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.»

Выполнил: студент гр.ИВТ-41-19

Васильев Даниил Дмитриевич

Проверил: Галибин С. В.

Чебоксары 2021 г.

**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, в которой необходимо**

**кратко указать назначение функции и её параметров:**

|  |  |
| --- | --- |
| glutInit | Инициализирует библиотеку GLUT и согласовывает сеанс с оконной системой. Во время этого процесса glutInit может вызвать завершение программы GLUT с сообщением об ошибке пользователю, если GLUT не может быть правильно инициализирован.  void glutInit(int \*argcp, char \*\*argv);  **argcp** - Указатель на неизмененную переменную argc программы из файла main. По возвращении значение, на которое указывает argcp, будет обновлено, потому что glutInit извлекает все параметры командной строки, предназначенные для библиотеки GLUT.  **argv** - Неизмененная переменная argv программы из файла main. Как и argcp, данные для argv будут обновлены, потому что glutInit извлекает все параметры командной строки, понятные библиотеке GLUT. |
| glutInitDisplayMode | Устанавливает начальный режим отображения.  void glutInitDisplayMode(unsigned int mode);  **mode** - Режим отображения, обычно побитовое ИЛИ битовых масок режима отображения GLUT.( GLUT\_RGBA, GLUT\_DOUBLE и т.д.) |
| glutInitWindowPosition | Устанавливает исходное положение.  void glutInitWindowPosition(int x, int y);  **x** - Расположение окна по оси X в пикселях.  **y** - Расположение окна по оси Y в пикселях. |
| glutInitWindowSize | Устанавливает исходный размер.  void glutInitWindowSize(int width, int height);  **width** – Ширина в пикселях.  **height** – Высота в пикселях. |
| glutCreateWindow | Создает окно верхнего уровня.  int glutCreateWindow(char \*name);  **name** - Строка символов ASCII для использования в качестве имени окна. |
| glutPostRedisplay | Помечает текущее окно как нуждающееся в повторном отображении.  void glutPostRedisplay(void); |
| glutDisplayFunc | Устанавливает обратный вызов отображения для текущего окна.  void glutDisplayFunc(void (\*func)(void));  **func** - Новая функция обратного вызова дисплея. |
| glutReshapeFunc | Устанавливает обратный вызов изменения формы для текущего окна.  void glutReshapeFunc(void (\*func)(int width, int height));  **func** - Новая функция обратного вызова reshape. |
| glutKeyboardFunc | Устанавливает обратный вызов клавиатуры для текущего окна.  void glutKeyboardFunc(void (\*func)(unsigned char key, int x, int y));  **func** - Новая функция обратного вызова клавиатуры. |
| glutTimerFunc | Регистрирует обратный вызов таймера, который будет запущен через указанное количество миллисекунд.  void glutTimerFunc(unsigned int msecs, void (\*func)(int value), value); |
| glutMainLoop | Входит в цикл обработки событий GLUT.  void glutMainLoop(void); |
| glutSwapBuffers | Меняет местами буферы текущего окна при двойной буферизации.  void glutSwapBuffers(void); |
| glutWireTeapot | Визуализирует каркасный чайник.  void glutWireTeapot(GLdouble size);  **size** – Относительный размер чайника. |

**Текст программы с подробными комментариями.**

#include <stdio.h>

#include <windows.h>

#include <GL/gl.h>

#include <GL/glu.h>

#include "GL/freeglut.h"

unsigned int KeyCode;

unsigned int timer = 0;

short int counter = 0;

float color[3] = { 1.0, 0.0, 0.0 };

float KitColor[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} };

// функция вызывается каждые 20 мс

void Simulation(int value)

{

// если нажата клавиша "Enter" (код 13), то с периодичностью в 1 секунду

// меняем цвет чайника

if (KeyCode == 13) {

++timer;

if (timer == 50) {

timer = 0;

// меняем цвет чайника

color[0] = KitColor[counter][0];

color[1] = KitColor[counter][1];

color[2] = KitColor[counter][2];

++counter;

}

if (counter == 4) counter = 0;

}

// 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, 5, 7.5, 0, 0, 0, 0, 1, 0);

// выводим объект ‐ значение берем из массива

glColor3f(color[0],color[1],color[2]);

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);

// присваиваем значение клавиши глобальной переменной

KeyCode = (unsigned int)key;

};

int 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);

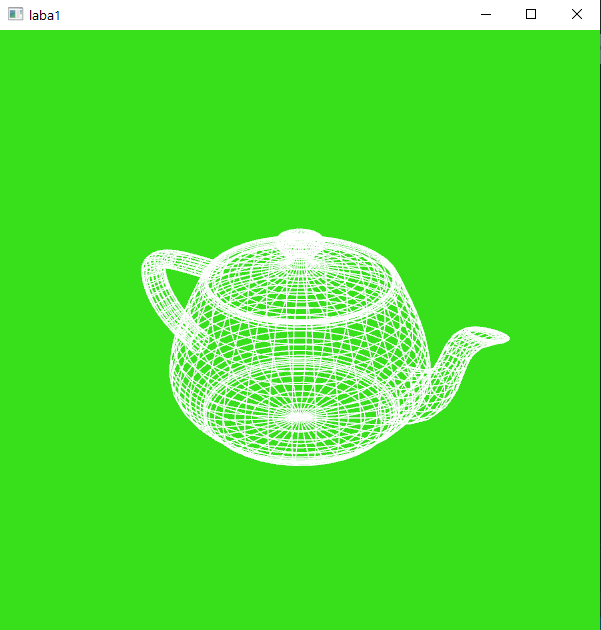
// устанавливаем функцию, которая будет вызываться при нажатии на клавишу

glutKeyboardFunc(KeyboardFunc);

glutMainLoop();

}

**Скриншот работы программы.**

****

****

**Вывод:** были получены практические навыки работы с библиотекой OpenGL, вспомогательной библиотекой freeglut, а так же со средой программирования Visual Studio.