

**Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут”**

**Факультет прикладної математики
Кафедра спеціалізованих комп'ютерних систем**

**Лабораторна робота № 1
з дисципліни «Комп'ютерна графіка»**

Виконав:
студент групи КВ-52

Затилюк Дмитро

Перевірів(ла): _____

Київ 2017

Постановка задачі

1. Реалізувати алгоритми прямої:
 - ЦДА
 - Брезенхема
 - Бу
2. Реалізувати алгоритм окружності Брезенхема

Код програми

```
function DDA(x1, x2, y1, y2) {

    console.log(x1, x2, y1, y2);

    var example = document.getElementById("example"),
        ctx = example.getContext("2d");

    example.width = 640;
    example.height = 480;

    ctx.clearRect(0, 0, example.width, example.height);

    //var x1 = 124, x2 = 560, y1 = 80, y2 = 360;

    var deltaX = Math.abs(x1 - x2);
    var deltaY = Math.abs(y1 - y2);

    var len = Math.max(deltaY, deltaX);
    // console.log(len);

    if (len === 0) {
        ctx.fillRect(x1, y1, x1+1, y1+1);
    } else {
        var dX = (x2 - x1) / len;
        var dY = (y2 - y1) / len;
        /*
        console.log(dX);
        console.log(dY);*/

        var x = x1;
        var y = y1;
        /*
            console.log(x);
            console.log(y);
        */
        len++;
        // console.log(len);

        while(len-->0) {
            x += dX;
            y += dY;

            ctx.strokeRect(Math.round(x), Math.round(y), 1, 1);
        }
    }
}
```

```

    }

}

function BR_L(x1, x2, y1, y2) {
    var example = document.getElementById("example"),
        ctx    = example.getContext('2d');

    example.width = 640;
    example.height = 480;

    ctx.clearRect(0, 0, example.width, example.height);

    //var x1 = 124, x2 = 560, y1 = 80, y2 = 360;

    var lenX = Math.abs(x2 - x1);
    var lenY = Math.abs(y2 - y1);

    var len = Math.max(lenY, lenX);

    if (len === 0) {
        ctx.fillRect(x1, y1, 1, 1);
    } else {

        var x = x1;
        var y = y1;

        len++;

        if (lenY <= lenX) {
            while(len--) {
                ctx.fillRect(x, Math.round(y), 1, 1);
                x++;
                y += lenY / lenX;
            }
        } else {
            while(len--) {
                ctx.fillRect(Math.round(x), y, 1, 1);
                x += lenX / lenY;
                y++;
            }
        }
    }
}

function WY(x1, x2, y1, y2) {
    var example = document.getElementById("example"),
        ctx    = example.getContext('2d');

    example.width = 640;
    example.height = 480;

```

```

ctx.clearRect(0, 0, example.width, example.height);

//var x1 = 124, x2 = 560, y1 = 80, y2 = 360;

var steep = Math.abs(y2 - y1) > Math.abs(x2 - x1);

if (steep) {
    [x1, y1] = [y1, x1];
    [x2, y2] = [y2, x2];
}
if (x1 > x2) {
    [x1, x2] = [x2, x1];
    [y1, y2] = [y2, y1];
}

ctx.fillRect(x1, y1, 1, 1);
ctx.fillRect(x2, y2, 1, 1);

var dX = x2 - x1;
var dY = y2 - y1;
var gradient = dY / dX;
var y = y1 + gradient;

for (var x = x1 + 1; x <= x2 - 1; x++) {
    ctx.fillStyle = "rgba(0,0,0," + (1 - (y - Math.floor(y))) + ")";
    ctx.fillRect(x, Math.floor(y), 1, 1);

    ctx.fillStyle = "rgba(0,0,0," + (y - Math.floor(y)) + ")";
    ctx.fillRect(x, Math.floor(y)+1, 1, 1);

    y += gradient;
}
}

function BR_C(x, y, R) {
    var example = document.getElementById("example1"),
        ctx = example.getContext("2d");

    example.width = 640;
    example.height = 480;

    ctx.clearRect(0, 0, example.width, example.height);

    //var x = 124, y = 80, R = 80;

    var sigma, f;
    var x1 = 0, y1 = R, yk = 0;
    var delta = 2*(1 - R);

    do{
        ctx.fillRect(x+x1, y+y1, 1, 1);
        ctx.fillRect(x-x1, y+y1, 1, 1);
    }

```

```

ctx.fillRect(x+x1, y-y1, 1, 1);
ctx.fillRect(x-x1, y-y1, 1, 1);

f = 0;
if (y1 < yk)
    break;

if (delta < 0) {
    sigma = 2*(delta + y1) - 1;
    if (sigma <= 0) {
        x1++;
        delta += 2*x1 + 1;
        f = 1;
    }
} else
if (delta > 0) {
    sigma = 2*(delta - x1) - 1;
    if (sigma > 0) {
        y1--;
        delta += 1 - 2*y1;
        f = 1;
    }
}
if (!f) {
    x1++;
    y1--;
    delta += 2*(x1 - y1 - 1);
}
}while(1);
}

var checkboxes = [document.getElementById("line_ch"),
document.getElementById("circle_ch")];

for (var i = 0; i < 2; i++) {
    checkboxes[i].onclick = function() {
        if (document.getElementById("line_ch").checked) {
            document.getElementById("lines").style.display = "block";
            document.getElementById("circle").style.display = "none";
        } else
        if (document.getElementById("circle_ch").checked) {
            document.getElementById("lines").style.display = "none";
            document.getElementById("circle").style.display = "block";
        }
    }
}

console.log("sfkljdsfkljdsf");
var buttons = document.getElementsByTagName("button");
console.log(buttons);

var x1 = document.getElementById("x1");
var x2 = document.getElementById("x2");

```

```

var y1 = document.getElementById("y1");
var y2 = document.getElementById("y2");
console.log(x1, x2, y1, y2);

var x = document.getElementById("x");
var y = document.getElementById("y");
var r = document.getElementById("R");

for (var i = buttons.length - 1; i >= 0; i--) {
    console.log(i);
    buttons[i].onclick = function() {

        switch(this.value) {
            case "1":
                DDA(+x1.value, +x2.value, +y1.value, +y2.value);
                break;
            case "2":
                BR_L(+x1.value, +x2.value, +y1.value, +y2.value);
                break;
            case "3":
                WY(+x1.value, +x2.value, +y1.value, +y2.value);
                break;
            case "4":
                BR_C(+x.value, +y.value, +r.value);
                break;
        }
        return false;
    }
}

```

Результати роботи

Draw line

Draw circle

x1

1

y1

1

x2

400

y2

200

DDA algorithm

Bresenham's algorithm

Wu's algorithm

