

Đồ họa máy tính – lab-01

Vẽ đường thẳng DDA - Bresenham

☒ Bài tập cá nhân

☐ Bài tập tự luyện

☐ Bài tập nhóm

☒ Bài tập cần nộp

Mục lục:

Vẽ đường thẳng DDA - Bresenham	1
1. Qui định về việc nộp bài:	2
1.1 Thời hạn nộp bài:	2
1.2 Cách nộp: nộp qua hệ thống courses.uit.edu.vn	2
1.3 Bài làm được nén lại thành một tập tin (.zip hoặc .rar) MSSV.zip (hoặc .rar)	2
2. Nội dung:	2
3. Yêu cầu:	6

1. Qui định về việc nộp bài:

1.1 Thời hạn nộp bài:

- Sau buổi thực hành – 01 tuần

1.2 Cách nộp: nộp qua hệ thống courses.uit.edu.vn

1.3 Bài làm được nén lại thành một tập tin (.zip hoặc .rar) MSSV.zip (hoặc .rar)

Bài nộp bao gồm :

- Source : source code và thư viện sử dụng

2. Nội dung:

Viết chương trình demo vẽ đường thẳng bằng thuật toán DDA và Bresenham với code gợi ý như sau:

1. Index.html : page html chính sẽ link tới tập tin javascript dda.js

```
1 <html>
2 <head>
3   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4   <meta charset="utf-8">
5   <title>DDA</title>
6   <link href="style/main.css" rel="stylesheet" type="text/css">
7 </head>
8 <body>
9
10  <div class="container">
11    <div class="heading">
12      <h1 class="title">DDA Algorithm</h1>
13      <h3 class="author">UIT</h3>
14    </div>
15    <p class="hint">ESC</p>
16
17    <div class="main-container">
18      <canvas id="canvas">
19      </canvas>
20    </div>
21
22    <div class="buttons">
23      <button id="reset">
24        Reset
25      </button>
26    </div>
27
28    <script type="text/javascript" src="src/dda.js"></script>
29  </div>
30
```

2. Tập tin dda.js:

- a) Khởi tạo các tham số ban đầu.
- b) Canvas được lấy theo id từ index.html

```
// Initialize

var canvas = document.getElementById("canvas");
var context = canvas.getContext("2d");

var width = 800;
var height = 600;

var bgRgba = [240, 240, 200, 255];
var pointRgba = [0, 0, 255, 255];
var lineRgba = [0, 0, 0, 255];
var vlineRgba = [255, 0, 0, 255];

canvas.setAttribute("width", width);
canvas.setAttribute("height", height);
```

- c) Tất cả các thao tác vẽ được vẽ trên imageData.

```
function Painter(context, width, height) {

    this.context = context;
    this.imageData = context.createImageData(width, height);
    this.points = [];
    this.now = [-1, -1];
    this.width = width;
    this.height = height;
}
```

d) Hàm `getPixelIndex` : lấy thông tin của pixel tại (x,y) cho trước

```
this.getPixelIndex = function(x, y) {
    if (x < 0 || y < 0 || x >= this.width || y >= this.height)
        return -1;
    return (x + y * width) << 2;
}
```

e) Hàm `setpixel`, `drawpoint`: vẽ 1 điểm tại (x,y) với màu cho trước.
 Lưu ý màu ở đây có 4 tham số: r g b và a (đọc thêm ở đây: https://www.w3schools.com/css/css_colors_rgb.asp)

```
this.setPixel = function(x, y, rgba) {
    pixelIndex = this.getPixelIndex(x, y);
    if (pixelIndex == -1) return;
    for (var i = 0; i < 4; i++) {
        this.imageData.data[pixelIndex + i] = rgba[i];
    }
}

this.drawPoint = function(p, rgba){
    var x = p[0];
    var y = p[1];
    for (var i = -1; i <= 1; i++)
        for (var j = -1; j <= 1; j++)
            this.setPixel(x + i, y + j, rgba);
}
```

f) Hàm `drawline`: hàm vẽ đường thẳng dùng thuật toán dda

```

this.drawLine = function(p0, p1, rgba) {
    var x0 = p0[0], y0 = p0[1];
    var x1 = p1[0], y1 = p1[1];
    var dx = x1 - x0, dy = y1 - y0;
    if (dx == 0 && dy == 0)
        return;
    if (Math.abs(dy) <= Math.abs(dx)) {
        if (x1 < x0) {
            var tx = x0; x0 = x1; x1 = tx;
            var ty = y0; y0 = y1; y1 = ty;
        }
        var k = dy / dx;
        var y = y0;
        for (var x = x0; x <= x1; x++) {
            this.setPixel(x, Math.floor(y + 0.5), rgba);
            y = y + k;
        }
    }
    else {
        if (y1 < y0) {
            var tx = x0; x0 = x1; x1 = tx;
            var ty = y0; y0 = y1; y1 = ty;
        }
        var k = dx / dy;
        var x = x0;
        for (var y = y0; y <= y1; y++) {
            this.setPixel(Math.floor(x + 0.5), y, rgba);
            x = x + k;
        }
    }
}
}

```

g) Các hàm xử lý sự kiện chuột và bàn phím.

```

getPosOnCanvas = function(x, y) {
    var bbox = canvas.getBoundingClientRect();
    return [Math.floor(x - bbox.left * (canvas.width / bbox.width) + 0.5),
            Math.floor(y - bbox.top * (canvas.height / bbox.height) + 0.5)];
}

doMouseMove = function(e) {
    if (state == 0 || state == 2) {
        return;
    }
    var p = getPosOnCanvas(e.clientX, e.clientY);
    painter.draw(p);
}

doMouseDown = function(e) {
    if (state == 2 || e.button != 0) {
        return;
    }
    var p = getPosOnCanvas(e.clientX, e.clientY);
    painter.addPoint(p);
    painter.draw(p);
    if (state == 0) {
        state = 1;
    }
}

doKeyDown = function(e) {
    if (state == 2) {
        return;
    }
    var keyId = e.keyCode ? e.keyCode : e.which;
    if (keyId == 27 && state == 1) { // esc
        state = 2;
        painter.draw(painter.points[painter.points.length - 1]); // clear red line
    }
}

```

3. Yêu cầu:

Dựa vào code mẫu trên các bạn hãy:

- Xây dựng thuật toán bresenham trong tập tin bresenham.js và vẽ đường thẳng có xử lý sự kiện chuột và bàn phím.
- Xây dựng thuật toán midpoint vẽ đường tròn có sử dụng chuột trong tập tin midpoint.js.
- Yêu cầu nâng cao:
Tạo menu trên index.html cho phép người dùng có thể chọn 1 trong 3 thuật toán dda, bresenham và midpoint để thao tác.