高级程序设计训练实验5

—实验报告

学期：2016-2017-2学期

报告人：杨宇

学号：16281237

班级：计算机1608

编制日期：2017年4月12日

目录

[一、 概述 3](#_Toc479877448)

[二、 实验目的 3](#_Toc479877449)

[1.初步了解计算机图形学算法 3](#_Toc479877450)

[2.掌握使用 MS VISIO绘制流程图 3](#_Toc479877451)

[三、 实验内容 3](#_Toc479877452)

[1．直线段的生成算法 3](#_Toc479877453)

[2．多边形结构体的定义 4](#_Toc479877454)

[3．多边形的旋转的实现 5](#_Toc479877455)

[4．流程图的绘制 6](#_Toc479877456)

[四、 实验体会 6](#_Toc479877457)

[五、 实验结果 6](#_Toc479877458)

1. 概述

本次的实验五与实验三和实验四有很大的不同。实验五和计算机图形学联系精密。在老师事先实现的一个简单的图形界面上绘制几何图形。

1. 实验目的
2. 初步了解计算机图形学算法

计算机图形学是计算机科学的重要分支。现如今，GUI界面相对于CLS界面更加的user-friendly，受到了广大用户的欢迎。以WINDOWS系统为代表的具有图形用户界面的操作系统，早已取代了命令行界面。通过计算机图形学来实现GUI和图形绘制，值得计算机科学的学习者去研究。

1. 掌握使用 MS VISIO绘制流程图

在程序设计的过程中，为了把复杂的程序结构清晰的表述，绘制流程图是一个很好的解决方案。MS office Professional中包含的Visio就是一款常用的绘制流程图的软件。掌握流程图的绘制很有必要。

1. 实验内容
2. 直线段的生成算法

本次的实验中，我所设计的直线段生成算法，通过给定的线段上两个端点，计算Δx来判断斜率k是否存在。如果斜率k不存在，则只需要调用已经实现的绘制垂直直线的函数DrawVertiline()绘制直线。如果Δx存在，则通过x = x + 1, y = y + k来找到直线上的点。但是因为屏幕上的点是离散的，而直线上的点是连续的，斜率k也是个浮点数。为了使找到的点相对准确，每次绘制时，需要对y值进行取舍。为了使所取的值最接近实际的数值。我对y从浮点数到整数的转化过程中，采取了四舍五入。

代码段：

void DrawLine(WINDOW\* pWindow, MYPOINT a, MYPOINT b,char chBrush)

{

if (IsValidPoint(pWindow, a) == 0 || IsValidPoint(pWindow, b) == 0)

{

return;

}

if (a.x > b.x)

{

MYPOINT temp = a;

a = b;

b = temp;

}

if (b.x == a.x)

{

if (a.y < b.y)

{

int temp = a.y;

a.y = b.y;

b.y = temp;

}

DrawVertiLine(pWindow,a,a.y - b.y,chBrush);

}

else

{

double k = (b.y - a.y)\*1.0 / (b.x - a.x);

MYPOINT l;

l.x = a.x;

l.y = a.y;

double fy = l.y;

while (l.x <= b.x)

{

l.y = (int)(fy + 0.5);

DrawAt(pWindow, l, chBrush);

fy = fy + k;

l.x++;

}

}

}

1. 多边形结构体的定义

为了在屏幕上描述多边形，我定义了一个结构体，用于记录多边形的顶点个数和顶点。通过Initduobianxin()函数初始化多边形，并为数组动态分配空间。而对数组中点的标记，则需要单独的按照顺时针（逆时针）依次设置。

struct duobianxin

{

int number = 0;

MYPOINT \* points;

};

typedef struct duobianxin DBX;

int Initduobianxin(DBX\* a,int n) {

a->number = n;

a->points = (MYPOINT\*)malloc(n\*sizeof(MYPOINT));

if (a->points == NULL)

return 0;

}

//没有读入点，只是分配了空间

return n;

}

1. 多边形的旋转的实现

多边形的旋转问题本质上是一个数学问题。为了围绕多边形的几何中心旋转。我通过向量的加法来找到多边形的几何中心（类似于重心的求法）。

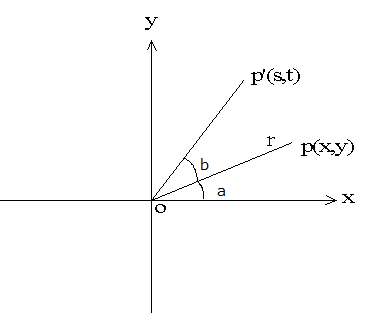


图5-1

s = r cos(a + b) = r cos(a)cos(b) – r sin(a)sin(b)   (1.1)

t = r sin(a + b) = r sin(a)cos(b) + r cos(a) sin(b)  (1.2)

其中 x = r cos(a)  , y = r sin(a)

代入(1.1), (1.2) ,

s = x cos(b) – y sin(b) (1.3)

t = x sin(b) + y cos(b) (1.4)

用行列式表达如下：

http://hi.csdn.net/attachment/201004/23/0_1272000306M0q2.gif

通过上面的数学推导，可以方便的求出围绕几何中心点旋转一定角度后得到的点的坐标。

值得注意的是，通过上式的计算所得到的结果并不是一个整点。为了使屏幕上画的点和直线尽可能的接近，也需要对点坐标进行四舍五入。

1. 流程图的绘制

流程图的绘制使用了visio，通过将左侧程序框拖动组合，连接程序框图之间的连线可以方便的绘制出程序流程图。

1. 实验体会
2. 对于大型的软件工程开发，对整个工程的分解和思路整理非常重要。
3. 代码的注释要尽可能的详细，尤其是在多人共同开发时，代码的可读性非常重要。
4. 计算机图形学和解析几何、线性代数等结合的非常紧密。在编程的同时要抓住数学知识。
5. 程序流程图和函数调用关系可以将整个代码可视化，有助于快速的理解代码。
6. 实验结果

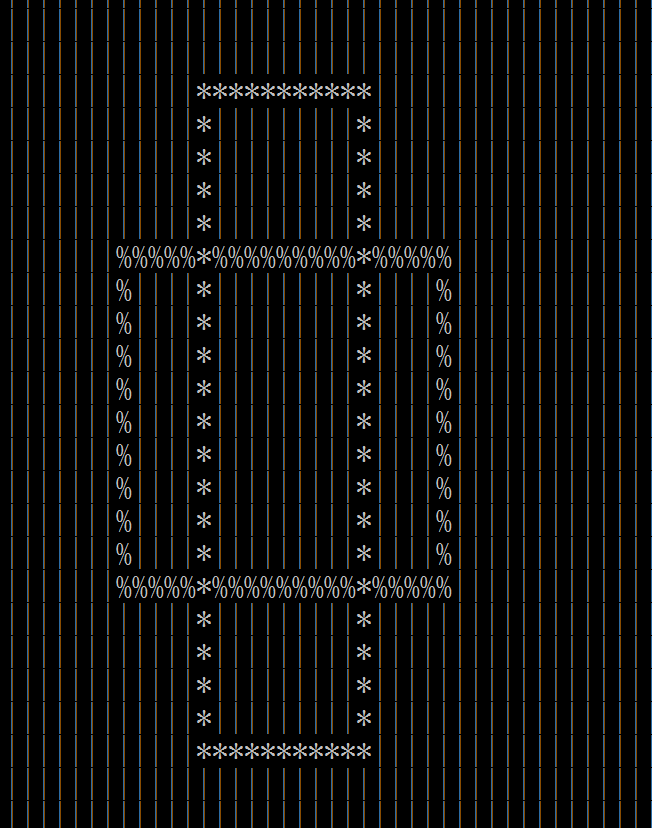


图5-2

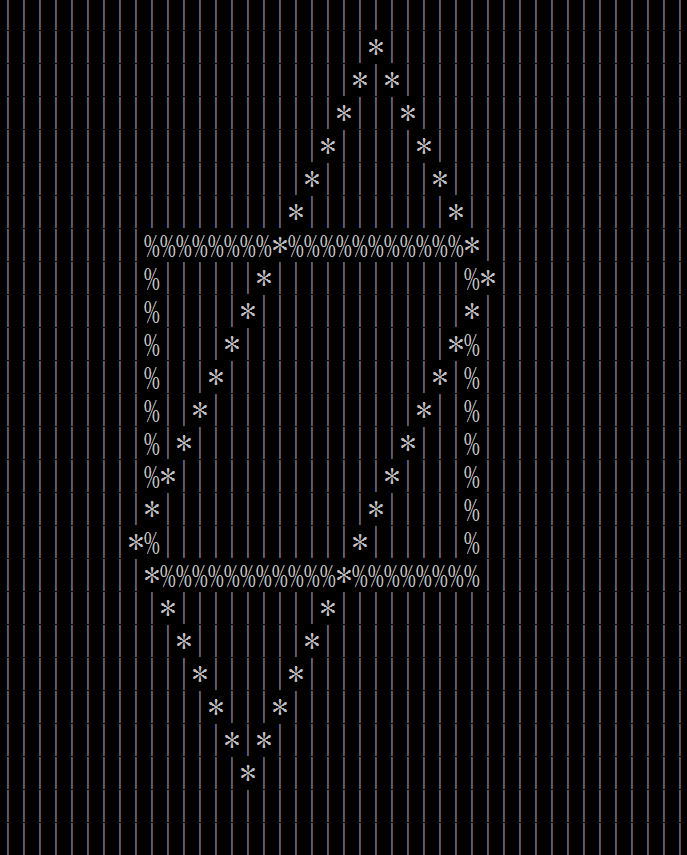


图5-3



图5-4

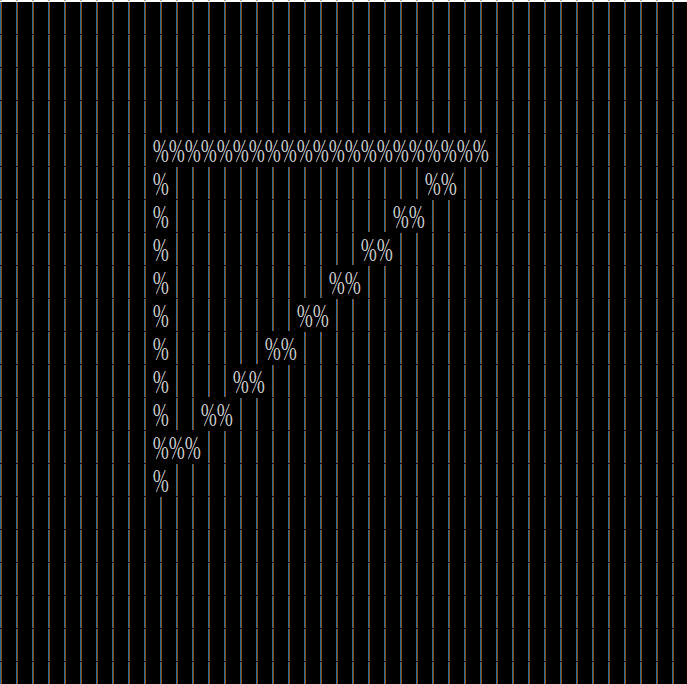


图5-5

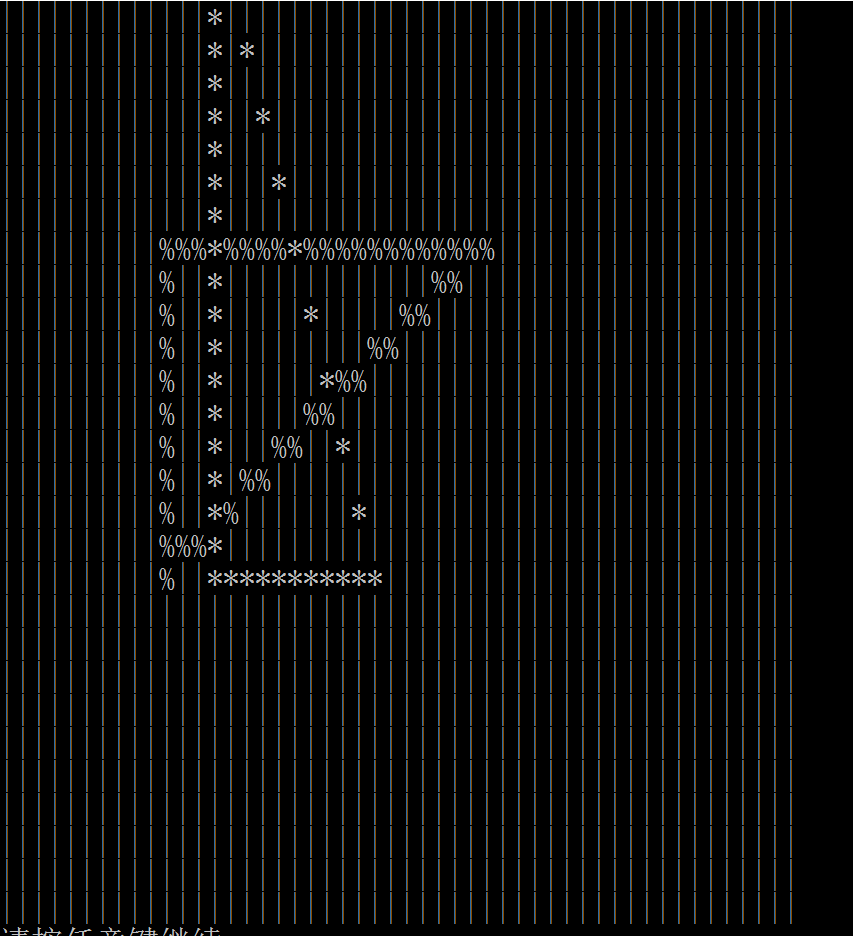


图5-6

