

WebGL Fundamentals

Video: https://youtu.be/_V0nabKV6k

Modul I: Translation

Source: <https://webglfundamentals.org/webgl/lessons/webgl-2d-translation.html>

Pada modul ini, eksperimen yang dilakukan adalah melakukan translasi terhadap bentuk geometri 2D. Langkah-langkah yang dilakukan untuk bisa mengimplementasikan translasi terhadap bangun geometri 2D yaitu, pertama-tama inisialisasi vertex shader dan fragment shader. Pada vertex shader, terdapat global variable `u_translation`, yang menyimpan nilai translasi yang akan diterapkan. Nilai translasi ini nantinya akan ditambahkan dengan `a_position` sebagai posisi akhir setelah translasi.

Untuk mengupdate nilai translasi, pada file script, akses uniform `u_translation` untuk mendapatkan posisi atau nilai translasi, kemudian update dengan nilai yang didapat dari slider. Setelah nilai diupdate maka fungsi `drawScene` akan dipanggil untuk menampilkan bangun geometri setelah dilakukan translasi terhadapnya.



Sering perubahan pada slider, nilai `u_translation` akan terus diupdate, kemudian mengupdate posisi akhir dari bangun geometri 2D tersebut.

Modul II: Rotation

Source: <https://webglfundamentals.org/webgl/lessons/webgl-2d-rotation.html>

Pada modul ini, eksperimen yang dilakukan adalah melakukan rotasi terhadap bentuk geometri 2D. Langkah-langkah yang dilakukan untuk bisa mengimplementasikan rotasi terhadap bangun geometri 2D yaitu, pertama-tama inisialisasi vertex shader dan fragment shader. Pada vertex shader, ditambahkan global variable `u_rotation` yang menyimpan koordinat `x,y` rotasi. Nilai `x` dan `y` rotasi ini didapatkan dari operasi trigonometri terhadap angle yang diterapkan pada slider. `x` adalah nilai $\cos(\text{angle})$ sedangkan `y` adalah nilai $\sin(\text{angle})$. Setelah mendapatkan nilai `x,y` rotasi, nilai tersebut dikalikan dengan posisi dari bangun geometri yang disimpan dalam uniform `a_position` dengan operasi sebagai berikut

```
rotatedX = a_position.x * u_rotation.y + a_position.y * u_rotation.x;  
rotatedY = a_position.y * u_rotation.y - a_position.x * u_rotation.x;
```

Untuk mendapatkan nilai `u_rotation` untuk `x` dan `y`, pada file script, `getUniformLocation(u_rotation)` untuk mendapatkan `u_rotation`, kemudian nilai tersebut di update berdasarkan angle yang didapat dari slider. Setelah mendapatkan angle, nilai `u_rotation` diupdate dengan nilai $\sin(\text{angle})$ untuk `u_rotation.y` dan $\cos(\text{angle})$ untuk `u_rotation.x`.

Kemudian posisi bangun geometri tersebut diupdate dengan nilai (`rotatedX`, `rotatedY`). Pada eksperimen ini, karena diterapkan pula translasi, maka yang digunakan sebagai posisi akhir adalah nilai setelah ditambahkan besaran translasi pada `u_translation`.



Modul III: Scale

Source: <https://webglfundamentals.org/webgl/lessons/webgl-2d-scale.html>

Pada modul ini, eksperimen yang dilakukan adalah melakukan scaling terhadap bentuk geometri 2D. Langkah-langkah yang dilakukan untuk bisa mengimplementasikan translasi terhadap bangun geometri 2D yaitu, pertama-tama inisialisasi vertex shader dan fragment shader. Pada vertex shader, ditambahkan global variable `u_scale` yang menyimpan nilai scale atau pengali seberapa besar atau kecil yang akan diterapkan terhadap ukuran semula. Nilai scale ini nantinya akan dikalikan dengan posisi bangun geometri 2D tersebut.

Untuk mengupdate nilai faktor scale, pada file script, akses uniform `u_scale` untuk mendapatkan posisi atau faktor scale untuk sumbu x dan y, kemudian update dengan nilai yang didapat dari slider. Setelah nilai diupdate maka fungsi `drawScene` akan dipanggil untuk menampilkan bangun geometri setelah dilakukan scale terhadapnya.



Sering perubahan pada slider, nilai `u_scale` akan terus diupdate, kemudian mengupdate posisi akhir dari bangun geometri 2D tersebut.