实验十 着色及建模实验

时间：2022年5月18日

地点：信息学院机房2202

1、实验内容

使用opengl，片元着色器着色，P523

使用opengl，书写教材P541页代码，进行颜色编码建模显示。

2、实验目的

调用函数完成颜色编码建模实验。

3、实验代码

片元器着色

|  |
| --- |
| #include<GL/glut.h>  varying vec3 light,view;  uniform sampler2D textureID;  float height(vec3 color){  float avg = (color.r+color.g)/2.0;  return mix(avg,.5,.985);  }  vec3 modNormal(vec3 point){  vec2 d0 = vec2(0,0.001);  vec2 d1 = vec2(-0.000866,-0.0005);  vec2 d2 = vec2(0.000866,-0.0005);  vec2 p0=point +d0;  vec2 p1=point +d1;  vec2 p2==point +d2;  float h0 = height(vec3(texture2D(textureID,p0)));  float h1 = height(vec3(texture2D(textureID,p1)));  float h2 = height(vec3(texture2D(textureID,p2)));  vec3 v0=vec3(d0,h0);  vec3 v1=vec3(d1,h1);  vec3 v2=vec3(d2,h2);  return normalize(vec3(cross(v1-v0,v2-v0)));  }  void main(){  vec4 base = texture2D(textureID,,gl\_TexCoord[0].st);  vec3 bump = modNormal(gl\_TexCoord[0].st);  vec4 color = gl\_LightSource[0].ambient\*base;  float NdotL = max(dot(bump,light),0.0);  color +=NdotL\*(gl\_LightSource[0].diffuse\*base);  gl\_FragColor =color;  } |

建模实验

|  |
| --- |
| #include<windows.h>  #include <GL/glut.h>  //设置一个窗口的大小  GLsizei winWidth = 500, winHeight = 500;  GLfloat xComplexMin = -2.00, xComplexMax = 0.50;  GLfloat yComplexMin = -1.25, yComplexMax = 1.25;  GLfloat complexWidth = xComplexMax - xComplexMin;  GLfloat complexHeight = yComplexMax - yComplexMin;  class complexNum  {  public:  GLfloat x, y;  };  struct color  {  GLfloat r, g, b;  };  void init(void)  {  glClearColor(1.0, 1.0, 1.0, 0.0);  }  void plotpoint(complexNum z)  {  glBegin(GL\_POINTS);  glVertex2f(z.x, z.y);  glEnd();  }  complexNum complexSquare(complexNum z)  {  complexNum zSquare;  zSquare.x = z.x \* z.x - z.y \* z.y;  zSquare.y = 2 \* z.x \* z.y;  return zSquare;  }  GLint mandelSqTransf(complexNum z0, GLint maxIter)  {  complexNum z = z0;  GLint count = 0;  while ((z.x \* z.x + z.y \* z.y <= 4.0) && (count < maxIter))  {  z = complexSquare(z);  z.x += z0.x;  z.y += z0.y;  count++;  }  return count;  }  void mandelbrot(GLint nx, GLint ny, GLint maxIter)  {  complexNum z, zIncr;  color ptColor;  GLint iterCount;  zIncr.x = complexWidth / GLfloat(nx);  zIncr.y = complexHeight / GLfloat(ny);  for (z.x = xComplexMin; z.x < xComplexMax; z.x += zIncr.x)  for (z.y = yComplexMin; z.y < yComplexMax; z.y += zIncr.y)  {  iterCount = mandelSqTransf(z, maxIter);  if (iterCount >= maxIter)  ptColor.r = ptColor.g = ptColor.b = 0.0;  else if (iterCount > (maxIter / 10))  {  ptColor.r = 1.0;  ptColor.g = 0.5;  ptColor.b = 0.0;  }  else if (iterCount > (maxIter / 10))  {  ptColor.r = 1.0;  ptColor.g = ptColor.b = 0.0;  }  else if (iterCount > (maxIter / 20))  {  ptColor.b = 0.5;  ptColor.r = ptColor.g = 0.0;  }  else if (iterCount > (maxIter / 100))  {  ptColor.r = ptColor.b = 0.0;  ptColor.g = 0.3;  }  else  {  ptColor.r = 0.0;  ptColor.g = ptColor.b = 1.0;  }  glColor3f(ptColor.r, ptColor.g, ptColor.b);  plotpoint(z);  }  }  void displayFcn(void)  {  GLint nx = 1000, ny = 1000, maxIter = 1000;  glClear(GL\_COLOR\_BUFFER\_BIT);  mandelbrot(nx, ny, maxIter);  glFlush();  }  void winReshapeFcn(GLint newWidth, GLint newHeight)  {  glViewport(0, 0, newHeight, newHeight);  glMatrixMode(GL\_PROJECTION);  glLoadIdentity();  gluOrtho2D(xComplexMin, xComplexMax, yComplexMin, yComplexMax);  glClear(GL\_COLOR\_BUFFER\_BIT);  }  int main(int argc, char \*\*argv)  {  glutInit(&argc, argv);  glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);  glutInitWindowPosition(50, 50);  glutInitWindowSize(winWidth, winHeight);  glutCreateWindow("Mandelbrot 集");  init();  glutDisplayFunc(displayFcn);  glutReshapeFunc(winReshapeFcn);  glutMainLoop();  } |

4、实验总结

由于书本代码可能存在问题，片元器着色实验代码并不能成功运行。