实验四 填充算法实验

时间：2022年4月13日

地点：信息学院机房2202

1、实验内容

1.根据教材P66，实现六边形的填充

2.用扫描线填充算法实现多边形的填充

2、实验目的

验证扫描线填充算法，指定任意的多边形边数，填充多边形

3、实验代码

实现六边形的填充：

|  |
| --- |
| #include <windows.h>  #include <GL/glut.h>  #include <stdlib.h>  #include <math.h>  const double TWO\_PI=6.2831853;  /\* Initial display-window size \*/  GLsizei winWidth = 400,winHeight = 400;  GLuint regHex;  class screenPt{  private:  GLint x,y;  public:  /\* Default Constructor:initializes coordinate position to (0.0).\*/  screenPt(){  x=y=0;  }  void setCoords(GLint xCoord,GLint yCoord){  x=xCoord;  y=yCoord;  }  GLint getx()const{  return x;  }  GLint gety()const{  return y;  }  };  static void init(void)  {  screenPt hexVertex,circCtr;  GLdouble theta;  GLint k;  /\*Set circle center coordinates.\*/  circCtr.setCoords(winWidth/2,winHeight/2);  glClearColor(1.0,1.0,1.0,0.0);//Display-window color = white.  /\*Set up a display list for a red regular hexagon.  \*Vertices for the hexagon are six equally spaced  \*points around the circumference of a circle.  \*/  regHex = glGenLists(1); //Get an identifier for the display list.  glNewList(regHex,GL\_COMPILE);  glColor3f(1.0,0.0,0.0); // Set fill color for haxagon to red.  glBegin(GL\_POLYGON);  for(k=0;k<6;k++)  {  theta = TWO\_PI\*k/6.0;  hexVertex.setCoords(circCtr.getx()+150\*cos(theta),circCtr.gety()+150\*sin(theta));  glVertex2i(hexVertex.getx(),hexVertex.gety());  }  glEnd();  glEndList();  }  void regHexagon(void)  {  glClear(GL\_COLOR\_BUFFER\_BIT);  glCallList(regHex);  glFlush();  }  void winReshapeFcn(int newWidth,int newHeight)  {  glMatrixMode(GL\_PROJECTION);  glLoadIdentity();  gluOrtho2D(0.0,(GLdouble)newWidth,0.0,(GLdouble)newHeight);  glClear(GL\_COLOR\_BUFFER\_BIT);  }  int main (int argc,char\*\* argv)  {  glutInit(&argc,argv);  glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);  glutInitWindowPosition(100,100);  glutInitWindowSize(winWidth,winHeight);  glutCreateWindow("Reshape\_Function & Display\_List Example");  init();  glutDisplayFunc(regHexagon);  glutReshapeFunc(winReshapeFcn);  glutMainLoop();  } |

任意多边形的填充：

|  |
| --- |
| #include <windows.h>  #ifdef \_\_APPLE\_\_  #include <GLUT/glut.h>  #else  #include <GL/glut.h>  #endif  #include <math.h>  #include <stdlib.h>  const double TWO\_PI = 6.28318;  /\*Initial display-window size\*/  GLsizei winWidth = 400,winHeight = 400;  GLuint regHex;  class screenPT{  private:  GLint x,y;  public:  screenPT(){  x=y=0;  }  void setCoords(GLint xCoord,GLint yCoord){  x = xCoord;  y = yCoord;  }  GLint getx() const {  return x;  }  GLint gety() const {  return y;  }  };  static void init (void)  {  screenPT hexVertex, circCtr;  GLdouble theta;  GLint k;  /\*set circle center coordinates\*/  circCtr.setCoords(winWidth/2,winHeight/2);  glClearColor(1.0,1.0,1.0,0.0);//display-window color=white  regHex = glGenLists(1);  glNewList(regHex,GL\_COMPILE);  glColor3f(1.0,0.0,0.0);  glBegin(GL\_POLYGON);  theta=TWO\_PI/6.0;  hexVertex.setCoords(circCtr.getx()+100\*cos(theta\*2),circCtr.gety()+150\*sin(theta\*2));  glVertex2i(hexVertex.getx(),hexVertex.gety());  hexVertex.setCoords(circCtr.getx()+140\*cos(theta\*2),circCtr.gety()+160\*sin(theta\*2));  glVertex2i(hexVertex.getx(),hexVertex.gety());  hexVertex.setCoords(circCtr.getx()+160\*cos(theta\*4),circCtr.gety()+170\*sin(theta\*3));  glVertex2i(hexVertex.getx(),hexVertex.gety());  hexVertex.setCoords(circCtr.getx()+180\*cos(theta\*4),circCtr.gety()+180\*sin(theta\*3));  glVertex2i(hexVertex.getx(),hexVertex.gety());  hexVertex.setCoords(circCtr.getx()+200\*cos(theta\*5),circCtr.gety()+190\*sin(theta\*4));  glVertex2i(hexVertex.getx(),hexVertex.gety());  hexVertex.setCoords(circCtr.getx()+250\*cos(theta\*5),circCtr.gety()+200\*sin(theta\*4));  glVertex2i(hexVertex.getx(),hexVertex.gety());  glEnd();  glEndList();  }  void regHexagen(void)  {  glClear(GL\_COLOR\_BUFFER\_BIT);  glCallList(regHex);  glFlush();  }  void winReshapeFcn(int newWidth,int newHeight)  {  glMatrixMode(GL\_PROJECTION);  glLoadIdentity();  gluOrtho2D(0.0, (GLdouble)newWidth,0.0,(GLdouble)newHeight);  glClear(GL\_COLOR\_BUFFER\_BIT);  }  int main(int argc,char\*\* argv)  {  glutInit(&argc, argv);  glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);  glutInitWindowPosition(100,100);  glutInitWindowSize(winWidth,winHeight);  glutCreateWindow("Reshape-Function & Display-List Example");  init();  glutDisplayFunc(regHexagen);  glutReshapeFunc(winReshapeFcn);  glutMainLoop();  return 0;  } |

4、实验总结

本次实验主要进行了多边形的填充实现，从书本上的规则六边形填充到任意六边形的填充实现。