**Министерство образования Республики Беларусь**

**Учреждение образования**

**«Белорусский государственный университет информатики   
и радиоэлектроники»**

Кафедра интеллектуальных

информационных технологий

**Отчёт**

по лабораторной работе № 3

по курсу «Графический интерфейс интеллектуальных систем»

Выполнили:

студенты группы 821701

Бозбей М.С.

Михневич Е.Д.

Минск 2011

**Цель:**

Изучить алгоритмы построения параметрических кривых, использую форму Эрмита, Безье и B-сплайн. Реализовать графический редактор, позволяющий построение параметрических кривых. Реализовать базовые функции матричных вычислений и применить их при построении параметрических кривых.

**Алгоритм построения кривой с использованием формы Эрмита:**

render: function() {

var t = 0;

while(t < 1) {

var tVector = $M([[Math.pow(t, 3), Math.pow(t,2), t, 1],

[0,0,0,0],

[0,0,0,0],

[0,0,0,0]

]);

var x1 = tVector.multiply(this.ermitMatrix).multiply(this.pVectorX).e(1);

var y1 = tVector.multiply(this.ermitMatrix).multiply(this.pVectorY).e(1);

t += 0.1;

tVector = $M([[Math.pow(t, 3), Math.pow(t,2), t, 1],

[0,0,0,0],

[0,0,0,0],

[0,0,0,0]

]);

var x2 = tVector.multiply(this.ermitMatrix).multiply(this.pVectorX).e(1);

var y2 = tVector.multiply(this.ermitMatrix).multiply(this.pVectorY).e(1);

var wuLine = new BresenhamLine({

pointStart: new Point({x: Math.round(x1), y: Math.round(y1)}),

pointEnd: new Point({x: Math.round(x2), y: Math.round(y2)}),

color: 'red'

});

this.canvas.draw(wuLine);

}

}

**Пример работы программы:**



**Алгоритм построения кривой с использованием формы Безье:**

render: function() {

var t = 0;

while(t < 1) {

var tVector = $M([[Math.pow(t, 3), Math.pow(t,2), t, 1],

[0,0,0,0],

[0,0,0,0],

[0,0,0,0]

]);

var x1 = tVector.multiply(this.bezierMatrix).multiply(this.pVectorX).e(1);

var y1 = tVector.multiply(this.bezierMatrix).multiply(this.pVectorY).e(1);

t += 0.1;

tVector = $M([[Math.pow(t, 3), Math.pow(t,2), t, 1],

[0,0,0,0],

[0,0,0,0],

[0,0,0,0]

]);

var x2 = tVector.multiply(this.bezierMatrix).multiply(this.pVectorX).e(1);

var y2 = tVector.multiply(this.bezierMatrix).multiply(this.pVectorY).e(1);

var wuLine = new BresenhamLine({

pointStart: new Point({x: Math.round(x1), y: Math.round(y1)}),

pointEnd: new Point({x: Math.round(x2), y: Math.round(y2)}),

color: 'red'

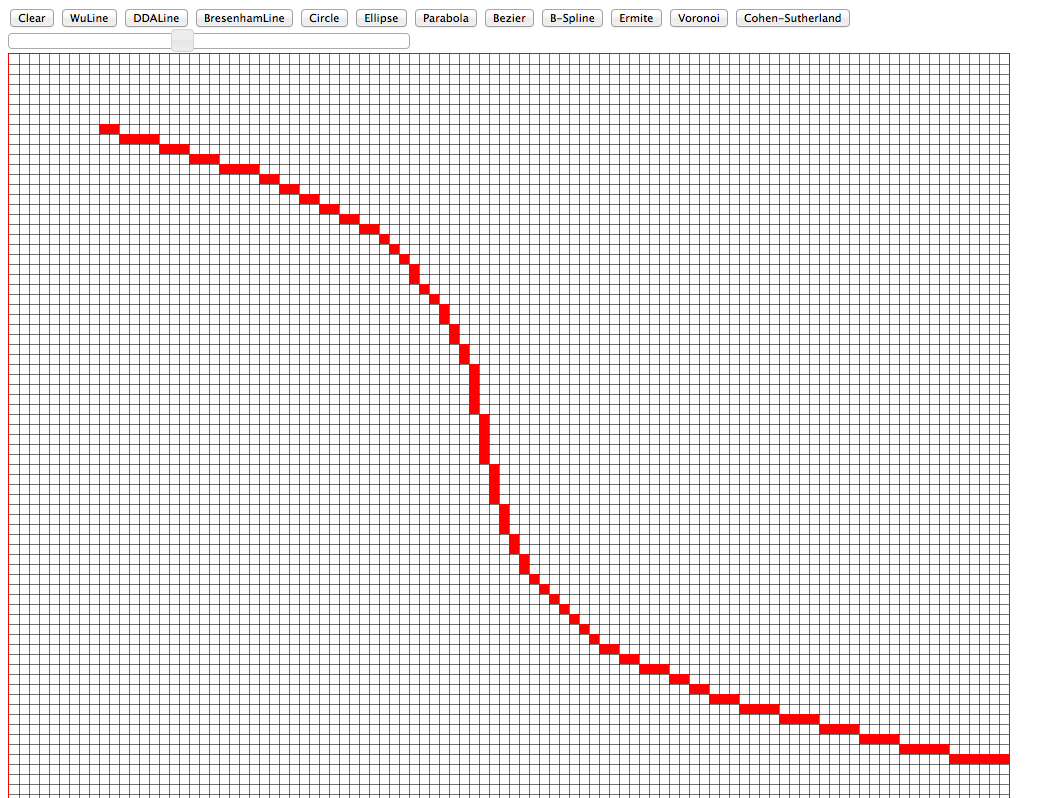
});

this.canvas.draw(wuLine);

}

}

**Пример работы программы:**



**Алгоритм построения кривой с использованием формы B-сплайн:**

function nextStepAlg() {

if (iter < cp.length - 2 && t - dt \* 1.5 >= 1) {

dx = Math.abs(cp.get((iter - 1)).get('x') - cp.get(iter).get('x');

dy = Math.abs(cp.get((iter + 1)).get('y') - cp.get((iter + 2)).get('y');

}

dt = (1f / (3 \* (dx + dy)));

data = $M([data[1 - iter][0] = cp.get(1 - 1).get("x"),

data[1 - iter][1] = cp.get(1 - 1).get("y")],

[data[2 - iter][0] = cp.get(2 - 1).get("x"),

data[2 - iter][1] = cp.get(2 - 1).get("y")],

[data[3 - iter][0] = cp.get(3 - 1).get("x"),

data[3 - iter][1] = cp.get(3 - 1).get("y")],

[data[4 - iter][0] = cp.get(3 - 1).get("x"),

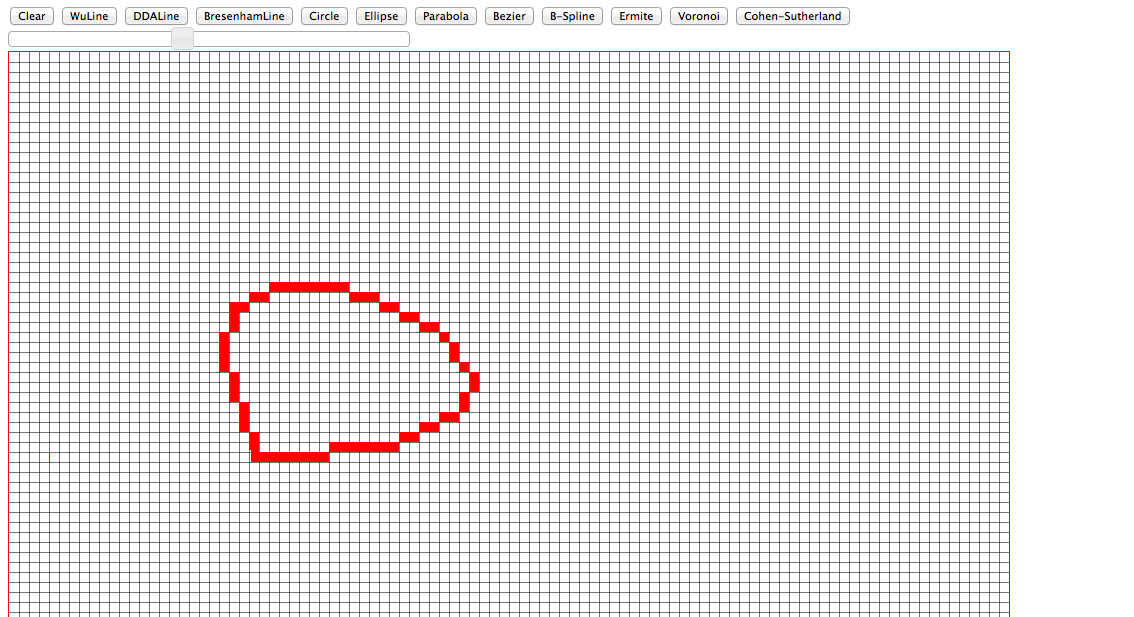
data[4 - iter][1] = cp.get(3 - 1).get("y")]]);

cur\_matrix = this.bSplineMatrix.multiply(data);

return cur\_matrix;

}

**Пример работы программы:**



**Выводы:**

В ходе лабораторной работы были изучены принципы построения параметрических кривых, изучены формы Эрмита, Безье и B-сплайн. Были реализованы матрицы и алгоритмы операций над ними. Из приведенных выше алгоритмов наиболее оптимизированным под вычислительную технику является алгоритм Безье. B-сплайн обеспечивает построение более гладких кривых, чем другие методы сглаживания за счет того, что кривые не проходят точно через заданные точки. Главный недостаток алгоритма Эрмита – неявное задание векторов.