README

作业4 3-1

作业如何进行？

完成此次作业分为两个大部分，第一部分为读取命令行中各参数并保存；第二部分是使用保存的数据进行渲染，并且保存为图片。

在示例代码的技术上，新增了main.cpp(整个程序的入口)，ifs.h，ifs.cpp(用于保存变换矩阵，随机数和对读取的数据进行渲染)。

第一部分

命令行输入的格式为ifs -input fern.txt -points 50000 -iters 30 -size 400 -output fern.tga，需要用变量记录5个参数。

-input 表示读入的文本文件

-points 表示需要进行变换的点数

-iters 表示每个点迭代的次数

-size 表示输出图片的尺寸

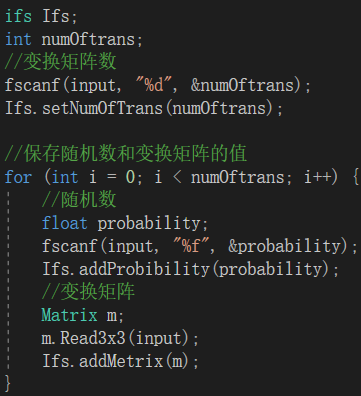
-output 表示输出图片的命名和类型

读取input值之后，要进行数据解析，文件中包含的数据有：传入的变换矩阵的个数，每个变换矩阵使用的概率和具体矩阵行列分布情况。

变换矩阵个数为整型，使用fscanf读入“%d”类型；变换矩阵使用的概率为浮点数，使用fscanf读入“%f”类型；矩阵数据使用示例代码中的Read3x3，可以自动识别文本中3x3的数据，并且转成矩阵形式。

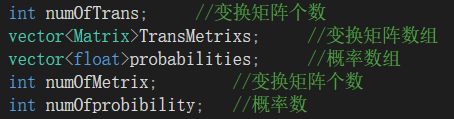
因为有多组随机数和矩阵数据，用for循环遍历即可，遍历次数即为读入的变换矩阵个数。

对应的代码为：



对读入的三个数据，都保存到了类ifs中。

类ifs的私有成员变量为



setNumOfTrans表示设置变换矩阵个数numOfTrans。

addMetrix表示push\_back matrix变量到TransMetrixs中。

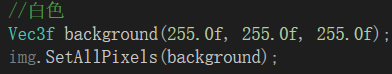
addProbibility表示push\_back float变量到probabilities中。

其他变量，在第二部分使用到。

第二部分

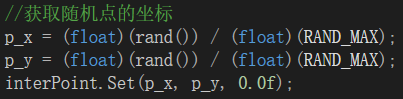
实现ifs中的函数render，该函数需要使用到的变量为类image，保存的变换矩阵和每个变换矩阵发生的概率数，迭代次数，进行迭代的点数。

首先设置图片的底色为白色，使用image成员函数SetAllPixels，代码表示为：



用vec3f保存随机生成点的坐标。

因为是2d平面所以z轴坐标为0，并且x，y坐标的范围为0~1，所以使用随机数生成的代码表示为：



即随机生成x，y之后，使用vec3f的set函数设置坐标数值。

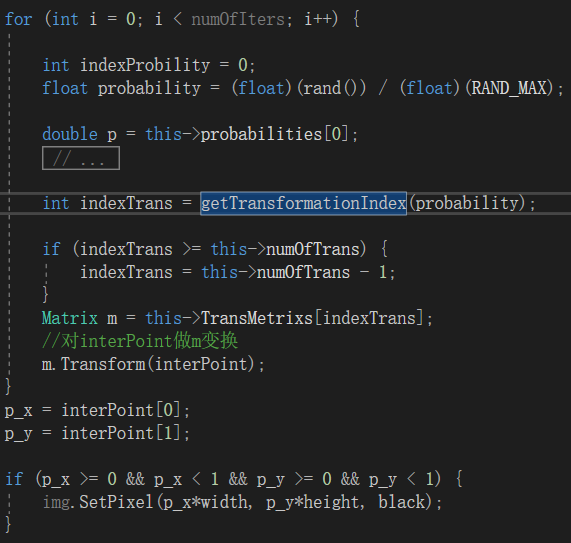
以上是生成一个点的语句，根据作业要求要对传入的points个点进行随机生成坐标，即在外层加上一层for循环。

在随机生成点坐标之后，根据迭代函数系统的原理，每个点必须通过iters次变换之后的坐标才是该点最后的坐标，这时候才能将像素数据写入图片中。

具体实现过程为:

再加入一层for循环，随机生成0~1之间的随机数，并且根据这个随机数判断是使用哪一个变换矩阵进行变换，重复iters次之后，此时点的坐标才是最终该点的坐标。

具体代码实现为：



其中的getTransformationIndex表示返回变换矩阵的下标

具体实现的思路为，

因为传入的每个变换矩阵发生概率的和为1，那么将整个变换矩阵的概率数想象成它们分布在一个长度为1的数轴上，此时有4个变换对应的概率为x1,x2,x3,x4，那么：

x1发生的范围在0~x1，变换矩阵下标为0；

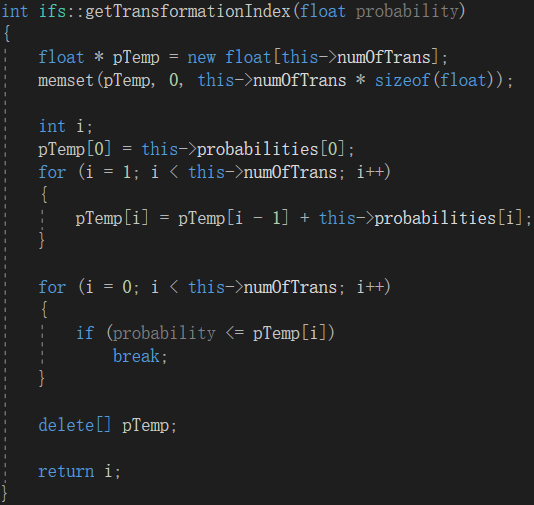
x2发生的范围在x1~x1+x2，变换矩阵下标为1；

X3发生的范围在x1+x2~x1+x2+x3，变换矩阵下标为2；

X3发生的范围在x1+x2+x3~1，变换矩阵下标为3；

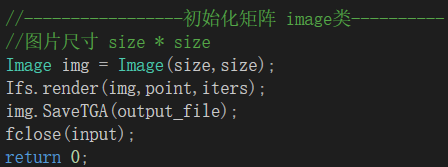
使用随机生成的数进行判断它落在哪个范围就取对应的下标即可。

对应的代码为：



完成渲染之后，使用image类中的SaveTGA，保存为.tga类型文件，输出文件名即使用命令行传入的output变量。

Main函数中具体代码实现为：



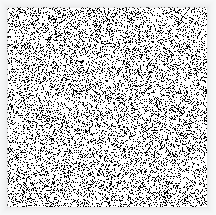
实习中遇到的困难和解决方案

在运行时，总是会发现vector数组越界的情况，之后在怀疑可能发生越界的位置使用assert进行捕获异常，发现是因为在求变换数组下标时，会发生越界的情况。

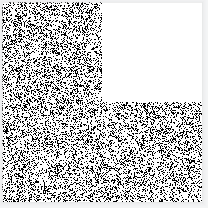
解决方法是对于所有越界的下标，都取最后一个变换数组的下标。

实习结果展示

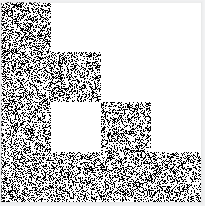
ifs -input sierpinski\_triangle.txt -points 10000 -iters 0 -size 200 -output sierpinski\_triangle\_0.tga



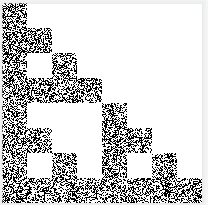
ifs -input sierpinski\_triangle.txt -points 10000 -iters 1 -size 200 -output sierpinski\_triangle\_1.tga



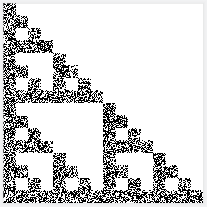
ifs -input sierpinski\_triangle.txt -points 10000 -iters 2 -size 200 -output sierpinski\_triangle\_2.tga



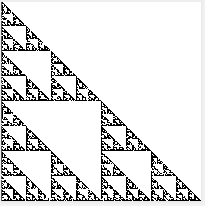
ifs -input sierpinski\_triangle.txt -points 10000 -iters 3 -size 200 -output sierpinski\_triangle\_3.tga



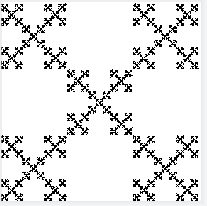
ifs -input sierpinski\_triangle.txt -points 10000 -iters 4 -size 200 -output sierpinski\_triangle\_4.tga



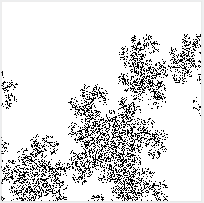
ifs -input sierpinski\_triangle.txt -points 10000 -iters 30 -size 200 -output sierpinski\_triangle.tga



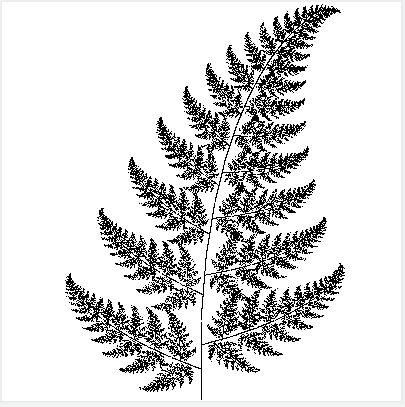
ifs -input giant\_x.txt -points 10000 -iters 30 -size 200 -output giant\_x.tga



ifs -input dragon.txt -points 10000 -iters 100 -size 200 -output dragon.tga



ifs -input fern.txt -points 50000 -iters 30 -size 400 -output fern.tga



总结

通过这次作业，对网课上讲述的迭代函数系统的相关知识加深了理解，将理论应用于实践中。并且进一步加深了变换的知识。