ФГБОУ ВО «Чувашский государственный университет»

им. И.Н. Ульянова.

Кафедра Вычислительной техники

Дисциплина «Алгоритмические основы компьютерной графики»

Лабораторная работа № 1

**Инициализация OpenGL.**

**Использование библиотеки freeglut.**

Выполнил: студент

группы ИВТ-42-19

Майоров Николай

Проверил:

Степанов В.В.

Чебоксары 2021

Задание к лабораторной работе.

В рамках данной лабораторной работы необходимо изучить функции GLUT для инициализации OpenGL и взаимодействия с операционной системой, разобраться с приведенным примером, а так же внести следующие изменения:

1. Используя функцию glutKeyboardFunc добавить функцию обработки события нажатия на клавишу;
2. Реализовать циклическое изменение цвета объекта по нажатию на клавишу, используя предварительно заданный массив цветов. В массиве цвета лежат в следующем порядке: черный, белый, синий и красный. Следует учесть, что во время защиты может потребоваться добавление новых цветов в массив;
3. Для защиты следует заранее подумать над тем, как организовать автоматическое изменение цветов с заданной скоростью;
4. Во время защиты реализовать плавную смену цветов и добавить приближение и увеличение чайника.

Код программы:

#include <stdio.h>

#include <Windows.h>

#include <gl/GL.h>

#include <gl/GLU.h>

#include "GL/freeglut.h"

int i, timer;

double q = 1;

double p = 0;

double x = 0;

double y = 0;

double z = 0;

double x1 = 0;

double y1 = 0;

double z1 = 0;

/\*double colors[4][3] = { {0.0,0.0,0.0},

{1.0,1.0,1.0},

{0.0,0.0,1.0},

{1.0,0.0,0.0},

};\*/

void Simulation(int value) // функция вызывается каждые 20 мс, изменение параметров сцены

{

glutPostRedisplay(); // уфункция отмечает, что окно нуждается в перерисовке

//таймер

/\* timer += 20;

if (timer >= 1000) {

if (i < 3) {

i++;

}

else

i=0;

timer = 0;

} \*/

glutTimerFunc(20, Simulation, 0); // функция вызывается каждые 20 мс

}

void Reshape(int w, int h) // фун-ия при изменении размеров окна

{

glViewport(0, 0, (GLsizei)w, (GLsizei)h); // устанавливаем новую область просмотра, равную всей области экрана

glMatrixMode(GL\_PROJECTION); // устанавливаем матрицу проекции с правильным аспектом

glLoadIdentity();

gluPerspective(25.0, (float)w / h, 0.2, 70.0);

}

void Display(void) //функция для перерисовки окна

{

// чистим буфер цвета

glClearColor(0.22, 0.88, 0.11, 1.0);

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

// включаем тест глубины

glEnable(GL\_DEPTH\_TEST);

// устанавливаем камеру и ее положение

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

gluLookAt(5, 5, 7.5, 0, 0, 0, 0, 1, 0);

timer += 20;

if (timer >= 20)

{

if (q < 1.5)

{

q += 0.02;

}

else

{

if (p > -0.5)

{

p -= 0.02;

}

else {

q = 1; p = 0;

}

}

timer = 0;

}

glScalef(q+p, q+p, q+p);

//

timer += 20;

if (timer >= 20)

{

if (x < 1)

x += 0.05;

else

{

if (y < 1)

y += 0.05;

else

{

if (z < 1)

z += 0.05;

else

{

if (x1 < 1)

x1 += 0.05;

else

{

if (y1 < 1)

y1 += 0.05;

else

{

if (z1 < 1)

z1 += 0.05;

else

{

x = 0; y = 0; z = 0; x1 = 0; y1 = 0; z1 = 0;

}

}

}

}

}

}

}

// выводим объект - чайник (красный (1,0,0))

glColor3f(x-x1 ,y-y1 ,z-z1 );

glutWireTeapot(1.0);

// смена переднего и заднего буферов

glutSwapBuffers();

}

/\*// функция обработи нажатия клавиш

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

{

printf("Key code is %i\n", (unsigned int)key); //пишет в консоль код клавиши

//смена при нажатии

if ((unsigned int)key) {

if (i < 3) {

i++;

}

else

i = 0;

}

}\*/

void main(int argc, char\*\* argv)

{

glutInit(&argc, argv);

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

// четверка цветов, двойная буферизация, буфер глубины, ин-ия для каждого пикселя

// создание окна

glutInitWindowPosition(200, 200); // верхний левый угол окна

glutInitWindowSize(700, 600); // размер окна

glutCreateWindow("LR\_1"); // создаем окно

glutDisplayFunc(Display); // функция для перерисовки окна

glutReshapeFunc(Reshape); //функция при изменении разверов окна

glutTimerFunc(20, Simulation, 0); // Функция, которая будет вызвана через 20 секунд

//glutKeyboardFunc(KeyboardFunc); // функция, которая вызывается при нажатии на клавишу

glutMainLoop(); // основной цикл обработки сообщений ОС

return;

}

|  |  |
| --- | --- |
| glutInit | Главная инициализирующая функция.  glutInit(int \*\*argcp, char \*\*argv);     argcp - это указатель на еще не измененную переменную argc главной функции программы (main). После возврата из функции, значение, на которое указывает argcp, может измениться, так как glutInit вычленяет все опции командной строки, относящиеся к библиотеке GLUT, например, в системе X-Window все опции, относящиеся к управлению окнами, ассоциируются с GLUT.     argv - это еще не измененная переменная argv главной функции. |
| glutInitDisplayMode | glutInitDisplayMode(unsigned int mode);     mode - это режим состояния экрана, который может быть получен с помощью двоичной ИЛИ-комбинации режимов GLUT (каждый режим представляет собой двоичную маску). |
| glutInitWindowPosition | glutInitWindowPosition(int x, int\*\*y);  x,y  = позиция окна на экране в пикселях (точнее - позиция левого верхнего угла окна) |
| glutInitWindowSize | glutInitWindowSize(int width, int \*\*height);  width,height  - ширина и высота окна в пикселях. |
| glutCreateWindow | int glutCreateWindow(const char \*title);  Эта команда создаёт окно с заголовком, который вы укажете в качестве параметра и возвращает HANDLER окна в виде числа int. |
| glutPostRedisplay | Функция для явного указания, что окно надо обновить |
| glutDisplayFunc | С помощью glutDisplayFunc() задается функция  рисования для окна приложения, которая вызывается при необходимости создания или восстановления изображения. |
| glutReshapeFunc | Через glutReshapeFunc() устанавливается функция обработки изменения размеров окна пользователем, которой передаются новые  размеры. |
| glutKeyboardFunc | glutKeyboardFunc(Func);Вызывает функцию Func при нажатии на клавишу |
| glutTimerFunc | void glutTimerFunc(unsigned int millis, void (\*func)(int value), int value);  Эта функция устанавливает таймер. Первый параметр millis задает время в миллисекундах, по истечении которых вызывается функция, которая указана как второй параметр. Третий параметр value указывает индентификатор таймера, т.е. таймеров может быть одновременно запущено несколько. То же самое значение value есть и в функции, которая вызывается таймером. Это нужно для того, чтобы определить, какой таймер сработал, т.е. если ваша функция обрабатывает несколько таймеров. |
| glutMainLoop | Контроль всех событий происходит внутри бесконечного цикла в функции void glutMainLoop (void), которая обычно вызывается в конце любой программы, использующей GLUT. |
| glutSwapBuffers | обменивает буфера, отображая окно целиком после того, как оно уже сформировано. |
| glutWireTeapot | Выводит чайник |

Результат работы программы:



