Họ và tên: Hoàng Minh Hải Đăng

MSSV: 2110120

**ĐỒ HỌA MÁY TÍNH (CO3059)**

**LAB LIGHTING**

**BÀI 1: Chỉ định vật liệu cho đối tượng**

Sinh viên đọc hiểu mã nguồn ở phần phụ lục, sau đó sửa chương trình cho phép nhấn phím để thay đổi vật liệu của đối tượng. Tham số của 9 loại vật liệu được liệt kê trong bảng ở bài thực hành. Nhấn phím ‘1’, đối tượng sẽ được tô bằng màu ‘nhựa đen’. Nhấn phím ‘2’, đối tượng được tô bằng màu ‘đồng thau’ v.v… Nhấn phím ‘9’, đối tượng được tô bằng màu ‘bạc đánh bóng’.

Ở bài này, để làm được như yêu cầu đề bài, trước hết ta sẽ tiến hành đăng ký hàm callback mỗi khi có sự kiện nhấn phím bằng cách gọi hàm glutKeyboardFunc(myKeyboard) trong hàm main.

Sau đó, ta sẽ viết hàm callback để thay đổi vật liệu theo phím bấm từ 1-9 như sau:

void myKeyboard(unsigned char key, int x, int y){

if(key == '1'){

GLfloat new\_mat\_ambient[]={0.0f, 0.0f, 0.0f, 1.0f};

GLfloat new\_mat\_diffuse[]={0.01f, 0.01f, 0.01f, 1.0f};

GLfloat new\_mat\_specular[]={0.5f, 0.5f, 0.5f, 1.0f};

mat\_shininess[0] = 32.0f;

std::copy(new\_mat\_ambient, new\_mat\_ambient + 4, mat\_ambient);

std::copy(new\_mat\_diffuse, new\_mat\_diffuse + 4, mat\_diffuse);

std::copy(new\_mat\_specular, new\_mat\_specular + 4, mat\_specular);

}

else if (key == '2'){

GLfloat new\_mat\_ambient[]= {0.329412f, 0.223529f, 0.027451f, 1.0f};

GLfloat new\_mat\_diffuse[]={0.780392f, 0.568627f, 0.113725f, 1.0f};

GLfloat new\_mat\_specular[]={0.992157f, 0.941176f, 0.807843f, 1.0f};

mat\_shininess[0] = 27.8974f;

std::copy(new\_mat\_ambient, new\_mat\_ambient + 4, mat\_ambient);

std::copy(new\_mat\_diffuse, new\_mat\_diffuse + 4, mat\_diffuse);

std::copy(new\_mat\_specular, new\_mat\_specular + 4, mat\_specular);

}

else if (key == '3'){

GLfloat new\_mat\_ambient[]= {0.2125f, 0.01275f, 0.027451f, 1.0f};

GLfloat new\_mat\_diffuse[]={0.714f, 0.4284f, 0.18144f, 1.0f};

GLfloat new\_mat\_specular[]={0.393548f, 0.271906f, 0.166721f, 1.0f};

mat\_shininess[0] = 25.6f;

std::copy(new\_mat\_ambient, new\_mat\_ambient + 4, mat\_ambient);

std::copy(new\_mat\_diffuse, new\_mat\_diffuse + 4, mat\_diffuse);

std::copy(new\_mat\_specular, new\_mat\_specular + 4, mat\_specular);

}

if(key == '4'){

GLfloat new\_mat\_ambient[]={0.25f, 0.25f, 0.25f, 1.0f};

GLfloat new\_mat\_diffuse[]={0.4f, 0.4f, 0.4f, 1.0f};

GLfloat new\_mat\_specular[]={0.774597f, 0.774597f, 0.774597f, 1.0f};

mat\_shininess[0] = 76.8f;

std::copy(new\_mat\_ambient, new\_mat\_ambient + 4, mat\_ambient);

std::copy(new\_mat\_diffuse, new\_mat\_diffuse + 4, mat\_diffuse);

std::copy(new\_mat\_specular, new\_mat\_specular + 4, mat\_specular);

}

if(key == '5'){

GLfloat new\_mat\_ambient[]={0.19125f, 0.0735f, 0.025f, 1.0f};

GLfloat new\_mat\_diffuse[]={0.7038f, 0.27048f, 0.0828f, 1.0f};

GLfloat new\_mat\_specular[]={0.256777f, 0.137622f, 0.086014f, 1.0f};

mat\_shininess[0] = 12.8f;

std::copy(new\_mat\_ambient, new\_mat\_ambient + 4, mat\_ambient);

std::copy(new\_mat\_diffuse, new\_mat\_diffuse + 4, mat\_diffuse);

std::copy(new\_mat\_specular, new\_mat\_specular + 4, mat\_specular);

}

if(key == '6'){

GLfloat new\_mat\_ambient[]={0.24725f, 0.1995f, 0.0745f, 1.0f};

GLfloat new\_mat\_diffuse[]={0.75164f, 0.60648f, 0.22648f, 1.0f};

GLfloat new\_mat\_specular[]={0.628281f, 0.555802f, 0.366065f, 1.0f};

mat\_shininess[0] = 51.2f;

std::copy(new\_mat\_ambient, new\_mat\_ambient + 4, mat\_ambient);

std::copy(new\_mat\_diffuse, new\_mat\_diffuse + 4, mat\_diffuse);

std::copy(new\_mat\_specular, new\_mat\_specular + 4, mat\_specular);

}

if(key == '7'){

GLfloat new\_mat\_ambient[]={0.10588f, 0.058824f, 0.113725f, 1.0f};

GLfloat new\_mat\_diffuse[]={0.427451f, 0.470588f, 0.541176f, 1.0f};

GLfloat new\_mat\_specular[]={0.3333f, 0.3333f, 0.521569f, 1.0f};

mat\_shininess[0] = 9.84615f;

std::copy(new\_mat\_ambient, new\_mat\_ambient + 4, mat\_ambient);

std::copy(new\_mat\_diffuse, new\_mat\_diffuse + 4, mat\_diffuse);

std::copy(new\_mat\_specular, new\_mat\_specular + 4, mat\_specular);

}

if(key == '8'){

GLfloat new\_mat\_ambient[]={0.19225f, 0.19225f, 0.19225f, 1.0f};

GLfloat new\_mat\_diffuse[]={0.50754f, 0.50754f, 0.50754f, 1.0f};

GLfloat new\_mat\_specular[]={0.508273f, 0.508273f, 0.508273f, 1.0f};

mat\_shininess[0] = 51.2f;

std::copy(new\_mat\_ambient, new\_mat\_ambient + 4, mat\_ambient);

std::copy(new\_mat\_diffuse, new\_mat\_diffuse + 4, mat\_diffuse);

std::copy(new\_mat\_specular, new\_mat\_specular + 4, mat\_specular);

}

if(key == '9'){

GLfloat new\_mat\_ambient[]={0.23125f, 0.23125f, 0.23125f, 1.0f};

GLfloat new\_mat\_diffuse[]={0.2775f, 0.2775f, 0.2775f, 1.0f};

GLfloat new\_mat\_specular[]={0.773911f, 0.773911f, 0.773911f, 1.0f};

mat\_shininess[0] = 89.6f;

std::copy(new\_mat\_ambient, new\_mat\_ambient + 4, mat\_ambient);

std::copy(new\_mat\_diffuse, new\_mat\_diffuse + 4, mat\_diffuse);

std::copy(new\_mat\_specular, new\_mat\_specular + 4, mat\_specular);

}

glutPostRedisplay();

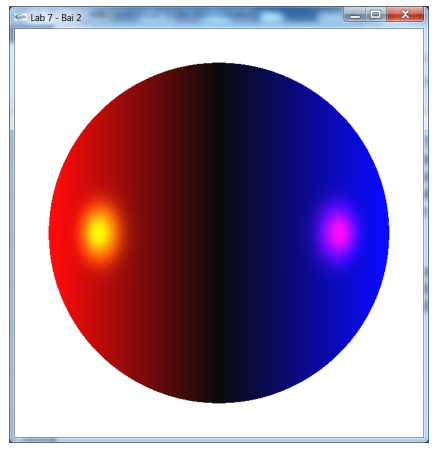
}

Kết quả thực thi của chương trình trên được quay lại và đính kèm ở link sau:

<https://drive.google.com/file/d/1_nkxIZ6JbZFygpv7D3i8dXDQSeuVuvao/view?usp=sharing>

**BÀI 2: Chỉ định tham số cho nguồn sáng và đối tượng**

Sinh viên sửa chương trình để nhận được kết xuất như hình bên dưới. Trong đó, nửa trái hình cầu có màu đỏ, phần ánh sáng phản chiếu màu vàng. Nửa phải hình cầu có màu xanh, phần ánh sáng phản chiếu màu tím.



Để đạt được yêu cầu của đề bài, ta sẽ tiến hành sửa tham số cho nguồn sáng và đối tượng như sau:

* Nguồn sáng bên trái: Thành phần khuếch tán màu đỏ, thành phần phản chiếu màu vàng:

const GLfloat leftLightDiffColor[] = { 1.0f, 0.0f, 0.0f, 1.0f }; // Mau khuech tan do

const GLfloat leftLightSpecColor[] = { 1.0f, 1.0f, 0.0f, 1.0f }; // Mau phan chieu vang

const GLfloat leftLightAmbColor[] = { 0.1f, 0.1f, 0.1f, 1.0f};

const GLfloat leftLightPos[] = { -5.0, 0.0, 0.0, 0.0 };

* Nguồn sáng bên trái: Thành phần khuếch tán màu xanh, thành phần phản chiếu màu tím:

const GLfloat rightLightDiffColor[] = { 0.0f, 0.0f, 1.0f, 1.0f }; // Mau khuech tan xanh

const GLfloat rightLightSpecColor[] = { 0.5f, 0.0f, 0.5f, 1.0f }; // Mau phan chieu tim

const GLfloat rightLightAmbColor[] = { 0.1f, 0.1f, 0.1f, 1.0f};

const GLfloat rightLightPos[] = { 5.0, 0.0, 0.0, 0.0 };

* Đối tượng được gán vật liệu phản xạ lại 100% cả 3 màu đỏ, lục xanh thành phần ánh sáng khuếch tán của nguồn sáng, phản xạ 100% cả 3 màu đỏ, lục, xanh thành phần ánh sáng phản chiếu của nguồn sáng:

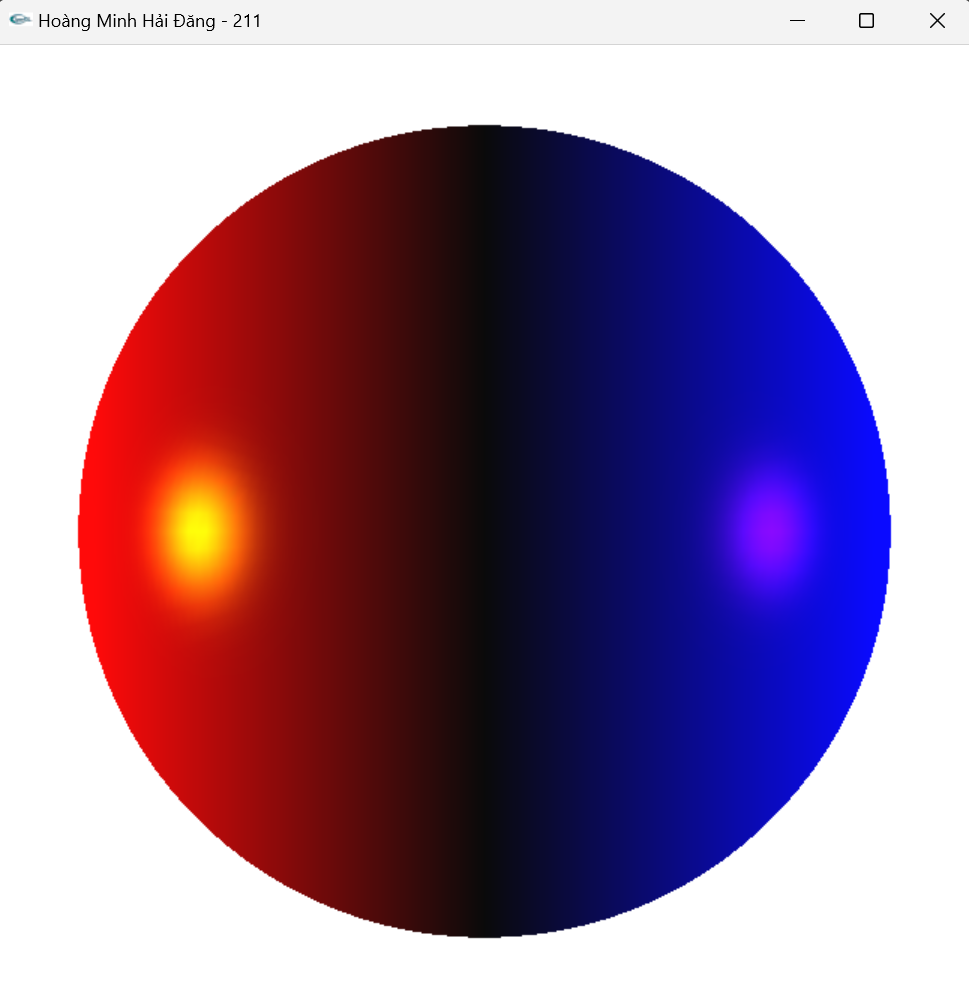
GLfloat ambient[] = { 0.1, 0.1, 0.1, 1.0f };

GLfloat diffuse[] = { 1.0, 1.0, 1.0, 1.0f }; // phan xa ca 3 mau thanh phan khuech tan

GLfloat specular[] = { 1.0, 1.0, 1.0, 1.0f }; // phan xa ca 3 mau thanh phan phan chieu

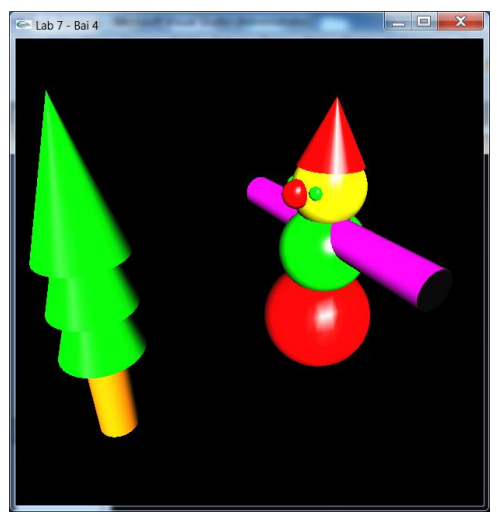
GLfloat shiness = 100.8;

Kết quả sau khi chạy chương trình như sau:



**BÀI 4: Vẽ ông già tuyết và cây thông Noel**

Sinh viên đọc hiểu chương trình mẫu trong phần phụ lục, sau đó tiến hành thêm những đối tượng còn thiếu để hoàn thiện khung cảnh ông già tuyết và cây thông Noel.



Để vẽ các hình còn thiếu, ta hoàn thiện hàm display như sau:

//draw the head

setMaterial(0.1, 0.1, 0.1,

1, 1, 0,

1, 1, 1);

glPushMatrix();

glTranslated(0.0, 1.5, 0.0);

glutSolidSphere(0.8, 50, 50);

glPopMatrix();

//draw nose

setMaterial(0.1, 0.1, 0.1,

1, 0, 0,

1, 1, 1);

glPushMatrix();

glTranslated(0.0, 1.5, 0.8);

glutSolidSphere(0.2, 50, 50);

glPopMatrix();

//draw left eye

setMaterial(0.1, 0.1, 0.1,

0, 1, 0,

1, 1, 1);

glPushMatrix();

glTranslated(-0.3, 1.5, 0.7);

glutSolidSphere(0.1, 50, 50);

glPopMatrix();

//draw right eye

setMaterial(0.1, 0.1, 0.1,

0, 1, 0,

1, 1, 1);

glPushMatrix();

glTranslated(0.3, 1.5, 0.7);

glutSolidSphere(0.1, 50, 50);

glPopMatrix();

//draw the hat

setMaterial(0.1, 0.1, 0.1,

1, 0, 0,

1, 1, 1);

glPushMatrix();

glTranslated(0.0, 1.7, 0.0);

glRotated(-90.0, 1.0, 0.0, 0.0);

glutSolidCone(0.75, 1.5, 30, 30);

glPopMatrix();

//draw left hand

setMaterial(0.1, 0.1, 0.1,

0, 0, 1,

1, 1, 1);

glPushMatrix();

glTranslated(0.0, 0.5, 0.0);

glRotated(-90.0, 0.0, 1.0, 0.0);

obj = gluNewQuadric();

gluQuadricDrawStyle(obj, GLU\_FILL);

gluCylinder(obj, 0.4, 0.4, 3.0, 30, 10);

glPopMatrix();

//draw right hand

setMaterial(0.1, 0.1, 0.1,

0, 0, 1,

1, 1, 1);

glPushMatrix();

glTranslated(0.0, 0.5, 0.0);

glRotated(90.0, 0.0, 1.0, 0.0);

obj = gluNewQuadric();

gluQuadricDrawStyle(obj, GLU\_FILL);

gluCylinder(obj, 0.4, 0.4, 3.0, 30, 10);

glPopMatrix();

Kết quả thực thi của chương trình trên được quay lại và đính kèm ở link sau:

<https://drive.google.com/file/d/1QWzF9AbG7d7Y8DUxHeZGaPkoGca_yV1X/view?usp=sharing>