Trường Đại học Sư phạm Kỹ thuật TP.HCM **ĐÁP ÁN MÔN: ĐỒ HỌA MÁY TÍNH**

Khoa CNTT **Học kỳ I – Năm học 2014-2015**

Bộ môn Tin học cơ sở **MÃ MÔN HỌC:** **1185040**

----------------------- -----------------------

***Tất cả các câu sau đây đều dùng các hàm của OpenGL***

**Câu 1: (2đ)**

Vẽ đồ thị của hàm trong hai chu kỳ.

**Đáp án:**

Trình bày phương pháp chuẩn hoá (0.5đ).

Code: (1.5đ)

void VeSin(void)

{

glColor3f(1,0,0);

int m, n;

double x, y, PI = 4.0\*atan(1.0);

glLineWidth(1);

glBegin(GL\_LINE\_STRIP);

for (m=0; m<=width/2; m++) {

x = 2\*2\*PI\*m/(width/2);

y = sin(x);

n = (int)(y\*250);

glVertex2i(m,n);

}

glEnd();

return;

}

**Câu 2: (1đ)**

Cho vector , hãy viết công thức và code chuẩn hóa vector *v* để nó có chiều dài bằng một.

**Câu 3: (1.5đ)**

Cho ba điểm *A*, *B*, *C* không thẳng hàng trong không gian là hãy viết công thức và code tính vector pháp tuyến của mặt phẳng đi qua ba điểm trên.

**Đáp án cho câu 2 và 3:**

void PhapVector(float P1[3], float P2[3], float P3[3], float V[3])

{

float x1,y1,z1;

float x2,y2,z2;

float x3,y3,z3;

float A,B,C,L;

x1 = P1[0];

y1 = P1[1];

z1 = P1[2];

x2 = P2[0];

y2 = P2[1];

z2 = P2[2];

x3 = P3[0];

y3 = P3[1];

z3 = P3[2];

A = y1\*(z2-z3) + y2\*(z3-z1) + y3\*(z1-z2);

B = z1\*(x2-x3) + z2\*(x3-x1) + z3\*(x1-x2);

C = x1\*(y2-y3) + x2\*(y3-y1) + x3\*(y1-y2);

L = sqrt(A\*A + B\*B + C\*C);

A = A/L;

B = B/L;

C = C/L;

V[0] = A;

V[1] = B;

V[2] = C;

return;

}

**Câu 4: (2đ)**

Hãy viết công thức và code của mô hình chiếu sáng dùng phản xạ ánh sáng môi trường và phản xạ ánh sáng khuếch tán.

**Đáp án:**

Trình bày công thức của mô hình chiếu sáng (0.5đ).

**Code:** (1.5đ)

void KhoiTaoMoHinhChieuSang(void)

{

glLightModeli(GL\_LIGHT\_MODEL\_TWO\_SIDE,GL\_TRUE);

// Anh sang moi truong

float mattrong[] = {0.0, 0.8, 0.0, 1.0};

float Ia[] = {0.8, 0.8, 0.8, 1.0};

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

glMaterialfv(GL\_BACK,GL\_AMBIENT,mattrong);

// Anh sang khuech tan

float Ip[] = {1.0, 1.0, 1.0, 1.0};

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

glMaterialfv(GL\_BACK,GL\_DIFFUSE,mattrong);

// Vector L cua tia sang toi

float L[] = {1.0, 1.0, 1.0, 0.0};

glLightfv(GL\_LIGHT0,GL\_POSITION,L);

// Anh sang phan xa guong

glLightfv(GL\_LIGHT0,GL\_SPECULAR,Ip);

glMaterialfv(GL\_BACK,GL\_SPECULAR,mattrong);

// Huong nhin song song voi truc z hoac nghieng voi truc z mot goc alpha

float alpha[] = {0};

glLightModelfv(GL\_LIGHT\_MODEL\_LOCAL\_VIEWER,alpha);

float ns[] = {50};

glMaterialfv(GL\_FRONT,GL\_SHININESS,ns);

glMaterialfv(GL\_BACK,GL\_SHININESS,ns);

return;

}

**Câu 5: (3.5đ)**

Xây dựng và viết code chiếu sáng mặt trụ đứng có bán kính R và chiều cao A (được sử dụng các hàm ở các câu 2, 3, và 4).

**Đáp án:**

Trình bày công thức của mặt trụ (0.5đ)

**Code:** (3.0đ)

float MatCong[300][300][3];

int N = 10, M = 10;

float gocquayx = 0;

float gocquayy = 0;

void Init(void)

{

// Thiet lap mau nen

glClearColor(1.0f, 1.0f, 1.0f, 0.0f);

// Thiet lap phep chieu

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(-width/2,width/2,-height/2,height/2,100,1000);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

glTranslatef(0,0,-400);

glEnable(GL\_DEPTH\_TEST);

KhoiTaoMatTru();

KhoiTaoMoHinhChieuSang();

return;

}

void KhoiTaoMatTru(void)

{

int R = 70, A = 150;

float PI = 4.0\*atan(1.0);

float u, du = 2\*PI/M;

float v, dv = 1.0/N;

float x, y, z;

int m, n;

for (n=0; n<=N; n++) {

v=n\*dv;

for (m=0; m<=M; m++) {

u=m\*du;

x = R\*cos(u);

y = A\*v;

z = R\*sin(u);

MatCong[n][m][0] = x;

MatCong[n][m][1] = y;

MatCong[n][m][2] = z;

}

}

return;

}

void ChieuSangMatCong(void)

{

glEnable(GL\_LIGHTING);

glEnable(GL\_LIGHT0);

glPushMatrix();

glRotatef(gocquayx,1,0,0);

glRotatef(gocquayy,0,1,0);

float mauden[] = {0.0, 0.0, 0.0, 1.0};

float mauvang[] = {0.8, 0.8, 0.0, 1.0};

float V[3];

int n, m;

for (n=0; n<=N-1; n++) {

if (n%2 == 0) {

glMaterialfv(GL\_FRONT,GL\_AMBIENT,mauden);

glMaterialfv(GL\_FRONT,GL\_DIFFUSE,mauden);

glMaterialfv(GL\_FRONT,GL\_SPECULAR,mauden);

}

else {

glMaterialfv(GL\_FRONT,GL\_AMBIENT,mauvang);

glMaterialfv(GL\_FRONT,GL\_DIFFUSE,mauvang);

glMaterialfv(GL\_FRONT,GL\_SPECULAR,mauvang);

}

for (m=0; m<=M-1; m++) {

glBegin(GL\_QUADS);

PhapVector(MatCong[n][m],MatCong[n+1][m],MatCong[n+1][m+1],V);

glNormal3fv(V);

glVertex3fv(MatCong[n][m]);

glVertex3fv(MatCong[n+1][m]);

glVertex3fv(MatCong[n+1][m+1]);

glVertex3fv(MatCong[n][m+1]);

glEnd();

}

}

glPopMatrix();

glDisable(GL\_LIGHT0);

glDisable(GL\_LIGHTING);

return;

}

-------HẾT-------