МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Чувашский государственный университет имени И.Н. Ульянова»

Факультет информатики и вычислительной техники

Кафедра вычислительной техники

Отчет по лабораторной работе №1

по дисциплине: «Алгоритмические основы компьютерной графики»

«Инициализация OpenGL. Использование библиотеки freeglut»

Выполнила: студент гр.ИВТ-41-19

Семенов Александр Александрович

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;

**Таблица с описанием используемых функций:**

|  |  |
| --- | --- |
| glutInit | происходит инициализация GLUT и обрабатываются предназначенные для нее аргументы командной строки. |
| glutInitDisplayMode | Задаётся режим (параметры) окна отображения. Параметры glutInitDisplayMode:  GLUT\_RGB - Используется режим RGB  GLUT\_RGBA - Используется режим RGBA  GLUT\_INDEX - Используется индексный режим (т.е. цвет выбирается из палитры)  GLUT\_SINGLE - Используется одинарный буфер (не для анимации)  GLUT\_DOUBLE - Используется двойной буфер (для анимации)  GLUT\_DEPTH - Используется также буфер глубины (для 3D изображений)  GLUT\_STENCIL - Используется буфер трафарета. |
| glutInitWindowPosition | задают положение окна |
| glutInitWindowSize | задаются размеры окна (ширина, высота) |
| glutCreateWindow | После этих шагов, окно может быть создано с помощью glutCreateWindow(название) |
| glutPostRedisplay | заставляет систему перерисовать текущее окно |
| glutDisplayFunc | отвечает за рендер сцены |
| glutReshapeFunc | вызывается при изменении размеров окна — функция выполняет масштабирование в окне |
| glutKeyboardFunc | задает функцию обработки нажатия клавиш клавиатуры |
| glutTimerFunc | функция устанавливает таймер |
| glutMainLoop | запуск основного цикла работы приложения, построенного на базе GLUT |
| glutSwapBuffers | меняет местами содержимое заднего и переднего буферов |
| glutWireTeapot | Поверхность чайника генерируется с использованием функций OpenGL построения кривых Безье. Параметр size устанавливает значение двойной точности с плавающей запятой - максимальный радиус тела чайника. |

Текст программы с подробными комментариями:

#include <GL/freeglut.h>

#include<GL/gl.h>

#include<GL/GLU.h>

#include <stdio.h>

#include <windows.h>

double X1 = 0, Y1 = 0, Z1=0;

struct RGB{

double R;

double G;

double B;

};

RGB Colors[4] = { {0.0,0.0,0.0},{1,1,1},{0,0,1},{1,0,0} };

double c1=0, c2=0, c3 =0 ;

int i = 0;

// функция вызывается каждые 20 мс

void Simulation(int value)

{

// устанавливаем признак того, что окно нуждается в перерисовке

glutPostRedisplay();

// эта же функция будет вызвана еще раз через 20 мс

glutTimerFunc(20, Simulation, 0);

};

// функция, вызываемая при изменении размеров окна

void Reshape(int w, int h)

{

// установить новую область просмотра, равную всей области окна

glViewport(0, 0, (GLsizei)w, (GLsizei)h);

// установить матрицу проекции с правильным аспектом

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

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

};

// функция вызывается при перерисовке окна

// в том числе и принудительно, по командам glutPostRedisplay

void Display(void)

{

// отчищаем 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

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

gluLookAt(5+X1, 5+Y1, 7.5+Z1, 0, 0, 0, 0+X1, 1+Y1, 0+Z1);

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

glColor3f(c1, c2, c3);

glutWireTeapot(1.0);

// смена переднего и заднего буферов

glutSwapBuffers();

};

// Функция обработки нажатия клавиш

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

{

printf("Key code is %i\n", (unsigned int)key);

switch (key)

{

case 'p':

exit(0);

break;

case 'a':

X1 += 1;

Y1 += 1;

Z1 += 1;

glutPostRedisplay();

break;

case 'd':

Y1 -= 1;

Z1 -= 1;

X1 -= 1;

glutPostRedisplay();

break;

case 'b':

i++;

c1 = Colors[i].R;

c2 = Colors[i].G;

c3 = Colors[i].B;

glutPostRedisplay();

}

};

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

{

// инициализация библиотеки GLUT

glutInit(&argc, argv);

// инициализация дисплея (формат вывода)

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

// создание окна:

// 1. устанавливаем верхний левый угол окна

glutInitWindowPosition(200, 200);

// 2. устанавливаем размер окна

glutInitWindowSize(600, 600);

// 3. создаем окно

glutCreateWindow("laba1");

// устанавливаем функцию, которая будет вызываться для перерисовки окна

glutDisplayFunc(Display);

// устанавливаем функцию, которая будет вызываться при изменении размеров окна

glutReshapeFunc(Reshape);

// устанавливаем функцию которая будет вызвана через 20 мс

glutTimerFunc(20, Simulation, 0);

// устанавливаем функцию, которая будет вызываться при нажатии на клавишу

glutKeyboardFunc(KeyboardFunc);

// основной цикл обработки сообщений ОС

glutMainLoop();

}

Скриншот работы программы:

