Министерство образования и науки Российской Федерации

Новосибирский государственный технический университет

Кафедра прикладной математики

**Лабораторная работа №1**

**По дисциплине: «Компьютерная графика»**

Факультет: ПМИ

Группа: ПМ-43

Студенты: Иост А.В.

Мошкина А.Д.

Преподаватель: Задорожный А.Г.

Новосибирск

2017

1. **Цель работы**

Ознакомиться с основами использования библиотеки OpenGL.

1. **Задание**

Замкнутая ломаная GL\_LINE\_LOOP

1. **Возможности**

* Рисование нескольких примитивов
* Изменение цвета отдельного примитива
* Выбор примитива
* Перемещение отдельного примитива
* Изменение толщины линии и размера точек для отдельного примитива
* Удаление отдельной точки
* Удаление текущего примитива

1. **Описание структур**

Сам контейнер задаётся следующим образом

/\* задание контейнера характеристик точек \*/

struct Characters

{

GLubyte ColorR = 0, ColorG = 0, ColorB = 0;

GLubyte PointSize = 5;

GLubyte LineSize = 3;

};

Контейнер вершин задается следующим образом

/\* задание контейнера вершин \*/

struct type\_point

{

GLint x, y;

type\_point(GLint \_x, GLint \_y) { x = \_x; y = \_y; }

};

Для хранения нескольких примитивов одновременно используется вектор векторов структур.

Также используется вектор характеристик точек.

vector <vector <type\_point>> Points;

vector <Characters> Numb;

1. **Описание работы с программой**

Правая кнопка мыши - вызов контекстного меню;

Левая кнопка мыши – рисование точки;

Колесо мыши – удаление последней точки;

Клавиша R - изменить красную компоненту цвета;

Клавиша G - изменить зеленую компоненту цвета;

Клавиша B - изменить синюю компоненту цвета;

Клавиша W - переместить вверх;

Клавиша S - переместить вниз;

Клавиша A - переместить влево;

Клавиша D - переместить вправо;

Клавиша U - увеличить размер точки;

Клавиша I - уменьшить размер точки;

Клавиша J - увеличить толщину линии;

Клавиша K - уменьшить толщину линии;

Клавиша X - выбрать следующий примитив;

Клавиша Z - выбрать предыдущий примитив;

Клавиша N - создать новый примитив;

Клавиша C - удалить текущий примитив;

1. **Текст программы**

/\*

Отключено консольное (главное) окно:

Linker -> Advanced -> Entery Point := "mainCRTStartup"

Linker -> System -> SubSystem := "Windows (/SUBSYSTEM:WINDOWS)"

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdlib.h>

#include <glut.h>

#include <vector>

using namespace std;

GLint Width = 1268, Height = 680;

int n = 0;

/\* задание контейнера характеристик точек \*/

struct Characters

{

GLubyte ColorR = 0, ColorG = 0, ColorB = 0;

GLubyte PointSize = 5;

GLubyte LineSize = 3;

};

enum keys { Empty, KeyR, KeyG, KeyB, KeyW, KeyA, KeyS, KeyD, KeyU, KeyI, KeyZ, KeyX, KeyN, KeyJ, KeyK, KeyC};

/\* задание контейнера вершин \*/

struct type\_point

{

GLint x, y;

type\_point(GLint \_x, GLint \_y) { x = \_x; y = \_y; }

};

vector <vector <type\_point>> Points;

vector <Characters> Numb;

/\* Функция вывода на экран \*/

void Display(void)

{

glClearColor(240, 255, 240, 1); glClear(GL\_COLOR\_BUFFER\_BIT);

for (int j = 0; j < Points.size(); j++)

{

glColor3ub(Numb[j].ColorR, Numb[j].ColorG, Numb[j].ColorB);

glPointSize(Numb[j].PointSize);

glBegin(GL\_POINTS);

for (int i = 0; i < Points[j].size(); i++)

glVertex2i(Points[j][i].x, Points[j][i].y);

glEnd();

}

for (int j = 0; j < Points.size(); j++)

{

glColor3ub(Numb[j].ColorR, Numb[j].ColorG, Numb[j].ColorB);

glLineWidth(Numb[j].LineSize);

glBegin(GL\_LINE\_LOOP);

for (int i = 0; i < Points[j].size(); i++)

glVertex2i(Points[j][i].x, Points[j][i].y);

glEnd();

}

glFinish();

}

/\* Функция изменения размеров окна \*/

void Reshape(GLint w, GLint h)

{

Width = w; Height = h;

glViewport(0, 0, w, h);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(0, w, 0, h, -1.0, 1.0);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

}

/\* Функция обработки сообщений от клавиатуры \*/

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

{

int i, m = Points[n].size();

/\* Изменение RGB-компонент цвета точек \*/

if (key == 'r') Numb[n].ColorR += 5;

if (key == 'g') Numb[n].ColorG += 5;

if (key == 'b') Numb[n].ColorB += 5;

/\* Изменение XY-кординат точек \*/

if (key == 'w') for (i = 0; i<m; i++) Points[n][i].y += 5;

if (key == 's') for (i = 0; i<m; i++) Points[n][i].y -= 5;

if (key == 'a') for (i = 0; i<m; i++) Points[n][i].x -= 5;

if (key == 'd') for (i = 0; i<m; i++) Points[n][i].x += 5;

/\* Изменение размера точек \*/

if (key == 'u') Numb[n].PointSize++;

if (key == 'i') Numb[n].PointSize--;

/\* Изменение размера линии \*/

if (key == 'j') Numb[n].LineSize++;

if (key == 'k') Numb[n].LineSize--;

/\* Изменение группы примитивов\*/

if (key == 'x')

if (n + 1 < Points.size())

{

Numb[n].LineSize--;

Numb[n].PointSize--;

n++;

Numb[n].LineSize++;

Numb[n].PointSize++;

}

if (key == 'z')

if (n != 0)

{

Numb[n].LineSize--;

Numb[n].PointSize--;

n--;

Numb[n].LineSize++;

Numb[n].PointSize++;

}

/\* новая группа примитивов \*/

if (key == 'n')

{

n++;

Points.resize(n + 1);

Numb.resize(n + 1);

}

/\* удаление текущего примитива \*/

if (key == 'c')

{

if (n != 0)

{

for (int j = n; j < Points.size() - 1 ; j++)

{

Points[j] = Points[j + 1];

Numb[j] = Numb[j + 1];

}

n--;

Points.resize(Points.size() - 1);

Numb.resize(Numb.size() - 1);

}

else

{

Points.pop\_back();

Points.resize(n + 1);

}

}

glutPostRedisplay();

char v[90]; sprintf(v, "Текущий цвет всех точек: R=%.3d G=%.3d B=%.3d Текущий номер группы примитивов: %d", Numb[n].ColorR, Numb[n].ColorG, Numb[n].ColorB, n + 1);

glutSetWindowTitle(v);

}

/\* Функция обработки сообщения от мыши \*/

void Mouse(int button, int state, int x, int y)

{

/\* клавиша была нажата, но не отпущена \*/

if (state != GLUT\_DOWN) return;

/\* новая точка по левому клику \*/

if (button == GLUT\_LEFT\_BUTTON)

{

type\_point p(x, Height - y);

Points[n].push\_back(p);

}

/\* удаление последней точки по центральному клику \*/

if (button == GLUT\_MIDDLE\_BUTTON)

{

if (Points[n].size() != 0)

Points[n].pop\_back();

else

{ }

}

glutPostRedisplay();

}

void Menu(int pos)

{

int key = (keys)pos;

switch (key)

{

case KeyR: Keyboard('r', 0, 0); break;

case KeyG: Keyboard('g', 0, 0); break;

case KeyB: Keyboard('b', 0, 0); break;

case KeyW: Keyboard('w', 0, 0); break;

case KeyS: Keyboard('s', 0, 0); break;

case KeyA: Keyboard('a', 0, 0); break;

case KeyD: Keyboard('d', 0, 0); break;

case KeyU: Keyboard('u', 0, 0); break;

case KeyI: Keyboard('i', 0, 0); break;

case KeyX: Keyboard('x', 0, 0); break;

case KeyZ: Keyboard('z', 0, 0); break;

case KeyN: Keyboard('n', 0, 0); break;

case KeyK: Keyboard('k', 0, 0); break;

case KeyJ: Keyboard('j', 0, 0); break;

case KeyC: Keyboard('c', 0, 0); break;

default:

int menu\_color = glutCreateMenu(Menu);

glutAddMenuEntry("Компонента R", KeyR);

glutAddMenuEntry("Компонента G", KeyG);

glutAddMenuEntry("Компонента B", KeyB);

int menu\_move = glutCreateMenu(Menu);

glutAddMenuEntry("Вверх", KeyW);

glutAddMenuEntry("Вниз", KeyS);

glutAddMenuEntry("Bлево", KeyA);

glutAddMenuEntry("Вправо", KeyD);

int menu\_size = glutCreateMenu(Menu);

glutAddMenuEntry("Увеличить", KeyU);

glutAddMenuEntry("Уменьшить", KeyI);

int menu\_numb\_primitiv = glutCreateMenu(Menu);

glutAddMenuEntry("Следующая группа примитивов", KeyX);

glutAddMenuEntry("Предыдущий группа примитивов", KeyZ);

int menu\_info = glutCreateMenu(Menu);

glutAddMenuEntry("Клавиша R - изменить красную компоненту цвета", KeyR);

glutAddMenuEntry("Клавиша G - изменить зеленую компоненту цвета", KeyG);

glutAddMenuEntry("Клавиша B - изменить синюю компоненту цвета", KeyB);

glutAddMenuEntry("Клавиша W - переместить вверх", KeyW);

glutAddMenuEntry("Клавиша S - переместить вниз", KeyS);

glutAddMenuEntry("Клавиша A - переместить влево", KeyA);

glutAddMenuEntry("Клавиша D - переместить вправо", KeyD);

glutAddMenuEntry("Клавиша U - увеличить размер точки", KeyU);

glutAddMenuEntry("Клавиша I - уменьшить размер точки", KeyI);

glutAddMenuEntry("Клавиша J - увеличить толщину линии", KeyJ);

glutAddMenuEntry("Клавиша K - уменьшить толщину линии", KeyK);

glutAddMenuEntry("Клавиша X - выбрать следующий примитив", KeyX);

glutAddMenuEntry("Клавиша Z - выбрать предыдущий примитив", KeyZ);

glutAddMenuEntry("Клавиша N - создать новый примитив", KeyN);

glutAddMenuEntry("Клавиша C - удалить текущий примитив", KeyC);

int menu = glutCreateMenu(Menu);

glutAddSubMenu("Смена цвета", menu\_color);

glutAddSubMenu("Перемещение", menu\_move);

glutAddSubMenu("Изменение размера точки", menu\_size);

glutAddSubMenu("Изменениегруппы примитивов", menu\_numb\_primitiv);

glutAddSubMenu("Горячие клавиши", menu\_info);

glutAttachMenu(GLUT\_RIGHT\_BUTTON);

Keyboard(Empty, 0, 0);

}

}

/\* Головная программа \*/

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

{

Points.resize(n + 1);

Numb.resize(n + 1);

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_RGB);

glutInitWindowSize(Width, Height);

glutCreateWindow("Текущий цвет всех точек:");

Menu(Empty);

glutDisplayFunc(Display);

glutReshapeFunc(Reshape);

glutKeyboardFunc(Keyboard);

glutMouseFunc(Mouse);

glutMainLoop();

}

1. **Скриншоты**



