

Tugas Besar

IF3260 - Grafika Komputer

WebGL Part 1:
2D Primitive Elements



Disusun oleh:

Kelompok GealGeol - K1

Daniel Egiant Sitanggang	13521056
Angela Livia Arumsari	13521094
Alexander Jason	13521100

Daftar Isi

Daftar Isi	1
Deskripsi Implementasi Program	2
Hasil Implementasi	8
Manual Program	10
Pembagian Tugas	18

Deskripsi Implementasi Program

Program WebGL diimplementasikan menggunakan vanilla HTML, CSS, dan Typescript dengan tambahan library Tailwind CSS. Terdapat beberapa komponen utama dalam implementasi program, yaitu GLSL shaders, classes, utils, dan main program.

1. GLSL shaders

GLSL shaders terdiri dari vertex shader dan fragment shader yang didefinisikan pada program untuk berkomunikasi dengan GPU. Berikut adalah GLSL shaders yang didefinisikan pada program.

Vertex Shader
<pre>export const VertexShaderSource = ` attribute vec2 a_position; attribute vec4 a_color; uniform vec2 u_resolution; varying vec4 v_color; void main() { vec2 zeroToOne = a_position / u_resolution; vec2 zeroToTwo = zeroToOne * 2.0; vec2 clipSpace = zeroToTwo - 1.0; gl_Position = vec4(clipSpace, 0, 1); v_color = a_color; }`;</pre>
Fragment Shader
<pre>export const FragmentShaderSource = ` precision mediump float; varying vec4 v_color; void main() { gl_FragColor = v_color; }`;</pre>

Pada vertex shader, didefinisikan atribut `a_position` untuk menerima buffer berisi posisi titik sudut dan atribut `a_color` untuk menerima buffer berisi warna titik sudut. Variabel `gl_Position` akan dihitung berdasarkan transformasi `a_position` yang masih berupa titik koordinat pada canvas ke titik koordinat yang digunakan pada WebGL

dengan memanfaatkan `u_resolution`. Variabel `v_color` yang akan menentukan warna titik sudut pada fragment shader di-assign dengan atribut `a_color`. Pada fragment shader, variabel `gl_FragColor` akan di-assign `v_color`.

2. Classes

Pada program didefinisikan kelas `Point` serta base class `Shape` dan kelas-kelas turunannya seperti `Line`, `Square`, `Rectangle`, dan `Polygon`.

Kelas Point

Atribut	Method
<pre>public x: number; public y: number; public color: Color;</pre>	<pre>public constructor(x: number, y: number, color = [0, 0, 0, 1]) public getColor() public getCoordinate() public equals(point: Point) public setCoordinate(coordinate: number[])</pre>

Kelas `Point` mendefinisikan suatu titik koordinat pada canvas. Suatu titik koordinat akan memiliki nilai `x`, `y`, dan `color`. `Color` dari titik sudut didefinisikan dengan interface `Color` yang memiliki nilai `r`, `g`, `b`, dan `a`.

Kelas Shape

Atribut	Method
<pre>public id: number; public shapeType: ShapeType; public positions: Point[]; public tx: number; public ty: number; public degree: number; public sx: number; public sy: number;</pre>	<pre>protected constructor(id: number, shapeType: ShapeType) public getPositions() public getColors() public getCentroid() abstract getPrimitiveType(gl: WebGLRenderingContext) public render(gl: WebGLRenderingContext, bufferPos: WebGLBuffer, bufferColor: WebGLBuffer) public applyTransformation()</pre>

Kelas `Shape` merupakan base class dari seluruh primitive types shape. Atribut `shapeType` merupakan enum yang dapat bernilai `Line`, `Square`, `Rectangle`, dan `Polygon`. Atribut `positions` akan menyimpan titik sudut `Shape` dalam bentuk array of `Point`. Pada kelas `shape`, didefinisikan method yang perlu digunakan pada setiap shape, seperti `render` dan `transformation`. Fungsi `render` pada `shape` akan melakukan `render` shape sesuai dengan `positions` dan primitive type masing-masing

bentuk. Setiap terjadi suatu transformasi terhadap shape, maka fungsi `applyTransformation()` akan dipanggil. Fungsi tersebut menginisialisasi matriks transformasi 4x4 matrix menggunakan fungsi `create()`. Lalu melakukan perulangan melalui setiap posisi yang disimpan di `this.positions`. Untuk setiap posisi, akan diambil koordinatnya menggunakan metode `getCoordinate()`. Lalu akan membuat vektor 2D vector yang akan ditransformasikan menggunakan matriks transformasi. Terakhir akan diperbarui koordinat posisi dengan koordinat yang telah ditransformasi menggunakan metode `setCoordinate()`.

Kelas Vector

Atribut	Method
<pre>public x: number; public y: number;</pre>	<pre>public constructor(x: number, y: number) Static create() Static fromValues(x: number, y: number) public transformMat(matrix: Float32Array) public toArray()</pre>

Kelas Vector mendefinisikan suatu vektor pada matriks transformasi. Suatu vektor koordinat akan memiliki nilai x, y.

Kelas Matrix

Atribut	Method
<pre>public data: Float32Array;</pre>	<pre>public constructor() Static create()</pre>

Kelas Matrix mendefinisikan suatu matriks identitas untuk transformasi. Suatu matrix hanya memiliki sebuah elemen matrik berupa array.

Kelas Line

Atribut	Method
<pre>public length: number; public oriLength: number;</pre>	<pre>public constructor(id: number, point: Point) public getPrimitiveType(gl: WebGLRenderingContext): number public setEndPoint(point: Point) public movePoint(dLength: number, pointIndex: number) public updateLength(newLength: number)</pre>

Kelas Line yang merupakan turunan dari kelas Shape dengan tambahan atribut length dan oriLength. Atribut length akan menyimpan panjang dari Line. Sedangkan oriLength akan menyimpan panjang sebelum dilakukan pemindahan titik sudut. Ketika kelas Line diinisialisasi, koordinat akan menjadi titik awal. Sedangkan, titik akhir akan diatur melalui method setEndPoint. Fungsi updateLength digunakan untuk fitur khusus Line yaitu perubahan panjang. Sedangkan, movePoint digunakan untuk memindahkan salah satu titik sudut Line.

Kelas Square

Atribut	Method
<pre>public center: Point; public length: number; public diagonal: number; public oriPositions: Point[];</pre>	<pre>public constructor(id: number, point: Point) public getPrimitiveType(gl: WebGLRenderingContext): number public updatePoint(point: Point) public updateLength(newLength: number) public movePoint(dLength: number, pointIndex: number)</pre>

Kelas Square yang merupakan turunan dari kelas Shape memiliki atribut tambahan center, length, diagonal, dan oriPositions. Ketika user menginisialisasi kelas Square, koordinat akan menjadi titik tengah Square. Selanjutnya akan dilakukan komputasi koordinat titik sudut lainnya berdasarkan salah satu koordinat titik sudut. Fungsi updateLength digunakan untuk melakukan perubahan panjang sisi Square. Sedangkan fungsi movePoint digunakan untuk pemindahan salah satu titik sudut Square.

Kelas Rectangle

Atribut	Method
<pre>public firstRef: Point null; public secondRef: Point null;</pre>	<pre>public constructor(id: number, points: Point[]) public getPrimitiveType(gl: WebGLRenderingContext): number public arrangePositions()</pre>

Kelas Rectangle yang merupakan turunan dari kelas Shape memiliki atribut tambahan firstRef dan secondRef. Kedua atribut tersebut merupakan titik koordinat yang membangun diagonal dari persegi panjang/ *rectangle* tersebut. Selanjutnya akan dilakukan komputasi koordinat titik sudut lainnya berdasarkan diagonal tersebut.

Kelas Polygon

Atribut	Method
	<pre>public constructor(id: number, points: Point[]) public getPrimitiveType(gl: WebGLRenderingContext): number public arrangePositions() public deletePoint(point: Point)</pre>

Kelas Polygon yang merupakan turunan dari kelas Shape tidak memiliki atribut tambahan apapun. Kelas polygon hanya melakukan rekonstruksi pada kumpulan points atribut positions untuk dapat menciptakan bidang convex dengan algoritma *grahams scan*. Selain itu, terdapat metode deletePoint untuk menghapus point pada poligon.

3. Utils

Utils web-gl

Method	Penjelasan
<pre>export function resizeCanvasToDisplaySize(canvas: HTMLCanvasElement)</pre>	Menyesuaikan lebar dan tinggi canvas sesuai dengan display
<pre>export function createShader(gl: WebGLRenderingContext, type: number, source: string)</pre>	Membuat WebGL Shader berdasarkan source berupa source code GLSL
<pre>export function createProgram(gl: WebGLRenderingContext, vertexShader: WebGLShader, fragmentShader: WebGLShader)</pre>	Membuat WebGL Program berdasarkan WebGLRenderingContext, vertex shader, dan fragment shader
<pre>export function bindBuffer(gl: WebGLRenderingContext, arr: Float32Array, buffer: WebGLBuffer)</pre>	Melakukan bind arr dengan WebGLBuffer

Utils save-load

Method	Penjelasan
<pre>export const loadFile = (): Promise<Shape[]></pre>	Melakukan pembacaan file input dengan format .json dan return isi file berupa array of Shape

<code>const objectToShape = (objects: ShapeObject[]): Shape[]</code>	Melakukan mapping dari object json ke array of Shape
<code>const objectToPoint = (object: Point)</code>	Melakukan mapping dari object json ke object Point
<code>const objectsToPoints = (objects: Point[])</code>	Melakukan mapping dari object json ke array of Point

Utils tools

Method	Penjelasan
<code>function componentToHex(c: number)</code>	Berfungsi mengubah sebuah sebuah angka menjadi hexadecimal
<code>function rgbToHex(rgba: number[]): string</code>	Berfungsi mengubah nilai array rgb menjadi hex string
<code>function hexToRgb(hex: string):number[]</code>	Berfungsi mengubah nilai hex string menjadi array rgb

Utils transformations

Method	Penjelasan
<code>function rotate(shape: Shape, angleInDegrees: number)</code>	Berfungsi memutar bangun datar sesuai dengan sudut yang diinginkan
<code>function translate(shape: Shape, tx: number, ty: number)</code>	Berfungsi mengubah posisi bangun datar sesuai dengan posisi yang diinginkan
<code>function scale(shape: Shape, sx: number, sy: number)</code>	Berfungsi mengubah ukuran bangun datar sesuai dengan skala yang diinginkan

Utils animation

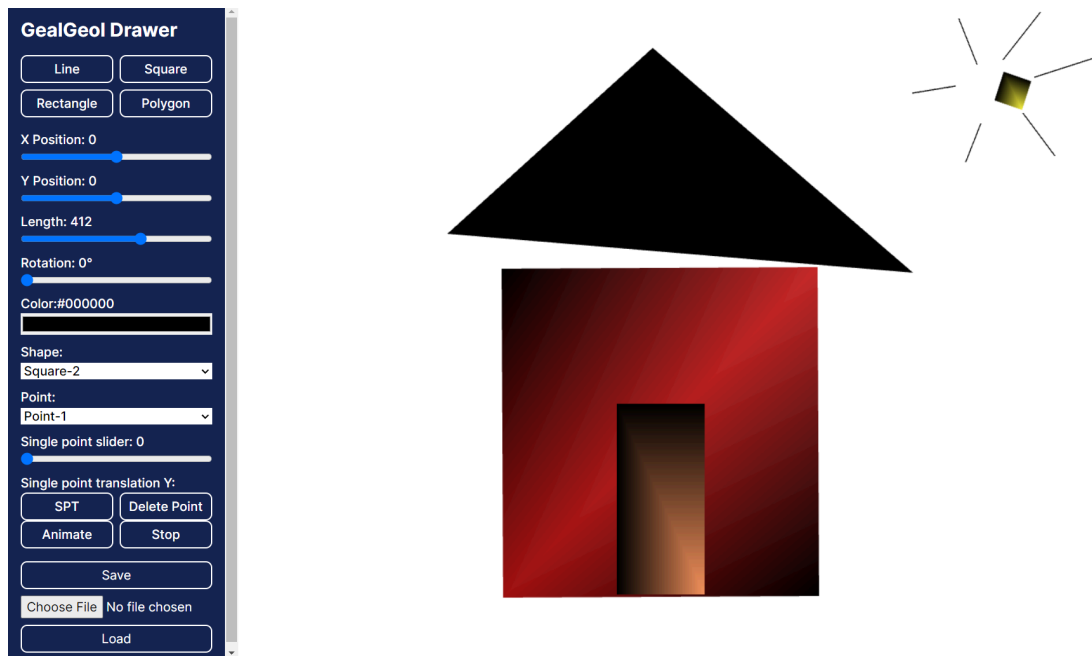
Method	Penjelasan
<code>function animateRotation(shape: Shape, angle: number): void</code>	Berfungsi memutar bangun datar untuk keperluan animasi

4. Main program

Program main sebagai program utama akan membuat program WebGL dari fungsi yang telah disediakan. Program ini akan mendefinisikan event listener dari element HTML yang ditampilkan. Dari event listener ini, program akan memanggil fungsi yang berkaitan pada masing-masing shape. Setiap perubahan pada shape, akan dipanggil fungsi `renderCanvas()` yang berfungsi untuk melakukan render ulang tiap shape pada canvas.

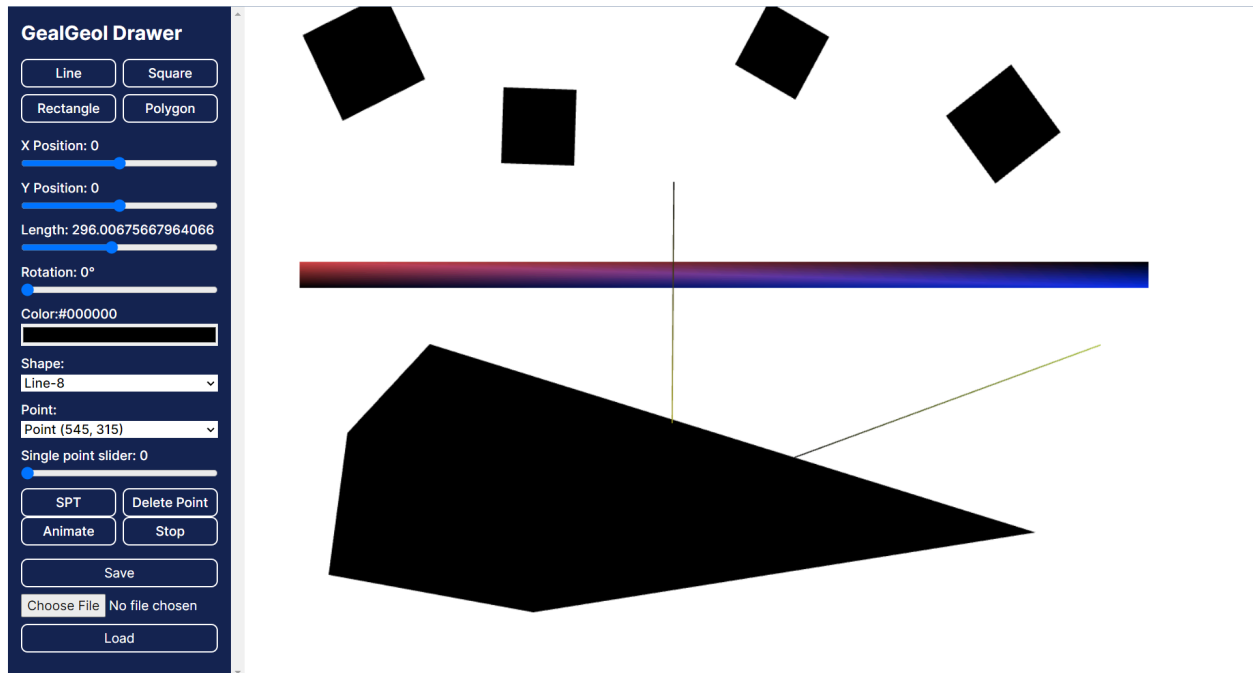
Hasil Implementasi

Berikut adalah contoh hasil load dari model1.json pada folder test di repository.



Hasil gambar di atas dibentuk dari seluruh bentuk dasar yang ada. Atap rumah dibentuk dengan menggambar Polygon dengan tiga titik sudut. BAngunan rumah sendiri digambar dengan bentuk Square. Pintu rumah digambar dengan bentuk Rectangle. Matahari pada gambar tersebut digambar dengan kombinasi Square dan Line. Warna-warna dari bentuk pada gambar tersebut diatur dengan memilih color pada setiap titik sudut di tiap bentuk.

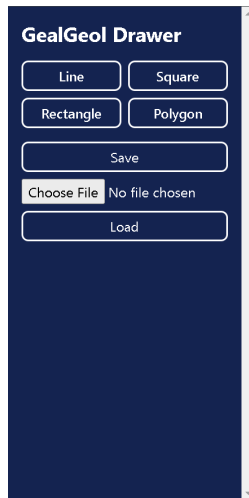
Berikut adalah contoh hasil load dari model2.json pada folder test di repository.



Hasil gambar di atas juga dibentuk menggunakan seluruh bentuk yang tersedia, yaitu Line, Square, Rectangle, dan Polygon. Penggambaran polygon di sini menggunakan metode convex hull. Dalam melakukan penggambaran, dapat dilakukan transformasi seperti translasi, perubahan panjang, dan rotasi. Titik sudut dari suatu bentuk juga dapat digeser. Terdapat juga animasi pada bentuk-bentuk pada canvas. Animasi ini akan membuat shape yang sedang aktif melakukan rotasi.

Manual Program

Tampilan awal program



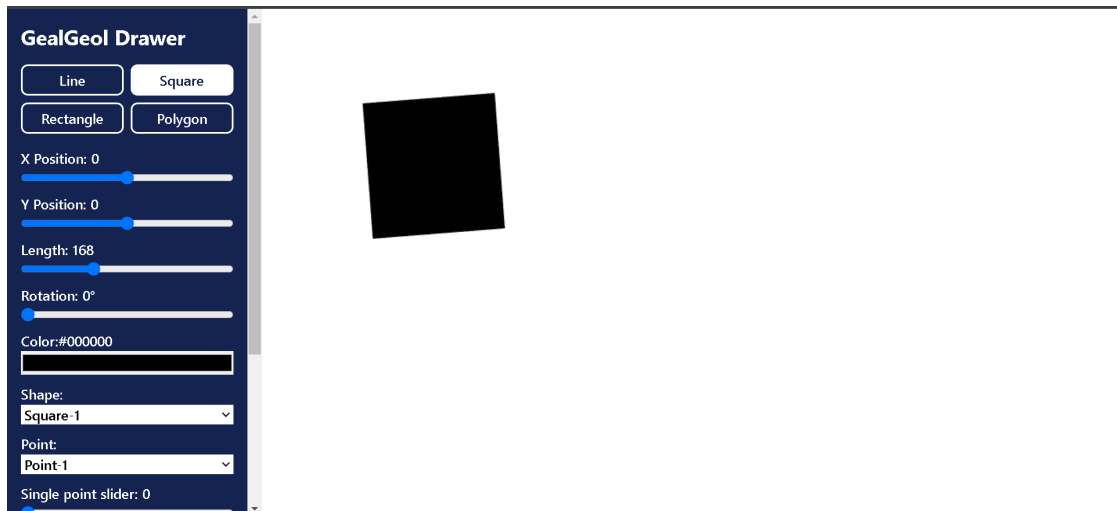
1. Membuat Garis

Klik button “Line” pada menu di sebelah kiri, klik ke dalam “canvas” dan klik lagi untuk mengakhiri pembuatan garis.



2. Membuat Persegi

Klik button “Square” pada menu di sebelah kiri, klik ke dalam “canvas” dan klik lagi untuk mengakhiri pembuatan persegi.



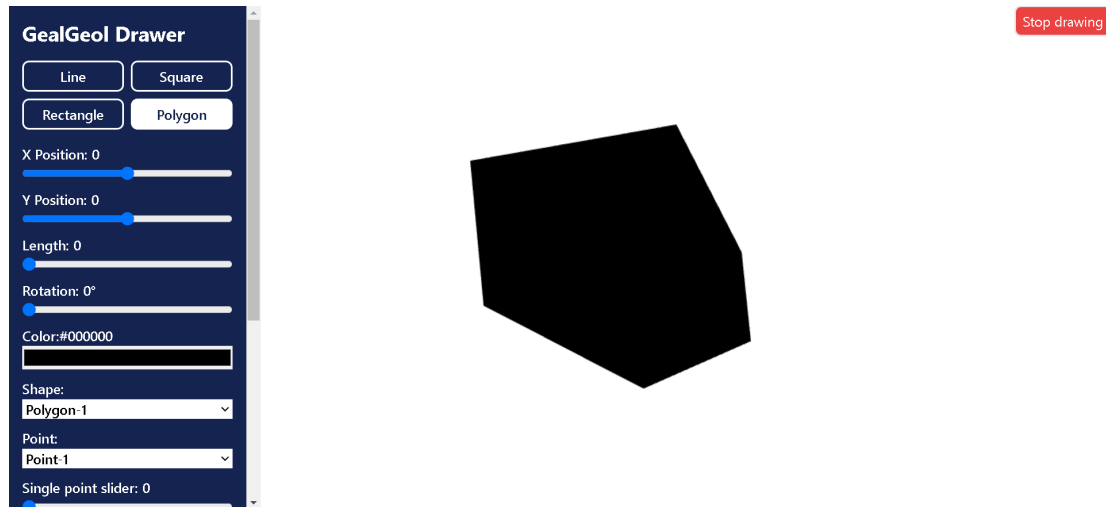
3. Membuat Persegi Panjang

Klik button “Rectangle” pada menu di sebelah kiri, klik ke dalam “canvas” dan klik lagi untuk mengakhiri pembuatan persegi panjang.



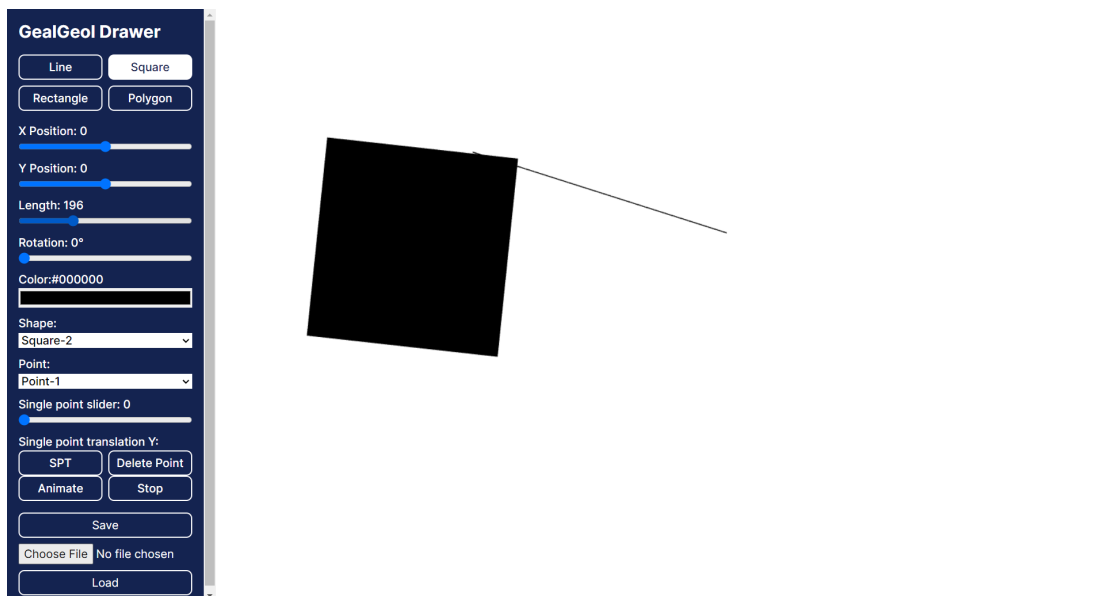
4. Membuat Polygon

Klik button “Polygon” pada menu di sebelah kiri, klik ke dalam “canvas” dan klik minimal 3 kali untuk membuat bentuk. Selanjutnya, klik pada kanvas untuk menambahkan titik sudut Polygon. Tekan tombol “Stop Drawing” untuk selesai menggambar Polygon.



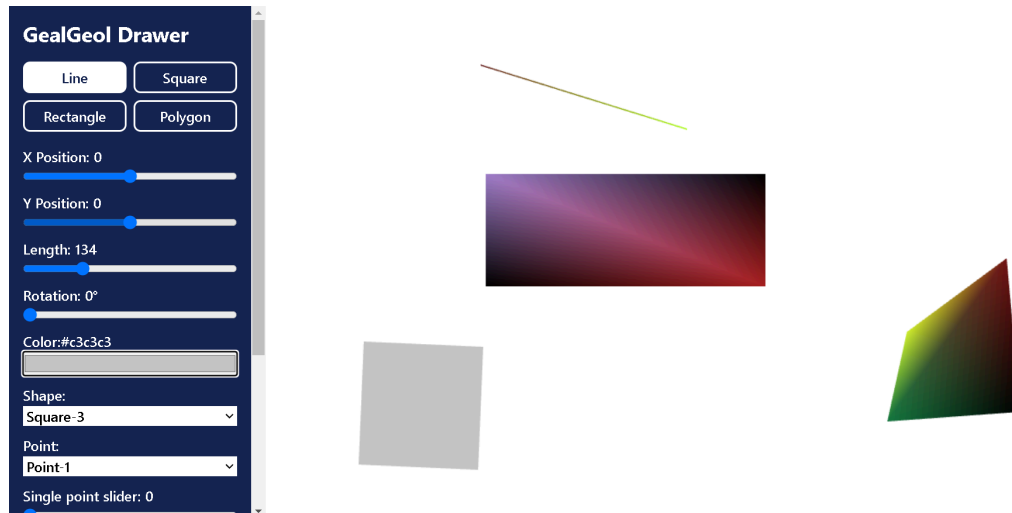
5. Mengubah ukuran Garis, Persegi, dan Persegi Panjang

Untuk mengubah panjang garis maupun panjang sisi persegi, dapat menggunakan slider "Length" yang tersedia. Sedangkan untuk persegi panjang, dapat mengubah panjang melalui slider "Length" dan lebar melalui slider "Width".



6. Mengubah warna titik sudut pada Garis, Persegi, Persegi Panjang, dan Poligon

Pilih salah satu "shape" pada "List of shapes" lalu pilih "point" yang tersedia, selanjutnya pilih warna pada bagian "Color".



7. Transformasi

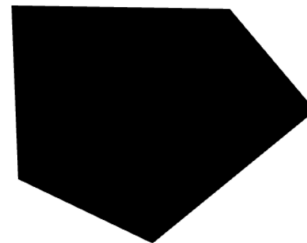
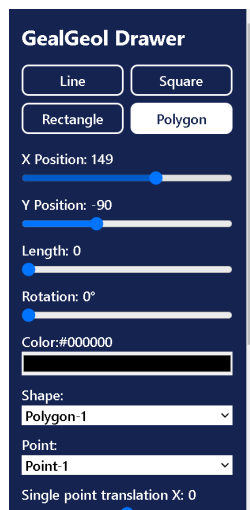
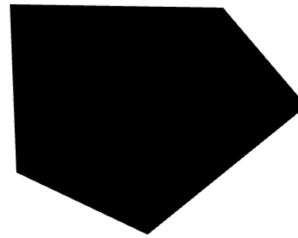
- Translasi

Pilih salah satu "shape" pada "Shape" lalu gunakan "slider" pada bagian "Slider X" atau "Slider Y" untuk melakukan translasi.

Sebelum:

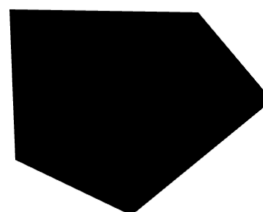


Sesudah:



- Rotasi
- Pilih salah satu "shape" pada "Shape" lalu gunakan "slider" pada bagian rotation

Sebelum:

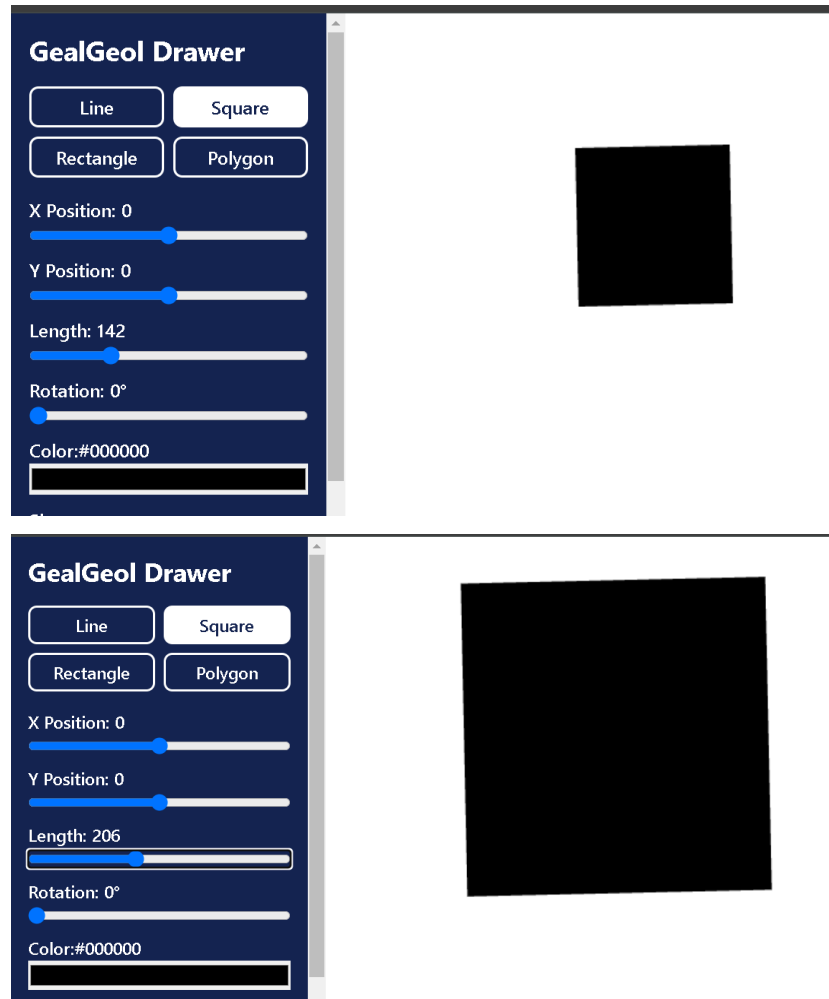


Sesudah:



- **[BONUS]** dilatasi

Untuk mengubah panjang garis maupun panjang sisi persegi, dapat menggunakan slider "Length" yang tersedia. Perubahan ini tetap mempertahankan kesebangunan. Hal ini diimplementasikan pada fungsi 'scale' pada utils transformation yang menggunakan matrix transformasi dalam perubahan panjang bangun

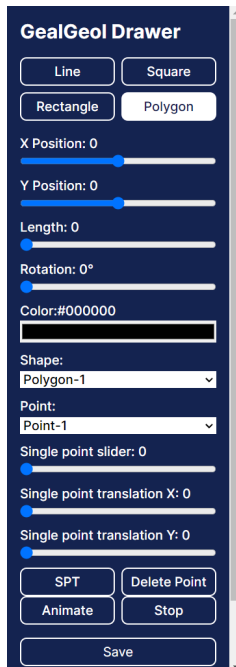


8. Export dan Import File

Export canvas ke dalam bentuk file dapat dilakukan dengan tombol Save. Canvas akan disimpan dalam file format json. File tersebut dapat di-import kembali ke dalam program. Setelah melakukan import, pengguna dapat melanjutkan melakukan perubahan pada shape tersebut.

9. **[BONUS]** Menambah titik sudut pada bentuk poligon (convexhull)

Ketika melakukan penggambaran bentuk Polygon, bentuk yang dihasilkan akan selalu berupa convex hull dari seluruh titik bukan bentuk berdasarkan urutan penggambaran titik.



10. **[BONUS]** Animasi rotasi

Pilih salah satu "shape" pada "Shape" lalu klik "Animate" untuk memulai animasi, dan "Stop" untuk menghentikan animasi.

Link video:

[Video Animasi Rotasi](#)

Pembagian Tugas

Nama Anggota	Kontribusi
Daniel Egiang Sitanggang (13521056)	<ol style="list-style-type: none">1. Implementasi bangun Rectangle dan Polygon2. Implementasi perubahan lebar dan panjang Rectangle3. Implementasi penghapusan titik Polygon4. Implementasi Convex Hull5. Implementasi slider titik sudut Rectangle dan Polygon
Angela Livia Arumsari (13521094)	<ol style="list-style-type: none">1. Setup WebGL program, GLSL, dan base class2. Implementasi bangun Line dan Square3. Implementasi perubahan panjang Line dan sisi Square4. Implementasi slider titik sudut Line dan Square
Alexander Jason (13521100)	<ol style="list-style-type: none">1. Implementasi transformasi bangun (translasi, rotasi, dilatasi)2. Implementasi pengubahan warna pada titik sudut3. Implementasi perubahan panjang garis4. Implementasi animasi rotasi