void Specular\_color(float mat\_specular,float light\_specular,int \*lightpos,int \*eyepos,float \*\*\* object\_color,float gloss)

//计算物体上每一个点的镜面反射颜色

{

float specular\_color=mat\_specular\*light\_specular;

int \*eye\_pot=new int[3];

float \*ref\_eye\_pot=new float[3];

int \*vec\_pot\_light=new int[3];

int \*nor\_vec=new int[3];

float length=0;

float coefficient=0;

for(int k =0;k<10;k++)

{

for(int j =0;j<10;j++)

{

//计算x=10的那个平面

//计算该店法线点乘该点到光源的单位向量，如果该值小于0，那么就用0代替此次结果

vec\_pot\_light[0]=lightpos[0]-9;

vec\_pot\_light[1]=lightpos[1]-k;

vec\_pot\_light[2]=lightpos[2]-j;

eye\_pot[0]=eyepos[0]-9;

eye\_pot[1]=eyepos[1]-k;

eye\_pot[2]=eyepos[2]-j;

//eye\_pot现在存储的是视点到该点的方向向量，但是不是单位向量，法向量是单位向量为nor\_vec

nor\_vec[0]=1;

nor\_vec[1]=0;

nor\_vec[2]=0;

length=sqrt(pow(eye\_pot[0],2)+pow(eye\_pot[1],2)+pow(eye\_pot[2],2));

//for(int i =0;i<3;i++) //将eyepot变为单位向量

// eye\_pot[i]/=length;

coefficient= eye\_pot[0]\*nor\_vec[0]+eye\_pot[1]\*nor\_vec[1]+eye\_pot[2]\*nor\_vec[2];

coefficient/=length;

length=sqrt(pow(vec\_pot\_light[0],2)+pow(vec\_pot\_light[1],2)+pow(vec\_pot\_light[2],2));

coefficient/=length;

for (int i =0;i<3;i++) //计算得到变换之后的eyepot的反射向量，计算参考的公式http://www.cnblogs.com/graphics/archive/2013/02/21/2920627.html

ref\_eye\_pot[i]=eye\_pot[i]-2\*coefficient\*nor\_vec[i];

coefficient=0;

for (int i =0;i<3;i++) //计算出了眼睛到该点的单位向量的反射向量和该点到光源的向量的乘积

coefficient+=ref\_eye\_pot[i]\*vec\_pot\_light[i];

coefficient/=length;

if (coefficient<0)

coefficient=0;

object\_color[9][k][j]+=pow(coefficient,gloss)\*specular\_color;

//计算y=10的那个平面

//计算该店法线点乘该点到光源的单位向量，如果该值小于0，那么就用0代替此次结果

vec\_pot\_light[0]=lightpos[0]-k;

vec\_pot\_light[1]=lightpos[1]-9;

vec\_pot\_light[2]=lightpos[2]-j;

eye\_pot[0]=eyepos[0]-k;

eye\_pot[1]=eyepos[1]-9;

eye\_pot[2]=eyepos[2]-j;

//eye\_pot现在存储的是视点到该点的方向向量，但是不是单位向量，法向量是单位向量为nor\_vec

nor\_vec[0]=0;

nor\_vec[1]=1;

nor\_vec[2]=0;

length=sqrt(pow(eye\_pot[0],2)+pow(eye\_pot[1],2)+pow(eye\_pot[2],2));

coefficient= eye\_pot[0]\*nor\_vec[0]+eye\_pot[1]\*nor\_vec[1]+eye\_pot[2]\*nor\_vec[2];

coefficient/=length;

length=sqrt(pow(vec\_pot\_light[0],2)+pow(vec\_pot\_light[1],2)+pow(vec\_pot\_light[2],2));

for (int i =0;i<3;i++) //计算得到变换之后的eyepot的反射向量，计算参考的公式http://www.cnblogs.com/graphics/archive/2013/02/21/2920627.html

ref\_eye\_pot[i]=eye\_pot[i]-2\*coefficient\*nor\_vec[i];

coefficient=0;

for (int i =0;i<3;i++) //计算出了眼睛到该点的单位向量的反射向量和该点到光源的向量的乘积

coefficient+=ref\_eye\_pot[i]\*vec\_pot\_light[i];

coefficient/=length;

if (coefficient<0)

coefficient=0;

object\_color[k][9][j]+=pow(coefficient,gloss)\*specular\_color;

//计算z=10的那个平面

//计算该店法线点乘该点到光源的单位向量，如果该值小于0，那么就用0代替此次结果

vec\_pot\_light[0]=lightpos[0]-k;

vec\_pot\_light[1]=lightpos[1]-j;

vec\_pot\_light[2]=lightpos[2]-9;

eye\_pot[0]=eyepos[0]-k;

eye\_pot[1]=eyepos[1]-j;

eye\_pot[2]=eyepos[2]-9;

//eye\_pot现在存储的是视点到该点的方向向量，但是不是单位向量，法向量是单位向量为nor\_vec

nor\_vec[0]=0;

nor\_vec[1]=0;

nor\_vec[2]=1;

length=sqrt(pow(eye\_pot[0],2)+pow(eye\_pot[1],2)+pow(eye\_pot[2],2));

coefficient= eye\_pot[0]\*nor\_vec[0]+eye\_pot[1]\*nor\_vec[1]+eye\_pot[2]\*nor\_vec[2];

coefficient/=length;

length=sqrt(pow(vec\_pot\_light[0],2)+pow(vec\_pot\_light[1],2)+pow(vec\_pot\_light[2],2));

for (int i =0;i<3;i++) //计算得到变换之后的eyepot的反射向量，计算参考的公式http://www.cnblogs.com/graphics/archive/2013/02/21/2920627.html

ref\_eye\_pot[i]=eye\_pot[i]-2\*coefficient\*nor\_vec[i];

coefficient=0;

for (int i =0;i<3;i++) //计算出了眼睛到该点的单位向量的反射向量和该点到光源的向量的乘积

coefficient+=ref\_eye\_pot[i]\*vec\_pot\_light[i];

coefficient/=length;

if (coefficient<0)

coefficient=0;

object\_color[k][j][9]+=pow(coefficient,gloss)\*specular\_color;

}

}

}