**计算机图形学实验**

**姓 名：白文强**

**学 号：20191060064**

**专 业：计算机科学与技术**

**教 师：钱文华**

实验三 中点算法生成直线段实验

时间：2020年10月16日

地点：信息学院2202机房

1. 实验内容

熟悉OPENGL，通过中点算法生成直线段

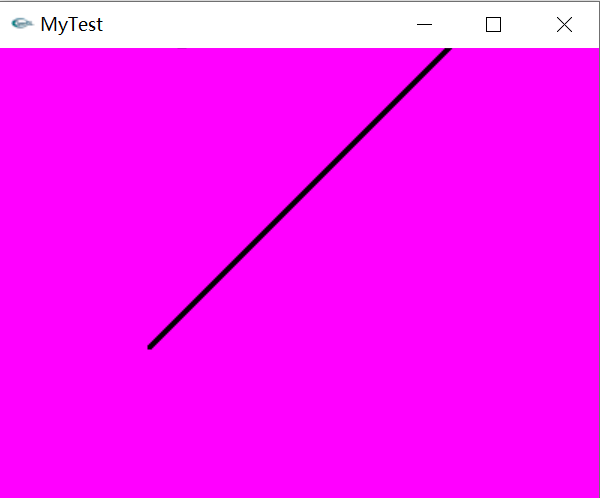
2、实验目的

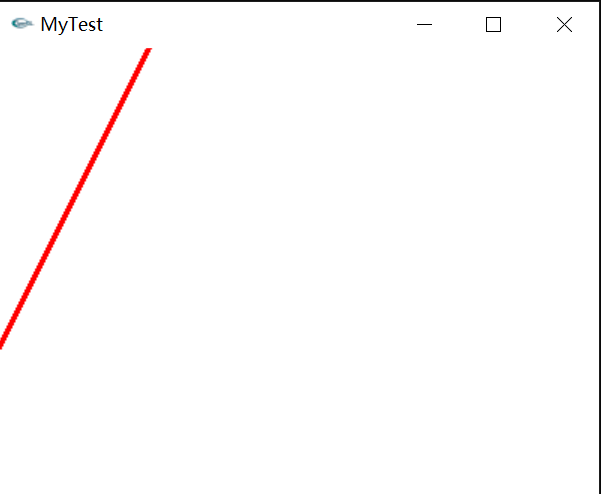
安装OPENGL，能编写代码运行，参考课本代码。

3、实验代码

1. #include <windows.h>
2. #include <GL/glut.h>
3. #include <stdlib.h>
4. #include <math.h>
5. **void** init(**void**)
6. {
7. glClearColor(1.0, 0.0, 1.0, 1.0);
8. gluOrtho2D(0.0, 400.0, 0.0, 300.0);
9. }
11. **void** setPixel(**int** x, **int** y)
12. {
13. glBegin(GL\_POINTS);
14. glVertex2i(x, y);
15. glEnd();
16. }
17. **void** lineBresenHam(**int** x0, **int** y0, **int** xEnd, **int** yEnd)
18. {
19. **int** x, y;
20. **int** dx = xEnd - x0;
21. **int** dy = yEnd - y0;
22. //每次都从最左边的点画
23. **if**(x0 > xEnd)
24. {
25. x = xEnd;
26. y = yEnd;
27. xEnd = x0;
28. yEnd = y0;
29. }
30. **else**
31. {
32. x = x0;
33. y = y0;
34. }
36. **float** k = **float**(dy) / dx;       //斜率
37. **float** d = 0;
38. setPixel(x, y);
39. **if**(fabs(dx) >= fabs(dy))         //斜率绝对值大于零小于1
40. {
41. **int** steps = fabs(dx);       //画点次数
42. **for**(**int** i = 0; i< steps; i++)
43. {
44. x++;
45. d += k;
46. **if**(k>=0)
47. {
48. **if**(d > 1)
49. {
50. d = d - 1;
51. }
52. **if**(d > 0.5)
53. {
54. y++;
55. }
56. }
57. **else**
58. {
59. **if**(d < -1)
60. {
61. d = d + 1;
62. }
63. **if**(d < -0.5)
64. {
65. y--;
66. }
67. }
68. setPixel(x, y);
69. }
70. }
71. **else**
72. {
73. **int** steps = fabs(dy);
74. **for**(**int** i = 0; i < steps; i++)
75. {
76. d += 1 / k;
77. **if**(k>=0)
78. {
79. y++;
80. **if**(d > 1)
81. {
82. d = d - 1;
83. }
84. **if**(d > 0.5)
85. {
86. x++;
87. }
88. }
89. **else**
90. {
91. y--;
92. **if**(d < -1)
93. {
94. d = d + 1;
95. }
96. **if**(d < -0.5)
97. {
98. x++;
99. }
100. }
101. setPixel(x, y);
102. }
103. }
104. }
105. **void** lineSegement(**void**)
106. {
107. glClear(GL\_COLOR\_BUFFER\_BIT);   // 清空显示窗口
108. glColor3f(0.0, 0.0, 0.0);       // 指定前景色
109. glPointSize(3.0);               //点的大小
110. glBegin(GL\_POINTS);
111. lineBresenHam(100, 100, 300,300 );
112. glEnd();
113. glFlush();
114. }
115. **int** main(**int** argc, **char**\*\* argv)
116. {
117. glutInit(&argc, argv);
118. glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);
119. glutInitWindowPosition(100, 100);
120. glutInitWindowSize(400, 300);
121. glutCreateWindow("MyTest");
122. init();
123. glutDisplayFunc(lineSegement);
124. glutMainLoop();
125. **return** 0;
126. }

4、实验结果





5、实验总结

　　写算法需要考虑多种情况，适应不同的需求，对该算法而言，要考虑斜率的正负以及斜率绝对值是否大于一的情况。考虑到程序的实现效率，推导出误差项的递推式以减少运算次数，提高运行效率