**作业要求：**

采用OpenGL模拟吹气球：

1.在YOZ平面上放置若干简单物体（立方体，球体，茶壶等）；

2.在固定位置放置一个以上灯光，有泛光、高光、聚光、遮挡、阴影效果；

3.按Crtl吹气球，按Alt释放，气球在空中自然漂浮，按住鼠标左键气球破灭；

4.按键盘上下左右摄像机前后左右移动；

5.透视投影，程序窗口大小改变物体不失真；

6.设计文档和程序注释清楚完整；

设计说明书

**1. 在YOZ平面上放置若干简单物体（立方体，球体，茶壶等）**

为了观察方便，在xoz平面画了一个平面，在上面放置了一盒茶壶，一个球，一个立方体

具体实现代码如下：

glColor3f(0.5f, 0.5f, 0.5f);

glBegin(GL\_QUADS);

glVertex3f(-10.0f, -5.0f, -20.0f);

glVertex3f(-10.0f, -5.0f, 0.0f);

glVertex3f(10.0f, -5.0f, 0.0f);

glVertex3f(10.0f, -5.0f, -20.0f);

glEnd(); //画一个平面

glPushMatrix(); //压入堆栈

glTranslatef(0.0f, -3.0f, -10.0f);

glColor3ub(0, 255, 255);

glutSolidTeapot(1.0); //茶壶程序

glPopMatrix(); //释放堆栈

glPushMatrix();

glTranslatef(5.0f,-3.0f,-5.0f);

glColor3ub(255,0, 0);

glutSolidSphere(1.0f, 100.0, 100.0); //球

glPopMatrix();

glPushMatrix();

glTranslatef(5.0f,-3.0f,-10.0f);

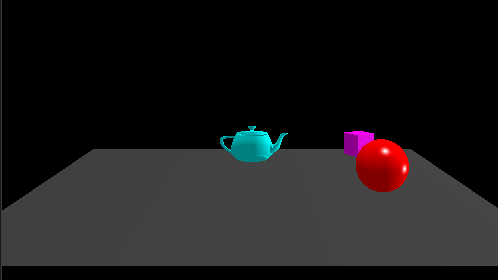
glRotatef(30, 0.0f, 1.0f, 0.0f);

glColor3ub(255, 0, 255);

glutSolidCube(1); //立方体

glPopMatrix();

实现效果：



**2. 在固定位置放置一个以上灯光，有泛光、高光、聚光、遮挡、阴影效果；**

在函数void SetupRC(void)中设置灯光，在代码前设置参数；

具体参数如下：

GLfloat lightPos[] = { 100.0f,0.0f,0.0f,1.0f };

GLfloat lightDirection[] = { -1.0,0.0,0.0 };

GLfloat lightPos1[] = {0.0f,100.0f,0.0f,1.0f };

GLfloat lightDirection1[] = { 0.0,-1.0,0.0 }；

GLfloat specular[] = { 1.0f,1.0f,1.0f,1.0f };

GLfloat specrof[] = { 1.0f,1.0f,1.0f,1.0f };

GLfloat ambientLight[] = { 0.5f,0.5f,0.5f,1.0f };

灯光设置如下：

（1）首先开启深度检测，并设定逆时针为正，对背面消隐；

glEnable(GL\_DEPTH\_TEST);

//当前像素前面是否有别的像素，如果别的像素挡道了它，那它就不会绘制

glFrontFace(GL\_CCW); //逆时针为正

glEnable(GL\_CULL\_FACE); //背面不显示

（2）设置一个全局环境光；

glEnable(GL\_LIGHTING); //开灯

glLightModelfv(GL\_LIGHT\_MODEL\_AMBIENT, ambientLight);

//全局环境光，提供轻微的环境光，以便可以看到物体；整个场景的环境光的RGBA的强度

（3）设置两个聚光灯LIGHT0和LIGHT1，一个方向从右向左，一个从上到下；

glLightfv(GL\_LIGHT0, GL\_DIFFUSE, ambientLight);

//光源中的散射光强度（光仅由漫反射和镜面反射组成）

glLightfv(GL\_LIGHT0, GL\_SPECULAR, specular); //光源中的镜面反射光强度

glLightfv(GL\_LIGHT0, GL\_POSITION, lightPos); //指定光源位置

glLightfv(GL\_LIGHT0, GL\_SPOT\_DIRECTION, lightDirection);

// 聚光灯主轴方向 spot direction

glLightf(GL\_LIGHT0, GL\_SPOT\_CUTOFF, 90.0f);

//指定光源的最大散布角 （创建聚光灯，v表示参数是向量（数组））

glLightf(GL\_LIGHT0, GL\_SPOT\_EXPONENT, 500.0f); //指定聚焦光源指数

glEnable(GL\_LIGHT0); //启动0号光源

glLightfv(GL\_LIGHT1, GL\_DIFFUSE, ambientLight);

//光源中的散射光强度（光仅由漫反射和镜面反射组成）

glLightfv(GL\_LIGHT1, GL\_SPECULAR, specular); //光源中的镜面反射光强度

glLightfv(GL\_LIGHT1, GL\_POSITION, lightPos1); //指定光源位置

glLightfv(GL\_LIGHT1, GL\_SPOT\_DIRECTION, lightDirection1);

// 聚光灯主轴方向 spot direction

glLightf(GL\_LIGHT1, GL\_SPOT\_CUTOFF, 90.0f);

//指定光源的最大散布角 （创建聚光灯，v表示参数是向量（数组））

glLightf(GL\_LIGHT1, GL\_SPOT\_EXPONENT, 500.0f); //指定聚焦光源指数

glEnable(GL\_LIGHT1);

glEnable(GL\_COLOR\_MATERIAL);

//使用颜色材质，颜色追踪。激活光照的情况下用glColor函数给物体上色

glColorMaterial(GL\_FRONT, GL\_AMBIENT\_AND\_DIFFUSE);

//来决定对物体的正面还是反面，对环境光、镜面光还是漫射光进行颜色跟踪

glMaterialfv(GL\_FRONT, GL\_SPECULAR, specrof); //材质的镜面反射颜色

glMateriali(GL\_FRONT, GL\_SHININESS, 128); //镜面反射指数

glDepthFunc(GL\_LEQUAL);

/\*

glDepthFunc(GL\_LEQUAL)深度小或相等的时候也渲染。

glDepthFunc(GL\_LESS)深度小的时候才渲染

\*/

glClearColor(0.0f, 0.50f, 0.0f, 1.0f); //背景为绿色

**3.按Crtl吹气球，按Alt释放，气球在空中自然漂浮，按住鼠标左键气球破灭；**

Crtl和Alt键属于特殊键，在函数void KeyBoards(unsigned char key, int x, int y)中实现方式如下：

int mod;

mod = glutGetModifiers();

if (mod == (GLUT\_ACTIVE\_CTRL))

{

gaobin111 += 1;

gaobin222 = 0;

position = -3.0f;

size = 0.0f;

}

但是因为不知为啥原因，crtl和alt键的键位始终无法实现效果，所以，吹气球和释放气球改用Z键和A键；

具体实现方式，当Z键按下，参数gaobin111加1，

void KeyBoards(unsigned char key, int x, int y)

{

switch (key)

{

case 65 :

case 97 : //x

gaobin222 += 1;

gaobin111 = 0;

position = -3.0f;

size = 0.0f;

break;

case 90:

case 122: //z

gaobin111 += 1;

gaobin222 = 0;

position = -3.0f;

size = 0.0f;

break;

case 27 :

exit(0);

}

}

在绘制函数中，当gaobin111大于等于2时，绘制一个黄色的气球，气球随着时间增大，函数

void TimerFunc(int value)

{

glutPostRedisplay();

glutTimerFunc(10, TimerFunc, 1); //持续调用

}

实现随着时间增加，气球变大的过程，每10ms刷新一次，每次气球直径增加0.005；最大半径为2；

if (gaobin111 >= 1)

{

glPushMatrix();

glTranslatef(0.0f, -3.0f, -5.0f);

glColor3ub(255, 255, 0);

size+= 0.005f;

if (size >= 2.0)size = 2.0;

glutSolidSphere(size, 50.0, 50.0); //吹一个黄色气球

glPopMatrix();

}

}

同样的，在吹完气球后，按A键参数gaobin222加1，当gaobin222大于等于1时，气球随着时间变化，每隔着10ms位置刷新，每次沿着Y轴向上移动0.01；

if (gaobin222 >= 1)

{

glPushMatrix();

size = 2;

glTranslatef(0.0f, position,-5.0f);

position += 0.01f;

if (position >= 8.0)position = 8.0;

glColor3ub(255, 255, 0);

glutSolidSphere(size, 50.0, 50.0); //释放黄色球

glPopMatrix();

}

在实现鼠标左键气球破时，用函数

void OnMouse(int button, int state, int x, int y)

{

if (button == GLUT\_LEFT\_BUTTON)

{

gaobin111 = 0;

gaobin222 = 0;

}

}

获取鼠标点击的信息，让绘制气球的参数都变为0，此时气球消失；

**4.按键盘上下左右摄像机前后左右移动；**

首先获取键盘方向键的信息：

void SpecialKey(GLint key, GLint x, GLint z)

{

switch (key)

{

case GLUT\_KEY\_LEFT:

angle -= 0.01f;

orientMe(angle);

break;

case GLUT\_KEY\_RIGHT:

angle += 0.01f;

orientMe(angle);

break;

case GLUT\_KEY\_UP:

moveMeFlat(1);

break;

case GLUT\_KEY\_DOWN:

moveMeFlat(-1);

break;

}

}

通过上下键来实现摄像机前后移动，下面是实现前后移动的函数

void moveMeFlat(int i) {

x = x + i \* (lx)\*0.1;

z = z + i \* (lz)\*0.1;

glLoadIdentity();

gluLookAt(x, y, z,x + lx, y + ly, z + lz,0.0f, 1.0f, 0.0f);

}

通过左右键来实现摄像机向左向右旋转实现左右的视野，下面是实现左右旋转的函数

void orientMe(float ang)

{

lx = sin(ang);

lz = -cos(ang);

glLoadIdentity();

gluLookAt(x, y, z,x + lx, y + ly, z + lz,0.0f, 1.0f, 0.0f);

}

**5.透视投影，程序窗口大小改变物体不失真；**

通过透视投影，限定观察范围，来实现透视效果，窗口大小可以调整，但观察物体的比例大小不变可实现物体不失真，具体在下面函数中实现；

void ChangeSize(int w, int h)

{

if (h == 0)h = 1;

fElect = w / h;

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glViewport(0, 0, w, h);

gluPerspective(45, fElect, 1, 1000);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity()；

gluLookAt(x, y, z,x + lx, y + ly, z + lz,0.0f, 1.0f, 0.0f);

}