ФГБОУ ВО “Чувашский государственный университет им.

И.Н. Ульянова”

Кафедра вычислительной техники

Лабораторная работа № 1

По дисциплине “Алгоритмические основы компьютерной графики”

Инициализация OpenGL.

Использование библиотеки freeglut.

Выполнил: студент группы ИВТ-42-19

Кузьмина Екатерина Андреевна

Проверил:

Степанов В.В.

Чебоксары 2021

**Цель работы:** Познакомиться с библиотекой OpenGLб вспомогательной библиотекой GLUT, а также со средой программирования Visual Studio.

**Выполнение работы**

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 | Используется для инициализации библиотеки GLUT и согласует сеанс с оконной системой. Параметры: &argc – количество аргументов, argv – их описание в виде указателя на строку |
| glutInitDisplayMode | Задает начальный режим отображения. Параметры: mode – режим отображения. Значения:   * GLUT\_RGBA и GLUT\_RGB - выбирает окно режима RGBA. * GLUT\_INDEX - режим цветового индекса. * GLUT\_SINGLE – режим одинарной буферизации. * GLUT\_DOUBLE - режим двойной буферизации. * GLUT\_ACCUM - буфер накопления. * GLUT\_ALPHA - выбор окна с альфа-компонентом для цветовых буферов. * GLUT\_DEPTH - буфер глубины. * GLUT\_STENCIL - буфер набора элементов. |
| glutInitWindowPosition | Устанавливает положение окна, его верхнего левого угла. Параметры:x - число пикселей от левой части экрана, y - количество пикселей от верхней части экрана. |
| glutInitWindowSize | Устанавливает размер окна. Параметры: width – ширина окна, height - высота окна. |
| glutCreateWindow | создает окно верхнего уровня. Параметр: «Имя» будет предоставлено оконной системе в качестве имени окна. Цель состоит в том, чтобы оконная система пометила окно именем. |
| glutPostRedisplay | Помечает текущее окно как нуждающийся в повторном воспроизведении. |
| glutDisplayFunc | задает обратный вызов дисплея для текущего окна. funс – новая функция обратного вызова данных. |
| glutReshapeFunc | задает обратный вызов изменения формы для текущего окна. Параметры width и height обратного вызова указывают новый размер окна в пикселях. funс – новая функция обратного вызова изменения формы. |
| glutKeyboardFunc | задает обратный вызов клавиатуры для текущего окна. Когда пользователь вводит в окно, каждое нажатие клавиши, генерирующее символ ASCII, будет генерировать обратный вызов клавиатуры.  Параметры: ключевым параметром обратного вызова является сгенерированный символ ASCII, x и y указывают расположение мыши в относительных координатах окна при нажатии клавиши. |
| glutTimerFunc | регистрирует обратный вызов таймера, который должен быть запущен через указанное количество миллисекунд. Параметр value для обратного вызова таймера будет значением параметра value для glutTimerFunc. Несколько обратных вызовов таймера в одно и то же или разное время могут быть зарегистрированы одновременно. |
| glutMainLoop | входит в цикл обработки событий GLUT. Эта процедура должна вызываться не более одного раза в программе GLUT. После вызова эта рутина никогда не вернется. Он будет вызывать по мере необходимости любые обратные вызовы, которые были зарегистрированы. |
| glutSwapBuffers | способствует тому, чтобы содержимое заднего буфера слоя, использующее текущее окно, стало содержимым переднего буфера |
| glutWireTeapot | отображает каркасный чайник соответственно |

Текст программы с подробными комментариями.

#include <windows.h>

#include <GL/gl.h>

#include <GL/glu.h>

#include "GL/freeglut.h"

#include <iostream>

double Colors[5][3] = { {1.0,0.0,0.0},{1.0,1.0,1.0},{0.0,0.0,1.0},{1.0,1.0,0.5},{0.0,1.0,1.0} };

int i = 0;

int timer = 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

timer += 20;

if (timer >= 500){

if (i < 5){

i++;

}

else

i = 0;

timer = 0;

}

glutTimerFunc(20, Simulation, 0);

}

// функция вызывается при перерисовке окна

// в том числе и принудительно, по командам glutPostRedisplay

void Display(void)

{

// отчищаем буфер цвета

glClearColor(0.22, 0.8, 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

glColor3f(Colors[i][0], Colors[i][1], Colors[i][2]);

glutWireTeapot(1.0);

// смена переднего и заднегkо буферов

glutSwapBuffers();

};

// Функция обработки нажатия клавиш

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

{

//std::cout << "Key is " << key<<"\n";

if (key == 'k'){

if (i < 5){

i++;

}

else

i = 0;

}

};

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);

};

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);

// основной цикл обработки сообщений ОС

glutMainLoop();

}

Скриншот работы программы.

