

# READ ME

姓名：周宁

班级：111171

学号：20171004140

## 一、实习思路：

### 1. 如何编译和运行代码

①在 vs 中创建一个工作项目，将作业“Assignment1.files”文件夹下的对应的.h 和.c 文件复制到项目当中，然后将.c 文件改为.cpp 文件即可，如：将文件“image.C”复制到工作目录当中，然后改为“image.cpp”。

### 2. 编写 C ++类 IFS

这个类对应着上课讲的迭代函数系统 IFS，具体实现可以参考那章的 ppt 进行相对应的实现。

①编写数据字段。按照作业上的提示，我设计了三个字段，分别为变换数、变换数组、概率数组，可以参考图 1。

```
protected:
    int n; //变换的数量
    Matrix * m_transformMatrixs; //变换数组
    float * m_probability; //概率数组
```

图 1

②实现函数 `void ReadDescription(char* input_file)`，该函数的功能是从传入的文件“input\_file”当中读取变换的数量、变换数组以及概率数组。

③实现构造函数以及虚函数。这个实现起来比较简单，在构造函数当中只需要调用②里面写的 `ReadDescription` 函数就行，而析构函数里面做释放内存的工作即可。

④实现函数 `void Render(Image* image, int numOfPoints, int numOfiters)`，该函数的功能是根据点的数量以及迭代的次数，对 image 进行处理，最终得到我们想要的图像，本次实习的精华也在这个实现当中。我采取的方法是，设置黑和白两个颜色，调用 image 的 `SetAllPixels` 的函数，先将整个图像设置为白色（如图 2）。

```
//设置颜色
Vec3f black(0.0f, 0.0f, 0.0f), white(255.0f, 255.0f, 255.0f);
image->SetAllPixels(white); //先将底色设置成白色
```

图 2

接着我使用 C++ 当中的随机数，随机得到 numOfPoints 数量的点，这里的“随机”指点的位置是随机的，具体可以参考图 3，其中的 width 和 height 指图片的宽和高。

```
vector<Vec2f> points;    //随机点
for (int i = 0; i < numOfPoints; ++i) {
    //srand((unsigned)time(NULL));
    float x = rand() % width;
    float y = rand() % height;
    points.push_back(Vec2f(x, y));
}
```

图 3

然后写一个两层的 for 循环对随机点和迭代次数进行遍历。在第一层 for 循环对迭代次数进行遍历，这样可以减少迭代次数为 0 时的重复，在第二层 for 循环对上诉得到的随机点进行按概率随机的变换。其中对于按照概率随机，我采取的方法是按照概率进行相对应的分段，比如概率为 0.3、0.3、0.4，那我将“1”按照概率对应的长度划分为 3 个区间，0 到 0.3 为第一个区间，0.3 到 0.6 为第二个区间，0.6 到 1 为第三个区间。然后取一个随机值将随机值 ran 除以“RAND\_MAX”，将其转化到 0~1，然后判断 ran 在那个区间，即可按照概率得到变换数组，具体的实现可以参考图

```
//进行核心的渲染算法
//将迭代次数写在外面可以减少迭代为0的时候的重复
for (int times = 0; times < numOfiters; ++times)
{
    //对点进行迭代
    for (int j = 0; j < numOfPoints; ++j)
    {
        int index = 0;
        //进行概率选择
        double ran = (double)(rand()) / (double)(RAND_MAX);
        double p = m_probability[0];

        while (p < ran && index < n) {
            /*if (index >= n-1) {
                cout << m_probability[0] << endl;
                cout << p << endl;
                cout << ran << endl;
            }*/
            assert(index < n);
            p += m_probability[++index];
        }
    }
}
```

图 4

```
Matrix transMatrix = m_transformMatrixs[index];
Vec2f t(points[j].x() / (float)(width), points[j].y() / (float)(height));
transMatrix.Transform(t); //有直接对vec2f进行转化的函数所以不需要三个点
points[j].Set(t.x() * width, t.y() * height);
```

图 5

最终再对随机点进行遍历，然后根据点的 x、y 坐标设置 image 对应位置的像素值为黑，就得到最终的转化图像，可以参考图 6。

```
//对点进行设置
for (int i = 0; i < numOfPoints; ++i) {
    if(!(points[i].x() >=0 && points[i].x() < width))
        continue;
    if(!(points[i].y() >=0 && points[i].y() < height))
        continue;
    image->SetPixel(points[i].x(), points[i].y(), black);
}
```

图 6

### 3. 编写主函数。

①再本题当中主函数文件中已经给出，在“demo0.CPP”文件当中，所以不需要太过复杂写，我在里面只是简单的创建 Image 实例，再从文件中读取 IFS 描述，然后创建 IFS 实例，再调用 IFS 的函数 **Render**，然后保存图像，可以参考图 7。

```
Image* image = new Image(width, height);
IFS ifs(input_file);
//ifs.ReadDescription(input_file);
ifs.Render(image, points, iters);
image->SaveTGA(output_file);//进行存文件
```

图 7

## 二、实习中遇到的困难和解决方案：

1. 不知道 tga 文件是什么文件，并且在 Windows 上并没有自带的查看 tga 文件的软件。

解决方法：百度 tga 文件后，选择下载了 TGA Viewer 软件进行相对应的查看。

2. 不知道如何按照概率进行随机。

解决方案：参考了网上的建议，采取上面提到的分段的方法进行概率性随机。

## 三、实习结果展示

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

结果图如图 8 所示，其中的灰色的外框是 TGA Viewer 软件背景为灰色，没有办法进行设置。

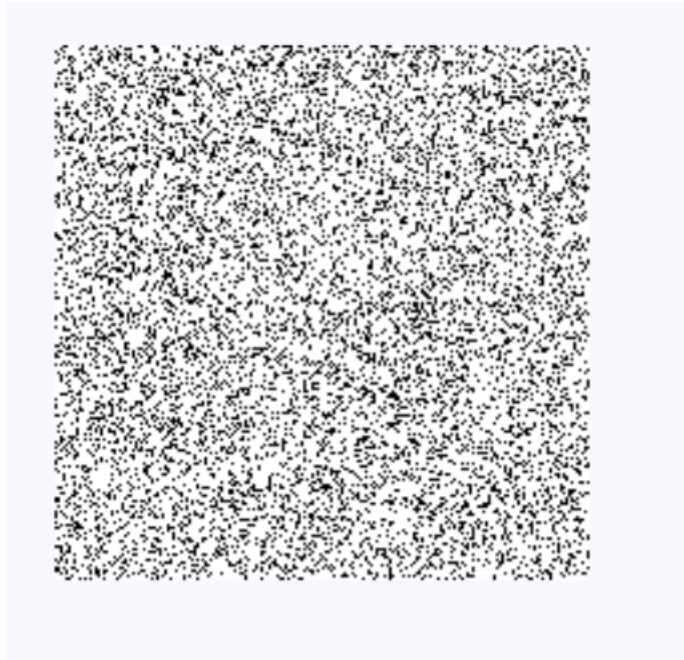


图 8

2. `ifs -input sierpinski_triangle.txt -points 10000 -iters 1 -size 200 -output sierpinski_triangle_1.tga`  
结果图如图 9 所示。

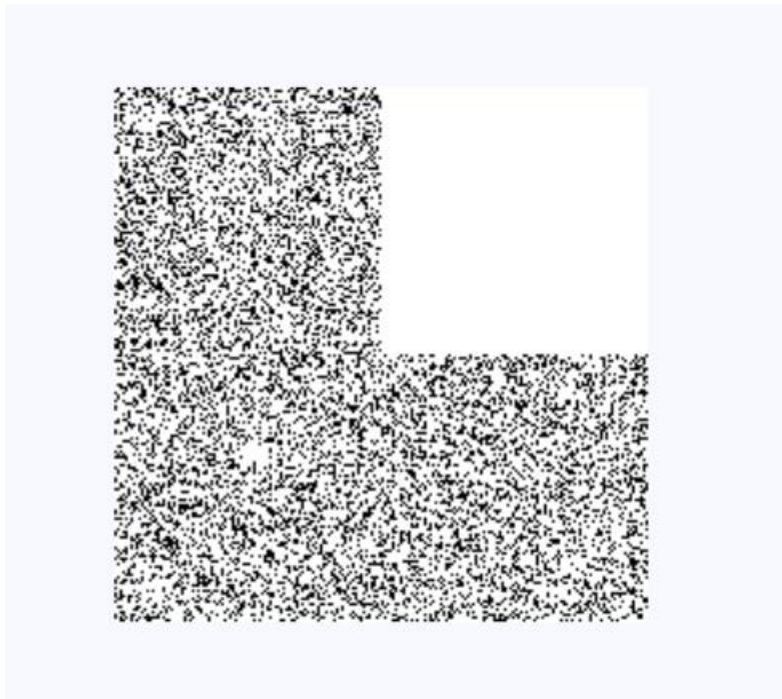


图 9

3. `ifs -input sierpinski_triangle.txt -points 10000 -iters 2 -size 200 -output sierpinski_triangle_2.tga`

结果图如图 10 所示。

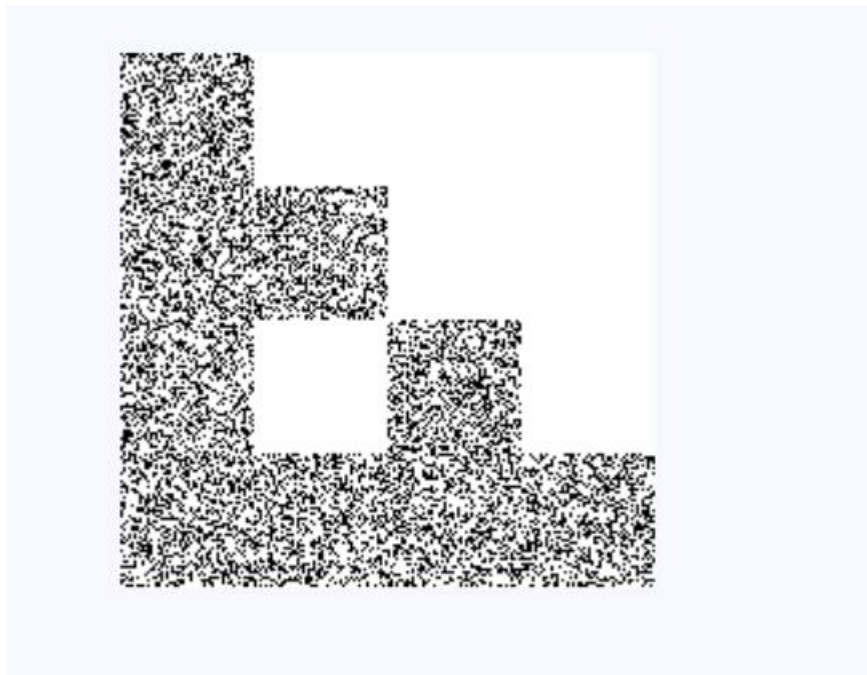


图 10

4. `ifs -input sierpinski_triangle.txt -points 10000 -iters 3 -size 200 -output sierpinski_triangle_3.tga`

结果图如图 11 所示。

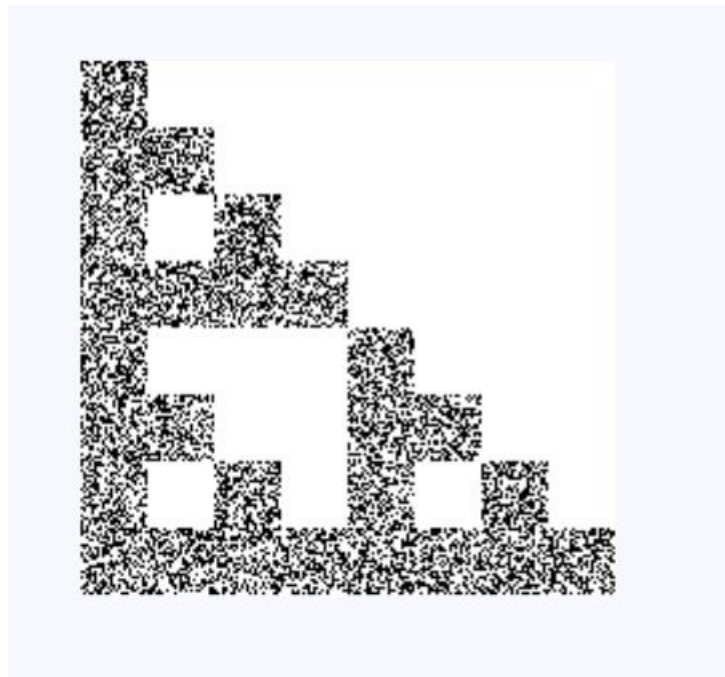


图 11

5. `ifs -input sierpinski_triangle.txt -points 10000 -iters 4 -size 200 -output sierpinski_triangle_4.tga`  
结果图如图 12 所示。

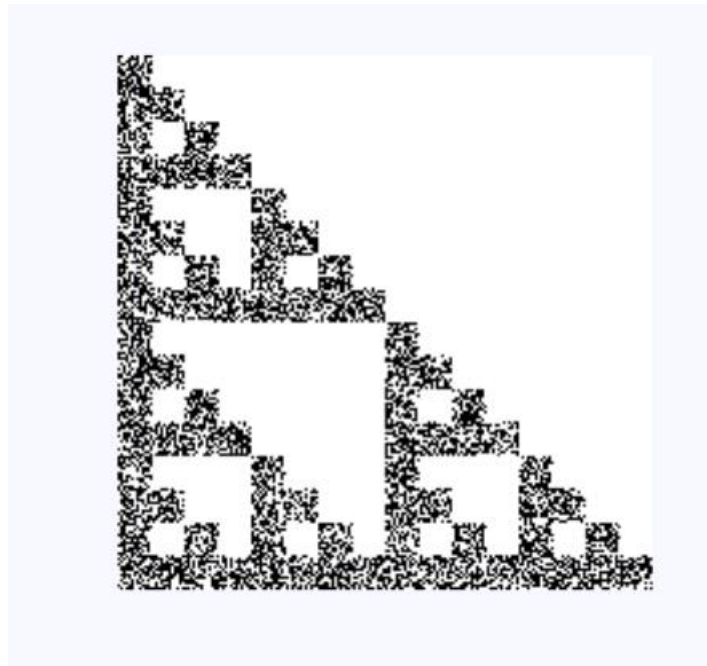


图 12

6. `ifs -input sierpinski_triangle.txt -points 10000 -iters 30 -size 200 -output sierpinski_triangle.tga`  
结果图如图 13 所示。

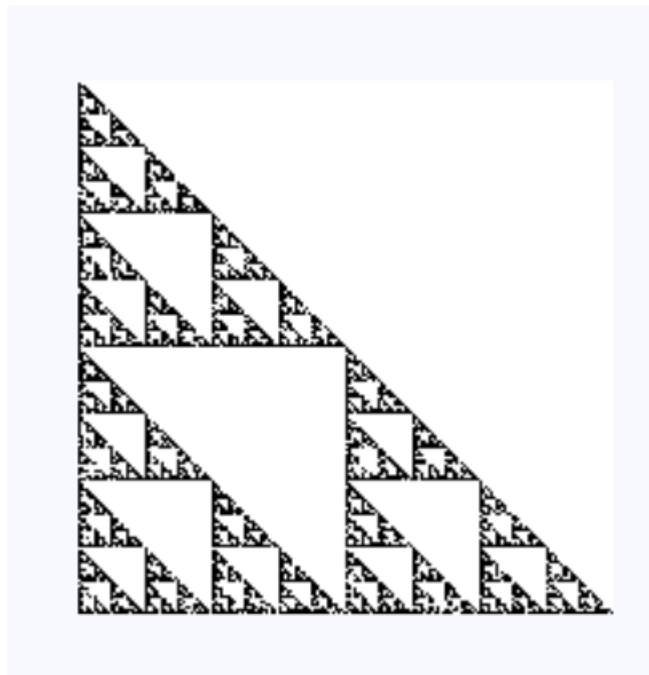


图 13



7. `ifs -input fern.txt -points 50000 -iters 30 -size 400 -output fern.tga`  
结果图如图 14 所示。

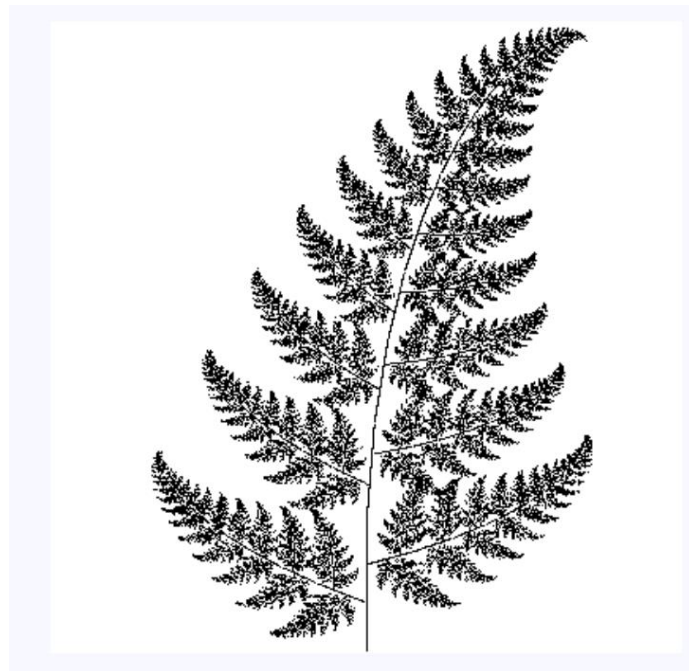


图 14

8. `ifs -input giant_x.txt -points 10000 -iters 30 -size 200 -output giant_x.tga`  
结果图如图 15 所示。

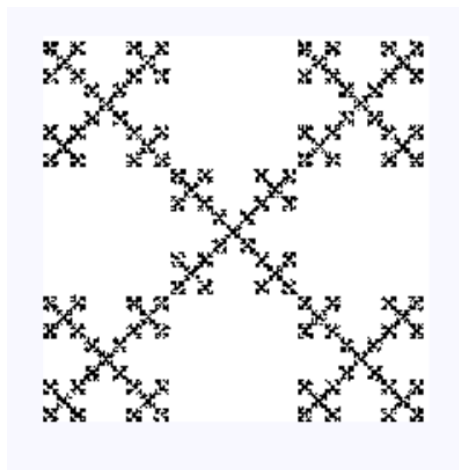


图 15

9. `ifs -input dragon.txt -points 10000 -iters 100 -size 200 -output dragon.tga`  
结果图如图 16 所示。发现图并没有全部显示为上课的那种龙的形状，我个人判断为“dragon.txt”中给的变换应该给少了，只有两种变换但是课上课 ppt 当中至少有做平移、斜着平移、旋转三种变换，所以应该是变换给少了，所以导致只能变成这样。

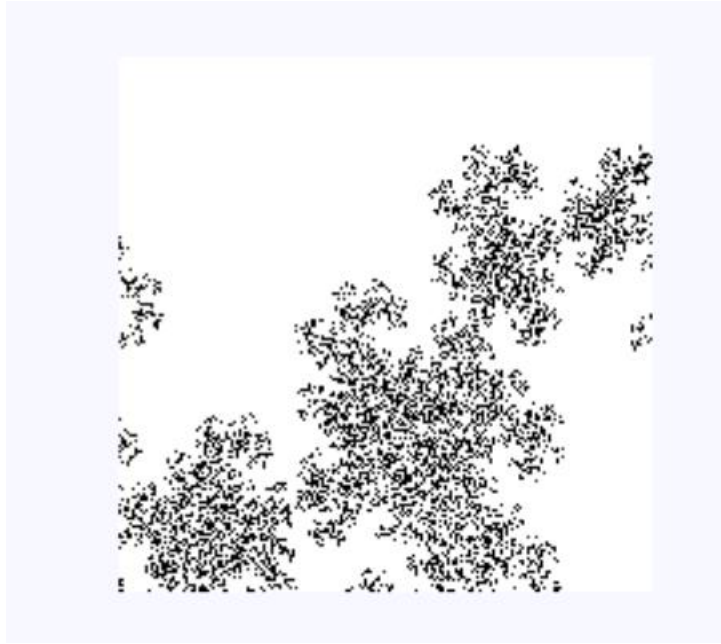


图 16

#### 四、实习总结

本次实习内容为上课所讲的奇趣变换，内容并不是特别难，看懂算法之后进行编写就比较简单，总体来说，收获还行，对变换方面的知识又多了一点。