

TUGAS BESAR I : 2D PRIMITIVE ELEMENTS

GRAFIKA KOMPUTER

IF3260



Anggota - NIM:

1. Henry Anand S. R. - 13521004
2. Hidayatullah Wildan Ghaly B. - 13521015

DAFTAR ISI

DAFTAR ISI	2
BAB I - Deskripsi Program	4
1.1 Deskripsi Umum	4
1.2 Spesifikasi	4
1.3 Cakupan	4
1.3.1 Fitur Wajib	4
1.3.2 Bonus (Minimal 1)	4
BAB II - Implementasi Program	5
2.1 Kelas	5
2.1.1 Base Model	5
2.1.2 Line	5
2.1.3 Square	6
2.1.4 Rectangle	6
2.1.5 Polygon	7
2.1.6 Shape Manager	7
2.2 Fungsi	8
2.2.1 Compile Shader	8
2.2.2 InitializeWebGL	9
2.2.3 Hex to Float Color	10
2.2.4 Add Vertex Dot	10
2.2.5 isCounterClockwise	11
2.2.6 Convex Hull	11
2.2 Transformasi	13
2.2.1 Translasi	13
2.2.2 Shear	13
2.2.3 Rotasi	13
2.2.4 Perubahan koordinat titik	13
2.2.5 Memperbesar objek sebangun	13
2.2.6 Refleksi	13
BAB III - Demonstrasi Program	14
3.1.1 Cara menjalankan program	14
DAFTAR PUSTAKA	16

BAB I - Deskripsi Program

1.1 Deskripsi Umum

Tugas besar ini digunakan untuk membantu pemahaman dalam pengenalan dunia grafika komputer dengan menggunakan WebGL sebagai platform utama. Tugas ini mencakup penerapan konsep-konsep dasar dalam WebGL 2D seperti penggambaran garis dan poligon, serta transformasi geometri termasuk translasi, rotasi, dan dilatasi. Penekanan diberikan pada pemahaman aspek geometri serta pengelolaan input dan output dalam konteks pengembangan dengan WebGL.

1.2 Spesifikasi

Terdapat beberapa fitur yang harus dikembangkan dalam tugas ini antara lain

- Model
 - Garis
 - Persegi
 - Persegi Panjang
 - Polygon
- Transformasi geometri (translasi, dilatasi, rotasi, shear)
- Menggerakkan satu titik sudut dengan slider atau drag and drop
- Mengubah warna salah satu atau semua titik sudut
- Save model dan load model

1.3 Cakupan

Berikut merupakan fitur yang kami kerjakan di cakupan tugas besar ini

1.3.1 Fitur Wajib

Kami mengerjakan semua fitur wajib, sebagai berikut.

- Model
 - Garis: Ubah panjang
 - Persegi: Ubah panjang sisi
 - Persegi panjang: Ubah panjang atau ubah lebar
 - Polygon: Penambahan dan penghapusan titik sudut
- Transformasi geometri (translasi, dilatasi, rotasi, shear)
- Menggerakkan satu titik sudut dengan slider atau drag and drop
- Mengubah warna salah satu atau semua titik sudut
- Save model dan load model

1.3.2 Bonus (Minimal 1)

Kami mengerjakan bonus berikut

- Animasi
- Convex Hull (Polygon)
- Fitur penguncian pada polygon (fitur congruence)

BAB II - Implementasi Program

2.1 Kelas

2.1.1 Base Model

Kelas `base_model` merupakan kelas yang menjadi abstract class dari semua kelas shape. Kelas ini menerima beberapa parameter penting seperti WebGL context, shader, dan beberapa posisi untuk lokasi vertex dan warna. Kelas ini juga menyimpan kanvas sebagai tempat grafik akan di render dan memiliki atribut type untuk mendapatkan tipe - tipe kelas turunannya.

Base Model

Attributes:

- `gl`
- `program`
- `positionAttributeLocation`
- `colorAttributeLocation`
- `canvas`
- `positions`
- `isPressed`
- `drawMode`
- `type`
- `positionBuffer`
- `colorBuffer`
- `colors`
- `mouseMoveHandler`
- `mouseUpHandler`
- `x1`
- `y1`

Functions:

- `constructor(gl, program, positionAttributeLocation, colorAttributeLocation, canvas, type)`
- `handleMouseDown(e)`
- `resize(idx)`
- `mouseUpHandler()`
- `toggleDrawMode()`
- `activate()`
- `save()`
- `updateBuffersAndDraw()`
- `mouseMoveHandler(e)`
- `updateCoordinates()`
- `draw()`

2.1.2 Line

Line merupakan kelas turunan dari `base_model` yang digunakan untuk membuat suatu object line di WebGL. Terdapat beberapa hal penting yang menjadi atribut dari kelas ini yaitu color, position, dan beberapa atribut yang berhubungan dengan WebGL.

Line

Attributes:

- colors
- x2
- y2

Functions:

- constructor(gl, program, positionAttributeLocation, colorAttributeLocation, canvas)
- mouseMoveHandler(e)
- updateCoordinates(x1, y1, x2, y2)
- draw()
- isInside(x, y)
- bindColorBuffer(colors)

2.1.3 Square

Square merupakan kelas turunan dari base_model yang digunakan untuk pembuatan geometri persegi dari titik yang ada. Kelas ini memiliki atribut warna, x2, dan y2.

Square

Attributes:

- colors
- x2
- y2

Functions:

- constructor(gl, program, positionAttributeLocation, colorAttributeLocation, canvas)
- mouseMoveHandler(e)
- updateCoordinates(x1, y1, dx, dy)
- draw()
- isInside(x, y)
- bindColorBuffer(colors)

2.1.4 Rectangle

Rectangle merupakan kelas turunan dari base_model yang digunakan untuk pembuatan geometri persegi panjang dari titik yang ada. Kelas ini memiliki atribut warna, x2, dan y2.

Rectangle

Attributes:

- colors
- x2
- y2

Functions:

- constructor(gl, program, positionAttributeLocation, colorAttributeLocation, canvas)
- mouseMoveHandler(e)
- updateCoordinates(x1, y1, dx, dy)
- draw()
- isInside(x, y)

- `bindColorBuffer(colors)`

2.1.5 Polygon

Polygon adalah salah satu kelas turunan dari `base_model` yang dapat digunakan untuk membuat geometri dengan menambahkan tiap vertex dengan menggunakan click mouse. Kelas ini memiliki atribut `vertices` dan `warna`.

Polygon

Attributes

- `vertices`
- `colors`

Functions

- `constructor(gl, program, positionAttributeLocation, colorAttributeLocation, canvas)`
- `convertVerticesToPosition()`
- `handleMouseClicked(vertices)`
- `draw()`
- `normalizePositions()`
- `customDraw()`
- `isInside()`
- `bindColorBuffer(colors)`

2.1.6 Shape Manager

Kelas ini merupakan kelas untuk mengatur semua shape yang ada di dalam canvas dan sebagai perantara antara transformasi geometri dengan kelas - kelas geometri yang ada. Fungsi utama dari kelas ini adalah untuk menyimpan setiap shape yang terbentuk dan tiap node yang ada. Berikut merupakan fungsi dan atribut dari shape manager.

Shape Manager

Attributes

- `gl`
- `program`
- `positionAttributeLocation`
- `colorAttributeLocation`
- `canvas`
- `shapes`
- `vertices`
- `activeShape`
- `lastTranslateX`
- `lastTranslateY`
- `lastShearX`
- `lastShearY`
- `lastRotate`
- `lastScale`
- `lastDrag`
- `lastSingleTranslateX`
- `lastSingleTranslateY`

- currentVertexIndex
- currentShapeIndex
- activeType

Functions

- constructor(canvas)
- changeVertexColor(shapeIndex)
- setupShapeCreation(buttonId, shapeType)
- mouseClickedHandler(e)
- updateDotPosition()
- mouseDownHandler(e)
- mouseUpHandler(e)
- setupVertexDotEventListeners(shapeIndex)
- showColorPicker(vertexIndex, shapeIndex)
- showTranslationBars()
- mouseMoveHandler(e)
- toggleDrawModeForOtherShapes(activeShape)
- createShape(shapeType)
- clearCanvas()
- save()
- colorPickerHandler()
- load()
- loadModel(jsonStr)
- updateBuffersAndDraw()
- translateActiveShapeX(dx)
- translateActiveShapeY(dy)
- translateSingleVertexX(dx, vertexIndex, shapeIndex)
- translateSingleVertexY(dy, vertexIndex, shapeIndex)
- shearActiveShapeX(factor)
- congruenceSingleVertex(value, vertexIndex, shapeIndex)
- shearActiveShapeY(factor)
- rotateActiveShape(value)
- scaleActiveShape(value)
- reflectionActiveShape(axis)

2.2 Fungsi

2.2.1 Compile Shader

Compile shader adalah suatu fungsi yang digunakan untuk melakukan binding antar shader source dan shader. Dalam WebGL, kami menggunakan fungsi ini untuk inisialisasi vertex shader dan fragment shader.

```
export function compileShader(gl, source, type) {
  const shader = gl.createShader(type);
  gl.shaderSource(shader, source);
  gl.compileShader(shader);

  if (!gl.getShaderParameter(shader, gl.COMPILE_STATUS)) {
    console.error(
      "An error occurred compiling the shaders:",

```

```

        gl.getShaderInfoLog(shader)
    );
    gl.deleteShader(shader);
    return null;
}

return shader;
}

```

2.2.2 InitializeWebGL

Fungsi ini digunakan untuk menyiapkan environment dari WebGL seperti inisiasi shader dengan bantuan fungsi Compile Shader, inisialisasi program dan penggunaan canvas juga binding dari posisi.

```

export function initializeWebGL(canvas) {
    const gl = canvas.getContext("webgl");
    if (!gl) {
        console.error(
            "Unable to initialize WebGL. Your browser may not support it."
        );
        return null;
    }

    const vertexShaderSource = `
        attribute vec2 a_position;
        attribute vec4 a_color;
        varying vec4 v_color;
        void main() {
            gl_Position = vec4(a_position, 0, 1);
            v_color = a_color;
        }
    `;

    const fragmentShaderSource = `
        precision mediump float;
        varying vec4 v_color;
        //uniform color
        //uniform vec4 u_color;

        void main() {
            gl_FragColor = v_color;
        }
    `;

    const vertexShader = compileShader(gl, vertexShaderSource,
gl.VERTEX_SHADER);
    const fragmentShader = compileShader(
        gl,
        fragmentShaderSource,
        gl.FRAGMENT_SHADER
    );

    const program = gl.createProgram();
    gl.attachShader(program, vertexShader);
    gl.attachShader(program, fragmentShader);

```



```

gl.linkProgram(program);
gl.useProgram(program);

if (!gl.getProgramParameter(program, gl.LINK_STATUS)) {
  console.error('Unable to initialize the shader program: ' +
gl.getProgramInfoLog(program));
  return null;
}
const positionAttributeLocation = gl.getAttribLocation(program,
"a_position");

const positionBuffer = gl.createBuffer();
gl.bindBuffer(gl.ARRAY_BUFFER, positionBuffer);
const colorBuffer = gl.createBuffer();
gl.bindBuffer(gl.ARRAY_BUFFER, colorBuffer);
const colorLocation = gl.getAttribLocation(program, "a_color");
return {
  gl,
  program,
  positionAttributeLocation,
  colorLocation,
};
}

```

2.2.3 Hex to Float Color

Tujuan dari fungsi ini adalah konversi dari kode warna hexadecimal ke dalam bentuk array nilai floating point untuk komponen warna merah, hijau, dan biru. Fungsi ini akan dipanggil ketika terdapat pergantian warna oleh pengguna baik di vertex atau shape secara keseluruhan.

```

export function hexColorToFloatArray(hexColor) {
  const r = parseInt(hexColor.substr(1, 2), 16) / 255;
  const g = parseInt(hexColor.substr(3, 2), 16) / 255;
  const b = parseInt(hexColor.substr(5, 2), 16) / 255;
  return { r, g, b };
}

```

2.2.4 Add Vertex Dot

Fungsi ini digunakan untuk menambahkan sebuah div dengan border radius 50% sehingga menimbulkan efek seperti sebuah point yang akan menandai setiap vertex. Kegunaan dari div ini adalah untuk mengganti warna dari sebuah vertex dan melakukan perubahan posisi.

```

export function addVertexDot(canvas, positions, shapeIndex) {
  const canvasRect = canvas.getBoundingClientRect();
  positions.forEach((pos, i) => {
    if (i % 2 === 0) {
      const dot = document.createElement('div');
      dot.className = 'vertex-dot';
      dot.setAttribute('data-vertex-index', i / 2);
      dot.setAttribute('data-shape-index', shapeIndex);
      dot.style.position = 'absolute';
      dot.style.width = '7px';
      dot.style.height = '7px';

```

```

        dot.style.backgroundColor = 'black';
        dot.style.borderRadius = '50%';

        const dotX = canvasRect.left + window.scrollX + pos;
        const dotY = canvasRect.top + canvas.clientHeight - positions[i +
1];

        dot.style.left = `${dotX}px`;
        dot.style.top = `${dotY}px`;

        document.body.appendChild(dot);
    }
});
}

```

2.2.5 isCounterClockwise

Merupakan fungsi yang digunakan sebagai helper dari fungsi convex hull Graham Scan. Digunakan untuk mendeteksi apakah suatu titik di arah yang berlawanan jarum jam dari titik sebelumnya, hal ini akan mempengaruhi apakah titik sebelumnya di pop dari stack atau tidak.

```

export function isCounterClockwise(p1, p2, p) {
    const res = (p2.x - p1.x) * (p.y - p1.y) - (p2.y - p1.y) * (p.x - p1.x);
    return res > 0;
}

```

2.2.6 Convex Hull

Algoritma untuk menentukan convex hull adalah algoritma dari Graham Scan. Algoritma ini memiliki beberapa proses penting.

1. Definisikan titik terendah
2. Sortir semua titik selain titik terendah berdasarkan sudut bersesuaian dengan titik terendah
3. Untuk setiap titik, lihat apakah suatu titik berada di arah yang berlawanan jarum jam dari titik sebelumnya, apabila tidak maka titik sebelumnya dapat di pop dari stack
4. Untuk setiap titik masukkan ke dalam stack
5. Kembalikan stack

Berikut merupakan aplikasinya dalam program kami

```

export function convexHull(points) {
    if (points.length < 3) return points;

    // find most bottom point
    let mostBottom = points.reduce((lowest, point) => {
        return (point.y < lowest.y || (point.y === lowest.y && point.x <
lowest.x)) ? point : lowest;
    }, points[0]);

    // sort points by angle
    const sorted = points.filter(p => p !== mostBottom).sort((a, b) => {
        const angleA = Math.atan2(a.y - mostBottom.y, a.x - mostBottom.x);

```

```

        const angleB = Math.atan2(b.y - mostBottom.y, b.x - mostBottom.x);
        return angleA - angleB;
    });

    const stack = [mostBottom, sorted[0], sorted[1]];

    for (let i = 2; i < sorted.length; i++) {
        let top = stack[stack.length - 1];
        let nextToTop = stack[stack.length - 2];
        while (stack.length >= 2 && !isCounterClockwise(nextToTop, top,
sorted[i])) {
            stack.pop();
            top = stack[stack.length - 1];
            nextToTop = stack[stack.length - 2];
        }
        stack.push(sorted[i]);
    }

    return stack;
}

```

2.2 Transformasi

2.2.1 Translasi

Translasi dilakukan dengan menambah nilai x dan y dengan dx dan dy

$$\begin{aligned}x &= x + dx \\y &= y + dy\end{aligned}$$

2.2.2 Shear

Shear dilakukan dengan menambah nilai x dan y dengan faktor tertentu dengan memanfaatkan rata-rata x dan y dari bentuk yang sedang dishear

$$\begin{aligned}x &= x + (x - AVG(x)) \times factor \\y &= y + (y - AVG(y)) \times factor\end{aligned}$$

2.2.3 Rotasi

Rotasi dilakukan dengan rumus berikut

$$\begin{aligned}angle &= factor \times \frac{\pi}{180} \\x &= AVG(x) + (x - AVG(x)) \times \cos(angle) - (y - AVG(y)) \times \sin(angle) \\y &= AVG(y) + (y - AVG(y)) \times \cos(angle) + (x - AVG(x)) \times \sin(angle)\end{aligned}$$

2.2.4 Perubahan koordinat titik

Jika bangun merupakan rectangle atau garis maka index dijadikan corner kebalikan dari index saat ini untuk mendapatkan anchor (koordinat index)

$$\begin{aligned}x &= x + (x - anchor) \times factor \\y &= y + (y - anchor) \times factor\end{aligned}$$

2.2.5 Memperbesar objek sebangun

Memanfaatkan slider lalu memanggil perubahan koordinat titik x dan y bersamaan

2.2.6 Refleksi

Mendapatkan median dari x dan y sebagai batas refleksi untuk refleksi x dan y

$$\begin{aligned}x &= x - 2 \times (x - median) \\y &= y - 2 \times (y - median)\end{aligned}$$

2.2.7 Dilatasi

Dilatasi dilakukan untuk memperbesar bangun saat ini ke seluruh arah

$$\begin{aligned}x &= x + (x - AVG(x)) \times factor \\y &= y + (y - AVG(y)) \times factor\end{aligned}$$

BAB III - Demonstrasi Program

3.1.1 Cara menjalankan program

Berikut merupakan tahapan dalam menjalankan program kami.

1. Nyalakan server sesuai dengan preferensi pengguna, untuk menggunakan. Apabila komputer Anda telah terinstall Node JS server maka jalankan dengan perintah `npm install -g http-server`
2. Buka alamat di web browser

3.1.2 Cara memakai program

Berikut merupakan kegunaan dari setiap fitur yang ada di program

1. Tombol tangan untuk berpindah shape atau vertex
2. Tombol line untuk menggambar garis click drag release
3. Tombol square untuk menggambar persegi dengan click drag release
4. Tombol rectangle untuk menggambar persegi panjang dengan click drag release
5. Tombol polygon untuk menggambar poligon dengan cara membentuk beberapa titik dan akan tergambar poligon jika sudah terbentuk minimal 2 titik
6. Tombol save untuk menyimpan file dalam bentuk json
7. Tombol load digunakan untuk memuat file ke dalam canvas
8. Slider Translate X digunakan untuk melakukan translasi shape yang telah aktif dalam koordinat x
9. Slider Translate Y digunakan untuk melakukan translasi shape yang aktif dalam koordinat y
10. Slide Transform X digunakan untuk melakukan transformasi pada semua vertex terhadap sumbu X
11. Slide Transform Y digunakan untuk melakukan transformasi pada semua vertex terhadap sumbu Y
12. Slide Rotate digunakan untuk melakukan rotasi pada shape yang aktif
13. Slide Scale digunakan untuk melakukan dilatasi atau scaling pada shape yang aktif
14. Slider Translate X Single untuk melakukan translasi untuk salah satu vertex terhadap sumbu X
15. Slider Translate Y Single untuk melakukan translasi untuk salah satu vertex terhadap sumbu Y
16. Slider Congruence digunakan untuk melakukan dilatasi atau scaling terhadap suatu titik anchor sehingga bentuk tetap sebangun
17. Tombol Reflect X digunakan untuk merefleksikan terhadap sumbu X
18. Tombol Reflect Y digunakan untuk merefleksikan terhadap sumbu Y

Terdapat hal - hal penting lain yang perlu diperhatikan antara lain

1. Untuk membuat suatu geometri baru, maka klik pada salah satu tombol dan lakukan drag and drop

2. Untuk mengganti warna pada salah satu vertex, klik vertex tersebut lalu gunakan color picker yang ada di sisi kiri
3. Untuk mengganti warna pada semua vertex, klik suatu vertex pada sebuah geometri dan gunakan color picker yang ada di tengah
4. Slider Translate X, Y dan Congruence membutuhkan masukan vertex, klik salah satu vertex terlebih dahulu sebelum menggunakannya.

DAFTAR PUSTAKA

GFX Fundamental. (2012). *WebGL Fundamentals*. WebGL Fundamentals. Retrieved April 5,

2024, from <https://webglfundamentals.org>

interactivecomputergraphics.com