МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

Лабораторна робота №4

з дисципліни

Комп’ютерна графіка

Виконав:

студент групи КВ-81

Петренко М. Е.

Київ 2011

*Код програми:*

Pgl.cpp:

#include "stdafx.h"

#include "load\_jpg.h"

float a,b,c;

struct vertex

{

float coo[3];

float nor[3];

};

struct field

{

float U[128][128];

};

vertex vertices[128][128];

field A,B;

field \*p=&A,\*n=&B;

void init()

{

int i,j;

memset(vertices,0,sizeof(vertices));

memset(&A,0,sizeof(A));

memset(&B,0,sizeof(B));

for(i=0;i<128;i++)

{

for(j=0;j<128;j++)

{

vertices[i][j].coo[0]=1.0f-2.0f\*i/127.0f;

vertices[i][j].coo[1]=1.0f-2.0f\*j/127.0f;

vertices[i][j].nor[2]=-4.0f/127.0f;

}

}

};

void time\_step()

{

int i,j,i1,j1;

i1=rand()%110;

j1=rand()%110;

/\*1\*/

if((rand()&127)==0)

for(i=-3;i<4;i++)

{

for(j=-3;j<4;j++)

{

float v=6.0f-i\*i-j\*j;

if(v<0.0f)v=0.0f;

n->U[i+i1+3][j+j1+3]-=v\*0.004f;

}

}

for(i=1;i<127;i++)

{

for(j=1;j<127;j++)

{

/\*2\*/

vertices[i][j].coo[2]=n->U[i][j];

vertices[i][j].nor[0]=n->U[i-1][j]-n->U[i+1][j];

vertices[i][j].nor[1]=n->U[i][j-1]-n->U[i][j+1];

/\*3\*/

#define vis 0.005f

float laplas=(n->U[i-1][j]+

n->U[i+1][j]+

n->U[i][j+1]+

n->U[i][j-1])\*0.25f-n->U[i][j];

/\*4\*/

p->U[i][j]=((2.0f-vis)\*n->U[i][j]-p->U[i][j]\*(1.0f-vis)+laplas);

}

}

/\*5\*/

for(i=1;i<127;i++)

{

glBegin(GL\_TRIANGLE\_STRIP);

for(j=1;j<127;j++)

{

glNormal3fv(vertices[i][j].nor);

glVertex3fv(vertices[i][j].coo);

glNormal3fv(vertices[i+1][j].nor);

glVertex3fv(vertices[i+1][j].coo);

}

glEnd();

}

/\*5\*/

field \*sw=p;p=n;n=sw;

}

void redraw()

{

static int counter=0;

static float cl;

if(counter==0)

{

cl=clock();

}

if(counter++==100)

{

counter=0;

printf("FPS = %f \n",2\*100000.0f/(clock()-cl));

cl=clock();

}

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

glLoadIdentity ();

float pos[4]={0.0f,0.0f,-1.0f,0.0f};

glTranslatef (0.0f, 0.3f, 0.0f);

glTranslatef (0.0f, 0.0f, -1.5f);

glRotatef(160.0f,1.0f,0.0f,0.0f);

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

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

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

time\_step();

glutSwapBuffers();

}

void motion(int x, int y)

{

c=x\*0.1f;

}

void reshape(int width, int height)

{

glMatrixMode (GL\_PROJECTION);

glLoadIdentity ();

gluPerspective(60,(width+.1)/(height+.1),0.1f,100.0f);

glMatrixMode (GL\_MODELVIEW);

glViewport (0, 0, width, height);

}

main(int argc, char \*argv[])

{

glutInit(&argc, argv);

glutInitWindowSize(800,600);

glutInitDisplayMode(GLUT\_RGBA|GLUT\_DEPTH|GLUT\_DOUBLE);

glutCreateWindow("Petrenko Maksim");

glutIdleFunc(redraw);

glutDisplayFunc(redraw);

glutMotionFunc(motion);

glutReshapeFunc(reshape);

glewInit();

init();

loadjpgGL("..\\MAP\\PHONG.jpg");

glTexEnvf(GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, GL\_BLEND);

glTexGeni(GL\_S, GL\_TEXTURE\_GEN\_MODE, GL\_SPHERE\_MAP);

glTexGeni(GL\_T, GL\_TEXTURE\_GEN\_MODE, GL\_SPHERE\_MAP);

glEnable(GL\_TEXTURE\_GEN\_S);

glEnable(GL\_TEXTURE\_GEN\_T);

glEnable(GL\_DEPTH\_TEST);

glEnable(GL\_TEXTURE\_2D);

glEnable(GL\_NORMALIZE);

glEnable(GL\_LIGHTING);

glEnable(GL\_LIGHT0);

glutMainLoop();

return 0;

}