。同样，也可降级。（注：本软件晋级模式更加多元，详见功能说明书）  （rand(),#include<stdio.h>）  分步实施：  1、初步完成总体设计，搭好框架，确定人机对话的界面，确定函数个数；  2、完成基本要求：能答对个位数的加法和减法，并有统计功能，即一次答对的成功率为90%上后可晋级，做十位数的加减（注：本软件改为积分晋级制）；  3、进一步要求：完成乘法和除法的学习，也有晋级功能。  要求：  1、用C语言实现程序设计；  2、系统的各个功能模块要求用函数的形式实现；  3、界面友好（良好的人机交互），程序加必要的注释。 |

**一、总体设计**

1、功能说明书



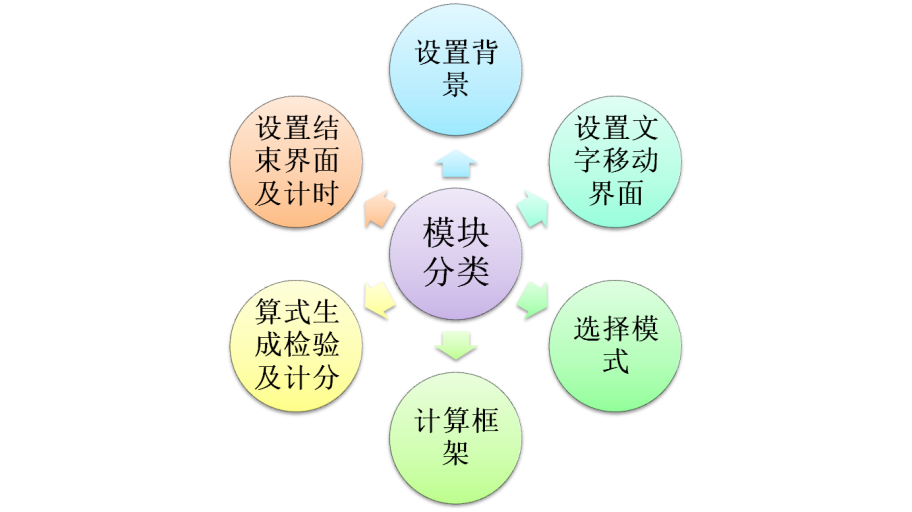
2、独立完成度与平台资源

本口算辅助软件是我在参考了EasyX Library后独立制作的（完成后发到了我的github），从构思、编写、调试、修正，都是我一个人完成。在做成软件后，我将软件发给别人进行用户体验测试，并根据用户反馈改善和更新了部分功能。

本软件开发使用VS2010平台，程序主体使用C语言，其中用到了支持VS平台的图形库EasyX。

3、函数说明

* 功能分类



* 函数声明
* void set\_background();

|  |  |  |
| --- | --- | --- |
| 输入参数 | | 无 |
| 返回值 | 无 | |
| 功能 | 设置界面的背景和窗口标题文字，在每次重绘界面时调用 | |

* void moving\_str();

|  |  |  |
| --- | --- | --- |
| 输入参数 | | 无 |
| 返回值 | 无 | |
| 功能 | 作为欢迎界面，显示作者的头像和下方移动的鼓励学习话语 | |

* int mouse\_click();

|  |  |  |
| --- | --- | --- |
| 输入参数 | | 无 |
| 返回值 | 无 | |
| 功能 | 绘制屏幕键盘与按钮，并实现鼠标点击的实时控制 | |

* int select\_mode();

|  |  |  |
| --- | --- | --- |
| 输入参数 | | 无 |
| 返回值 | 无 | |
| 功能 | 选择模式（童生1、秀才2、举人3、进士4） | |

* void calculate\_base(int time\_start, int mode);

|  |  |  |
| --- | --- | --- |
| 输入参数 | | time\_start:训练开始的时间  mode:所选择的模式（童生1、秀才2、举人3、进士4） |
| 返回值 | 无 | |
| 功能 | 作为计算的框架，控制随机数生成、实时计分（通过调用）、控制晋级 | |

* int cal\_sign (int val1, int val2, int count, int level, int time\_start);

|  |  |
| --- | --- |
| 输入参数 | val1:随机生成的第一个数  val2:随机生成的第二个数  count:当前计分得分  level:级别  time\_start:开始做训练的时间 |
| 返回值 | 经过计算后的最新得分 |
| 功能 | 生成显示在屏幕上的算式，检验得数，实时计分 |

* void draw\_end (int time\_start, int count);

|  |  |  |
| --- | --- | --- |
| 输入参数 | | time\_start:训练开始的时间  count:最终得分 |
| 返回值 | 无 | |
| 功能 | 在结束训练时，绘制结束界面，显示最终得分和所用时长 | |

4、流程结构

使用顺序、分支、循环结构的结合，利用if...else语句、switch语句、while语句等实现总过程。本软件输入依托鼠标点击，输出显示都是基于绘图界面。

5、实现效果

下图为模拟用户使用软件过程中部分界面：

|  |  |
| --- | --- |
|  | * 显示窗口标题“给小学生的口算系统” * 显示作者的“标识”（作者自己的照片）； * 下方是横向滚动的文字——学好数学，从口算开始！数学魅力妙妙妙！ |
|  | * 口算有4种模式：童生、秀才、举人、进士，其难度逐级上升。 * 通过点选对应黄色矩形块，进入对应的模式。相应说明在矩形块右方显示。 |
|  | * 算式在屏幕显示； * 鼠标点击左下方的黄色键盘可以选择数字，选择的数字即时显示在算式等号右方。 * 如果想要清除输入的数，可以点“X”按钮。 * 确认输入的得数就按“确认”按钮。 * 想要结束训练，就按“结束”按钮。 * 得分会在左上方即时显示。 |
|  | * 4种模式都是满20分从低级晋级到高级。 |
|  | * 实时操作：鼠标点击屏幕按钮，算式等号右方可实时显示输入的数字。如图为乘法实例。 |
|  | * 保留一位小数的除法实例（四舍五入） |
|  | * 显示总共用时 * 显示总共得分 |

**二、详细代码设计**

下面的函数只显示主要部分，有详细的设计说明。

1、main函数：（软件运行的流程）

void main()

{

int mode; //mode对应模式1童生 2秀才 3举人 4进士

initgraph(640,480); //初始化绘图图形界面

set\_background(); //设置背景

moving\_str(); //进入界面：先显示进入界面以及滚动文字，一定时间后跳转到模式选择界面

mode = select\_mode();//按钮界面选择mode, 放在这里是为了选择模式后才开始计时

time\_t time\_start; //开始训练的时间：先在此时计时，程序结束后统计总用时长

time\_start = time(NULL);

calculate\_base(time\_start, mode); //计算过程：实现算式生成、结果输入、得数检查、统计分数、状态显示一体化；相关细节的函数的调用在调用该函数时进行

getch(); //任意按键

closegraph(); //关闭界面

}

2、设置背景：set\_background

void set\_background() //设置背景

{

for(int i = 0; i < 640; i++)//设置渐变紫色背景

{

setlinecolor(RGB(100,100,100+int(i/640.0\*155))); //渐变色

line(i, 0, i, 479);}

HWND hwnd = GetHWnd(); //获取窗口句柄并设置窗口标题文字

SetWindowText(hwnd, "给小学生的口算系统");

}

3、绘制有移动文字的欢迎界面：moving\_str

void moving\_str() //移动的文字和欢迎界面

{

settextcolor(RGB(128,255,128));//设置文字颜色格式

settextstyle(50, 0, \_T("隶书"));

setbkmode(TRANSPARENT); //设置文字底色为透明，不要遮挡背景

outtextxy(270,100,"作者：");

IMAGE photo; //显示作者的照片

loadimage (&photo, "个人照缩小.jpg");

putimage(270,170,&photo);

TCHAR str[] = \_T("学好数学，从口算开始！数学魅力妙妙妙！"); //待移动的字符串

int w = textwidth(str); //存储文字对象的宽度高度

int h = textheight(str);

IMAGE temp;//存储临时背景图像

settextstyle(30, 0, \_T("隶书")); //设置下方滚动文字的制式，放在这里才有效

settextcolor(WHITE);

setbkmode(TRANSPARENT); //透明色，是为不遮挡背景

for (int i=0; i<100; i++) //绘制移动的文字

{

getimage(&temp, i, 400, w, h); //暂时存储目标位置的背景

outtextxy(i, 400, str);

Sleep(30);

putimage(i, 400, &temp); //重新加载回背景

}

cleardevice(); //清屏

set\_background(); //重设背景

}

4、选择模式：select\_mode

int select\_mode() //选择计算的模式

{

int mode;

setfillcolor(YELLOW); //画底板矩形

fillrectangle(100, 100, 200, 150); //...其余类似

settextcolor(YELLOW); //文字

settextstyle(50,0,\_T("方正舒体"));

RECT r1 = {200, 0, 450, 80};

drawtext(\_T("模式选择"), &r1, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE); //...其余类似

settextcolor(YELLOW);

settextstyle(20,0,\_T("宋体"));

outtextxy(220, 120, "加减法，从1位数到2位数，得分满20晋级");//...其余类似

MOUSEMSG m; //鼠标操作

while(true) //鼠标操作的while模式

{

m = GetMouseMsg();

if (m.uMsg == WM\_LBUTTONDOWN) //左键按下

{

if (m.x>=100 && m.x<=200 && m.y>=100 && m.y<=150)

{

mode = 1;

break;

}//...其余类似

}

}

cleardevice();

set\_background();

return mode;

}

5、计算框架：calculate\_base

void calculate\_base(int time\_start, int mode)

{

int flag = 1, count=0, levelgrade[4][2]={1,2,3,4,5,6,7,8}; //count是得分,levelgrade是算术级别

while(flag) //flag==1时进行循环

{

int val1, val2; //算式左方的两个值

float input; //存放鼠标按键最终确认得数

srand ((unsigned)time(NULL)); //随机数种子随机生成数 <time.h>

switch(mode)

{

case 1: //童生、秀才级别的都是满20分晋级（从1位数到2位数）;不满20分自动降级了

if(count < 20)

{

val1 = rand()%10; //生成1位数随机数

val2 = rand()%10;

count = cal\_sign(val1, val2, count,levelgrade[0][0], time\_start);

}

if(count>=20)

{

val1 = rand()%100;//生成2位数随机数

val2 = rand()%100;

count = cal\_sign(val1, val2, count,levelgrade[0][1], time\_start);

}

break; //注意位置

//...其余类似

default:

exit(0);

}

}

}

6、算式生成及得数检验和计分：cal\_sign

int cal\_sign (int val1, int val2, int count, int level, int time\_start) //算式生成及得数检验和计分过程，返回得分

{

int a, sign, temp, tag; //temp是临时值，用作交换 tag标志除法算法级别：1是出现题目一定整除，2是可能不整除，保留一位小数

float final, input; //final算式正确结果，input输入最终结果

srand((unsigned)time(NULL)); //随机数种子

switch (level) //选择运算符号级别：%2加减，%3加减乘，%4加减乘除

{

case 1:

sign = rand()%2+1;break;

case 2:

sign = rand()%2+1;break; //...其余类似

default:exit(0);

}

switch (sign)

{ //...其余类似

case 4: //除法

//把整数转成浮点数才能精确计算

if(tag == 2)

{

while(val2 == 0) //除数不能为0

{

val2 = rand()%100;

}

}

if (tag == 1)

{

while(val1%val2 !=0) //不能整除则重新生成随机数

{

val1 = rand()%100;

val2 = rand()%100;

while(val2 == 0) //除数不能为0

{

val2 = rand()%100;

}

}

}

TCHAR d1[5], d2[5], d3[5]; //把数字转换成字符串

\_stprintf(d1, \_T("%d"), val1);

\_stprintf(d2, \_T("%d"), val2);

\_stprintf(d3, \_T("%d"), count);

if (count < 20) //使用count判断而不使用level判断：因为count是实时反映分数的

{

settextcolor(RGB(0,200,200)); //显示当前级别

settextstyle(50, 0, \_T("楷体"));

outtextxy(200,10,"初级");

}

else if (count >= 20)

{

settextcolor(RGB(0,200,200)); //显示当前级别

settextstyle(50, 0, \_T("楷体"));

outtextxy(200,10,"高级");

}

settextcolor(GREEN);

settextstyle(100, 0, \_T("宋体"));//...其余类似

final = (float(val1) / float(val2)); //转换成浮点数才是精确值

temp = (int(final)\*10) + int((final-int(final))\*10+0.5); //目的：保留一位小数，四舍五入（得到乘十倍的数）

if(tag == 1)

{

input = mouse\_click(); //整除了，不会出现小数点

}

else if (tag == 2)

{

if (val1%val2 != 0) //如果正确答案是有小数的（不是整除）,按照用户一般行为，用户会输入小数点

{

input = mouse\_click() \* 0.1; //除法：考虑小数点留一位，特殊操作

}

else

{

input = mouse\_click(); //整除了，不会出现小数点

}

}

while(fabs(input- temp\*0.1)>=1e-6)//如果做错就继续写直到对为止,input浮点数存储可能会有精度缺失

{

if (input == -9999) //鼠标点了结束按钮

{

draw\_end (time\_start,count);

}

if(input!=9999) //如果是真的算错了，那就扣一分；注意9999是清除操作的标志数

{

count = count - 1;

}

cleardevice();

set\_background(); //在清屏后立即重新绘制背景

//...其余类似

}}

cleardevice();

set\_background();

break;

default:printf("出错！");exit(0);

}

count = count+2; //答对加2分

return count;

}

7、鼠标控制：mouse\_click

int mouse\_click() //除法情况下：在母步骤再\*0.1 因为此函数无法返回float

{

int flag = 1, draw = 1;//flag点击循环标记, draw是否显示数字

int i, j,count=0, value = 0;

char c;

char cal\_show[4][3]; //屏幕键盘绘制

TCHAR s1[5];

for (i=0; i<3; i++)

for (j=0; j<3; j++)

{

cal\_show[i][j] = '0'+ (3\*j + (i+1));

}

cal\_show[3][0] = 'X';

cal\_show[3][1] = '.';

cal\_show[3][2] = '0';

//...其余类似

MOUSEMSG m;

while(flag == 1)

{

draw = 1; //标识：draw == 0 时重绘界面但是不扣分（就是清除得数）

m = GetMouseMsg(); //count++;不能放在这里

switch(m.uMsg)

{

case WM\_LBUTTONDOWN:

if(m.x>=50 && m.x<=100 && m.y>=250 && m.y<=300)

{

c = '1';

count++; //数：这是第几次按键

value = value\*10+1;

}

else if (m.x>=50+65 && m.x<=100+65 && m.y>=250 && m.y<=300)

{

c = '2';

count++; //数：这是第几次按键

value = value\*10+2;

}

//...其余类似

else if (m.x>=50+65\*3 && m.x<=100+65\*3 && m.y>=250 && m.y<=300)

{

draw = 0;

value = 9999; //为了跟真正结果为0区别开来；本次测试不会出现9999的真正结果

}

else if (m.x>=50+65\*3 && m.x<=100+65\*3 && m.y>=250+80 && m.y<=300+80) //出现小数点只有在tag==2情况

{

c = '.';

count++; //数：这是第几次按键

}

else if (m.x>=400 && m.x<=500 && m.y>=300 && m.y<=350)//确认键确认

{

flag = 0;

}

else if (m.x>=500 && m.x<=600 && m.y>=400 && m.y<=450) //结束键结束

{

draw = 0;

value = -9999; //为了跟真正结果为0区别开来；本次测试不会出现-9999的真正结果；-9999是标记数

}

else //点了界面其它空白处

{

draw = 0;

value = 9999; //为了跟真正结果为0区别开来；本次测试不会出现9999的真正结果；9999是标记数

}

if (flag == 1 && draw == 1) //按下什么数字，即时在屏幕上输出什么数字

{

settextcolor(GREEN);

settextstyle(100, 0, \_T("宋体"));

setbkmode (TRANSPARENT);

outtextxy(350+50\*(count-1),100,c );

}

else if (flag == 1 && draw == 0)

{

flag = 0;

}

break;

case WM\_RBUTTONDBLCLK:

exit(0); // 双击鼠标右键退出程序

}

}

return value;

}

8、结束界面：draw\_end

void draw\_end (int time\_start, int count) //绘制结束界面

{

time\_t time\_end; //结束训练的时间

time\_end = time(NULL);

int total\_time = time\_end - time\_start;

set\_background(); //设置背景

TCHAR x1[5],x2[5];

\_stprintf(x1, \_T("%d"), total\_time);

\_stprintf(x2, \_T("%d"), count);

settextcolor(RGB(0,200,200));

settextstyle(60, 0, \_T("楷体"));

outtextxy(10,100,"用时：");

outtextxy(400,100,x1);

outtextxy(500,100,"秒"); //其余类似

getch(); //按任意键结束

closegraph();

}

**三、测试及调试（测试方案、存在的问题及解决方法）**

* **测试方法**
* 写出计算流程大体框架，不断调试，自行测试四则运算是否成功
* 编写绘图界面的大体框架，通过调试调整显示
* 将计算流程嵌入绘图界面代码，通过调试查错修改
* 加入附加模块，先局部加入，调试通过后再实现完整的代码
* 自己模拟用户操作，想到各种可能出现的情况，逐一试验
* 将程序发给他人体验，参考他人的反馈建议改进代码
* 自己再模拟用户操作，改进代码
* **问题及解决方案**

1. 为实现良好界面交互，图形库是必不可少的。然而graphics.h**图形库**在TC才能直接使用；VS2010平台不能直接使用该图形库，那么该如何处理？

* 上网搜索使用VS编写含有图形的程序的方案，通过比较和选择，决定使用EasyX库（在EasyX官方网站下载EasyX库）。EasyX的优点： EasyX 库支持真彩色； EasyX 库多采用和 Windows 贴近的常量命名；含有其他扩充函数，例如鼠标、批操作等。通过参考学习EasyX Library，我掌握了该图形库的基本用法，并在我的代码中实现灵活运用。

1. 应该设计何种用户**输入方式**：使用键盘输入还是鼠标点选？

* 从用户体验的角度考虑：笔记本电脑用户的键盘的数字是一字横向排开的，输入时会很不方便；如果使用鼠标点击屏幕来选择数字，只需将屏幕显示做成适合用户习惯的模式，就可以达到良好用户体验的效果。我编写的时候，参考了计算器按钮设计来设计屏幕显示。另外，EasyX库对键盘输入的实时显示支持能力较低（有时出现栈溢出），我在多次失败的尝试后决定彻底放弃键盘输入的方式。

1. 一个好的交互软件需要在不同的环节变换界面，那么应该如何实现**界面的跳转**？

* 一开始我毫无头绪；参考了Easy Library的范例程序后，我得出了自己的解决方案：清屏→通过函数调用重绘背景→根据需要设置不同的显示参数。

1. 怎样才能**减法不出现负数**？

* 考虑到小学生减法教学中没有出现负数形式，本软件的减法运算也规避了出现负数的情况。我想到两种解决方案：一，检验随机生成的两个数的差的情况，如果差为负数，重新生成随机数，直到差不为负数；二，如果两个随机数的差为负数，则交换两个随机数的数值。显然，第二种方法的时间复杂度低，所以我采用了第二种方法。

1. **除法运算**可能出现的问题及如何确定除法的精准性？

* 在除法部分的代码编写中，我遇到了一些问题：int型的“/”是抹去小数部分的地板除，然而一开始没有注意到；浮点数的除法可能损失精度；随机生成的除数若为0，则系统会报错；让得数保留一位小数并不像“输出结果保留一位小数”那样只需“%.1f”就能实现……这些问题都是我在设计和不断调试的过程中发现的。不过，通过尝试，以上问题都得到了解决：或是使用了一些编写的小技巧，或者是利用自己定义的标识数等（具体实现详见主要代码设计）。

1. 怎样处理用户的**误操作**？如点击界面空白处，输错了数字。

* 如点击界面空白处，不会有状态改变；对于输错数字而用户想及时更改的情况，我设计了“X”按钮，用户点击后输入数字会清空，其余状态不变。具体代码实现的思路是在鼠标点击函数中设置特定的标识参数，当函数返回值返回的时候，可以进行特殊的操作。

1. 怎样才能实现**文字滚动**？

* 欢迎界面的滚动文字看起来简单，实际上需要每隔一段的时间局部重绘屏幕。具体实现需要暂时存储屏幕截图，然后实现位移，然后再放上截图（具体实现详见主要代码设计）。

1. 关于调试：在部分正确代码下有红色波浪线，因而**无法启动调试**？

* 经过上网查阅资料，我发现每次新建项目时要先修改一下项目属性：把Unicode改成多字节字符集；同时设置：代码生成——基本运行检查——默认值。这样运行时就不会报错。
* **存在的不足**

1. 得数为整数时，输入“.0”形式会被软件计为答错。如：9/3=3, 如果用户输入了3.0并确认，系统会扣分。这是因为我在设计代码时，设想的都是只有除法（非整除）运算才会出现小数点，而忽略了以上的情况。不过，考虑到用户输入习惯（整除情况下不会输入“.0”），这个问题不算大。
2. 该软件缺少：可视化计时器，返回菜单按钮以及多进程控制。这需要进一步完善。

**四、小结**

在这次课题设计中，我感触良多。学到了很多东西，也做出了自己比较满意的成果。在这个过程中，我也积累了一些经验：

* 拿到课题，先自己构想出大概的框架和主要的功能实现，不要顾虑某种语言是否能够实现。因为，总会有办法实现。
* 学会利用搜索引擎：输入关键词（有时候要变一下表述），从上往下浏览，找到可能合适的资料要认真看，并记下网址。
* 学会利用交流社区和论坛：有问题百度不到，就问啊！问啊！积极问啊！把问题表述出来，上社区论坛去问，总比孤苦无助好！
* 如果要系统学习某种语言或某种技巧，不要放弃书本；有时候，语言手册会比网上搜的小个技巧管用。
* 勤做笔记！学到新的技巧后，可以用PPT做简要笔记，或者是写一份技术日记。把相关链接和资源也记下来。
* 关于写代码：建议先用手写主要部分，手写代码不容易走神……还可以广泛集合各种好的代码。写完框架，思考各个所需功能是否实现了，再输入电脑。这次比较吃亏的地方就是，用电脑写好框架后，有模块需求发现遗漏了没有实现，却已经无法改了。
* 我的特点：愿意花很多时间思考自己的东西，数学思考可以给满分（误），还特别有耐心；但是，我却不太愿意花时间看别人写的东西，不愿意花时间学新东西……这得改变才行啊！