**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**

实验十 Bezier曲线生成

时间：2020年12月18日

地点：信息学院2202机房

1. 实验内容

使用opengl，教材P323，Bezier曲线生成程序。

1. 实验目的

采用Bezier函数验证曲线生成。

3、实验代码

1. #include <windows.h>
2. #include <GL/glut.h>
3. #include <stdlib.h>
4. #include <math.h>
6. /\*  Set initial size of the display window.  \*/
7. GLsizei winWidth = 600, winHeight = 600;
9. /\*  Set size of world-coordinate clipping window.  \*/
10. GLfloat xwcMin = -50.0, xwcMax = 50.0;
11. GLfloat ywcMin = -50.0, ywcMax = 50.0;
13. **class** wcPt3D {
14. **public**:
15. GLfloat x, y, z;
16. };
18. **void** init (**void**)
19. {
20. /\*  Set color of display window to white.  \*/
21. glClearColor (1.0, 1.0, 1.0, 0.0);
22. }
24. **void** plotPoint (wcPt3D bezCurvePt)
25. {
26. glBegin (GL\_POINTS);
27. glVertex2f (bezCurvePt.x, bezCurvePt.y);
28. glEnd ( );
29. }
31. /\*  Compute binomial coefficients C for given value of n.  \*/
32. **void** binomialCoeffs (GLint n, GLint \* C)
33. {
34. GLint k, j;
36. **for** (k = 0;  k <= n;  k++) {
37. /\*  Compute n!/(k!(n - k)!).  \*/
38. C [k] = 1;
39. **for** (j = n;  j >= k + 1;  j--)
40. C [k] \*= j;
41. **for** (j = n - k;  j >= 2;  j--)
42. C [k] /= j;
43. }
44. }
46. **void** computeBezPt (GLfloat u, wcPt3D \* bezPt, GLint nCtrlPts,
47. wcPt3D \* ctrlPts, GLint \* C)
48. {
49. GLint k, n = nCtrlPts - 1;
50. GLfloat bezBlendFcn;
52. bezPt->x = bezPt->y = bezPt->z = 0.0;
54. /\*  Compute blending functions and blend control points. \*/
55. **for** (k = 0; k < nCtrlPts; k++) {
56. bezBlendFcn = C [k] \* pow (u, k) \* pow (1 - u, n - k);
57. bezPt->x += ctrlPts [k].x \* bezBlendFcn;
58. bezPt->y += ctrlPts [k].y \* bezBlendFcn;
59. bezPt->z += ctrlPts [k].z \* bezBlendFcn;
60. }
61. }
63. **void** bezier (wcPt3D \* ctrlPts, GLint nCtrlPts, GLint nBezCurvePts)
64. {
65. wcPt3D bezCurvePt;
66. GLfloat u;
67. GLint \*C, k;
69. /\*  Allocate space for binomial coefficients  \*/
70. C = **new** GLint [nCtrlPts];
72. binomialCoeffs (nCtrlPts - 1, C);
73. **for** (k = 0;  k <= nBezCurvePts;  k++) {
74. u = GLfloat (k) / GLfloat (nBezCurvePts);
75. computeBezPt (u, &bezCurvePt, nCtrlPts, ctrlPts, C);
76. plotPoint (bezCurvePt);
77. }
78. **delete** [ ] C;
79. }
81. **void** displayFcn (**void**)
82. {
83. /\*  Set example number of control points and number of
84. \*  curve positions to be plotted along the Bezier curve.
85. \*/
86. GLint nCtrlPts = 4, nBezCurvePts = 1000;
88. wcPt3D ctrlPts [4] = { {-10.0, -40.0, 0.0}, {-10.0, 50.0, 0.0},
89. {10.0, -100.0, 0.0}, {40.0, 40.0, 0.0} };
91. glClear (GL\_COLOR\_BUFFER\_BIT);   //  Clear display window.
92. glPointSize (4);
93. glColor3f (1.0, 0.0, 0.0);      //  Set point color to red.
95. bezier (ctrlPts, nCtrlPts, nBezCurvePts);
96. glFlush ( );
97. }
99. **void** winReshapeFcn (GLint newWidth, GLint newHeight)
100. {
101. /\*  Maintain an aspect ratio of 1.0.  \*/
102. glViewport (0, 0, newHeight, newHeight);
104. glMatrixMode (GL\_PROJECTION);
105. glLoadIdentity ( );
107. gluOrtho2D (xwcMin, xwcMax, ywcMin, ywcMax);
109. glClear (GL\_COLOR\_BUFFER\_BIT);
110. }
112. **int** main (**int** argc, **char**\*\* argv)
113. {
114. glutInit (&argc, argv);
115. glutInitDisplayMode (GLUT\_SINGLE | GLUT\_RGB);
116. glutInitWindowPosition (50, 50);
117. glutInitWindowSize (winWidth, winHeight);
118. glutCreateWindow ("Bezier Curve");
120. init ( );
121. glutDisplayFunc (displayFcn);
122. glutReshapeFunc (winReshapeFcn);
124. glutMainLoop ( );
125. **return** 0;
126. }

4、实验结果

