МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное   
учреждение высшего образования

«Чувашский государственный университет имени И. Н. Ульянова»

Факультет информатики и вычислительной техники

Кафедра вычислительной техники

# **Лабораторная работа 1.**

# Инициализация OpenGL и использование библиотеки freeglut

по дисциплине Алгоритмические основы компьютерной графики

Выполнил: Фисунов А.П.  
студент группы: ИВТ-42-22

Проверил: Васильев Н. С.

**Чебоксары, 2024**

Лабораторная работа №1

## Задание

В рамках данной лабораторной работы необходимо изучить функции GLUT для инициализации

OpenGL и взаимодействия с операционной системой, разобраться с приведенным примером, а

также внести следующие изменения:

1. Реализовать циклическое изменение цвета объекта по нажатию на клавишу, используя предварительно заданный массив цветов. В массиве цвета лежат в следующем порядке: черный, белый, синий, красный и фиолетовый. Следует учесть, что во время защиты может потребоваться добавление новых цветов в массив;

2. Для защиты следует заранее подумать над тем, как организовать автоматическое изменение цветов с заданной скоростью, например так, чтобы цвета менялись каждую секунду.

3. Особое внимание необходимо уделить оформлению программы. В частности, необходимо чтобы блоки кода были выравнены в соответствии с уровнем вложенности, переменные и функции носили осмысленные имена, дающие возможно определить их назначение, а также в программе должно присутствовать необходимое количество комментариев.

## Текст программы

#include <windows.h>

#include <iostream>

#include <vector>

#include <GL/gl.h>

#include <GL/glu.h>

#include "GL/freeglut.h"

// используем пространство имен стандартной библиотеки

using namespace std;

#define CHANGE\_TIME\_MS 2000

int index = 0;

vector<double> black = { 0, 0, 0 };

vector<double> white = { 1, 1, 1 };

vector<double> blue = { 0, 0, 1 };

vector<double> red = { 1, 0, 0 };

vector<double> violet = { 1, 0, 1 };

vector< vector<double>> colors = { black, white, blue, red, violet };

int max\_index = colors.size();

// 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 и буфер глубины

glClearColor(0.2, 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 - красный (1,0,0) чайник

auto color\_now = colors[index];

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

glutWireTeapot(1.0);

// смена переднего и заднего буферов

glutSwapBuffers();

};

// 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);

};

#ifdef CHANGE\_TIME\_MS

void change\_color\_by\_timer(int value) {

// смена цвета по таймеру каждые CHANGE\_TIME\_MS мс

index = (index + 1) % max\_index;

glutPostRedisplay();

// эта же функция будет вызвана еще раз через мс

glutTimerFunc(CHANGE\_TIME\_MS, change\_color\_by\_timer, 0);

}

#endif

// Функция обработки нажатия клавиш

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

{

index = (index + 1) % max\_index;

printf("Key code is %i\n Color changed\n", key);

};

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

{

setlocale(LC\_ALL, "ru");

// 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(800, 600);

// 3. создаем окно

glutCreateWindow("Laba\_01");

// УСТАНОВКА ФУНКЦИЙ ОБРАТНОГО ВЫЗОВА

// 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);

#ifdef CHANGE\_TIME\_MS

glutTimerFunc(20, change\_color\_by\_timer, 0); // смена цвета по таймеру каждые CHANGE\_TIME\_MS мс

#endif

// основной цикл обработки сообщений ОС

glutMainLoop();

};

## Скриншот работы программы

