*Горшков Алексей Олегович*

*Группа 4212*

**Отчет о выполнении лабораторной работы №7 «Растровая графика»**

1. **Задание (вариант 1):**

Методами оценочной функции и ЦДА построить заданную линию:

*Отрезок прямой в заданной четверти в указанном направлении.*

1. **Листинг программы:**

#include <GL/glut.h>

#include <cmath>

//Границы координатных осей.

const int left\_side{ -10 }, right\_side{ 10 }, down\_side{ -10 }, up\_side{ 10 };

//Предобъявление функций.

void drawBresenhamLineSegment(const int& x1, const int& y1, const int& x2, const int& y2);

void drawDDALineSegment(const float& x1, const float& y1, const float& x2, const float& y2);

//Функция инициализации.

void init()

{

glClearColor(1, 1, 1, 0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(left\_side, right\_side, down\_side, up\_side);

}

//Функция рисования.

void draw()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glClearColor(1.0, 1.0, 1.0, 0.0);

glViewport(0, 0, 601, 601);

//Построение клеточек (сетки растра).

glColor3f(0.0, 0.0, 0.5);

glLineWidth(1.0);

GLdouble grid\_size{ -10.0};

glBegin(GL\_LINES);

do

{

glVertex2f(grid\_size, -10.0); glVertex2f(grid\_size, 10.0);

glVertex2f(-10.0, grid\_size); glVertex2f(10.0, grid\_size);

grid\_size += 1.0;

} while (grid\_size <= 10.0);

glEnd();

//Построение координатных осей

glColor3f(0.0, 0.0, 1.0);

glLineWidth(2.0);

glBegin(GL\_LINES);

glVertex2f(0.0, down\_side); glVertex2f(0.0, up\_side);

glVertex2f(left\_side, 0.0); glVertex2f(right\_side, 0.0);

glEnd();

/\*РАСТЕРИЗАЦИЯ ИЗОБРАЖЕНИЯ ВЕКТОРА\*/

//Построение отрезка прямой OpenGL (GL\_LINES).

glColor3f(0.0, 1.0, 0.0);

glBegin(GL\_LINES);

glVertex2i(-9, -9);

glVertex2i(-4, -1);

glEnd();

//Крайние точки выбранного отрезка прямой в III координатной четверти.

int x1{ -9 }, y1{ -9 }, x2{ -4 }, y2{ -1 };

//Построение отрезка прямой по упрощенному методу Брозенхейма (Брезенхема) - метод оценочной функции.

glColor3f(1.0, 0.0, 0.0);

glBegin(GL\_LINE\_STRIP);

drawBresenhamLineSegment(x1, y1, x2, y2);

glEnd();

//Построение отрезка прямой по методу цифровых дифференциальных анализаторов (ЦДА).

glColor3f(0.0, 0.0, 0.0);

glBegin(GL\_LINE\_STRIP);

drawDDALineSegment(x1, y1, x2, y2);

glEnd();

glFlush();

}

/\*Построение указанного отреза прямой по упрощенному алгоритму Брозенхейма (Брезенхема) - метод оценочной функции.

\*Выбранная прямая и отрезок на ней:

\*координаты крайних точек отрезка: (-9; -9) и (-4; -1);

\*отрезок находится в III координатной четверти;

\*уравнение прямой, на которой лежит отрезок: y = 1.6x + 5.4 => 5y - 8x = 27 = F(x,y) (значение '27' необходимо для алгоритма).

\*/

void drawBresenhamLineSegment(const int& x1, const int& y1, const int& x2, const int& y2)

{

//Число ячеек сетки растра по оси X = A = 5.

//Число ячеек сетки растра по оси Y = B = 8.

int A{ x2 - x1}, B{ y2 - y1 };

//F(x,y) = Ay - Bx = 5y - 8x, где (x,y) - координаты изображающей точки (ИТ).

int x{ x1 }, y{ y1 }, fXY{};

//Построение вектора по методу оценочной функции.

while (x <= x2 && y <= y2)

{

glVertex2i(x, y);

fXY = A \* y - B \* x;

if (fXY < 27)

y++;

else if (fXY > 27)

x++;

else

{

if (A >= B)

x++;

else

y++;

}

}

}

/\*Построение того же самого отрезка прямой по методу цифровых дифференциальных анализаторов (ЦДА).

\*Вычисления (по методичке):

\*F(x,y) = 1,6x - y + 5,4.

\*F'x = 1,6 и F'y = -1 => sqrt((F'x)^2 + (F'y)^2) = sqrt(1.6^2 + (-1)^2) = sqrt(89/25) = sqrt(89)/5;

\*kdt = 1/sqrt((F'x)^2 + (F'y)^2) = 5/sqrt(89);

\*dxi = -k\*F'y\*dt = 5/sqrt(89);

\*dyi = k\*F'x\*dt = 8/sqrt(89);

\*xi+1 = xi + dxi = xi + 5/sqrt(89);

\*yi+1 = yi + dyi = yi + 8/sqrt(89).

\*/

void drawDDALineSegment(const float& x1, const float& y1, const float& x2, const float& y2)

{

//Координаты изображающей точки (ИТ).

double x{ x1 }, y{ y1 };

//Приращение координат ИТ, вычесленное согласно методичке (sqrt((5/sqrt(89))^2 + (8/sqrt(89))^2) = 1).

const double dxi{ 5 / sqrt(89) }, dyi{ 8 / sqrt(89) };

//Построение вектора по методу ЦДА.

while (round(x + dxi) <= x2 && round(y + dyi) <= y2)

{

glVertex2i(round(x), round(y));

x += dxi;

y += dyi;

}

}

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

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(601, 601);

glutInitWindowPosition(50, 50);

glutCreateWindow("Lab\_7");

init();

glutDisplayFunc(draw);

glutMainLoop();

}

1. **Работа программы:**

На изображении ниже зеленым цветом изображен отрезок прямой, построенный с помощью *GL\_LINES*; красным цветом – *методом оценочной функции*; черным цветом – *методом цифровых дифференциальных анализаторов (ЦДА)* с расчетами соответствующих приращений по координатам X и Y изображающей точки (ИТ), согласно методичке:

