

WebGL: 2D Translation

1. Translasi merupakan sebuah transformasi grafik yang digunakan untuk menggerakkan suatu objek, baik secara vertikal maupun pada orientasi horizontal.
2. Pada dasarnya, segala bentuk transformasi yang dilakukan secara interaktif pada suatu objek WebGL akan memerlukan *re-rendering* atau penggambaran ulang objek tersebut secara keseluruhan. Dalam hal ini, WebGL mampu memberikan kesan yang *smooth* karena proses ini dapat dilakukan secara cepat.
3. Pada implementasinya, akan diperlukan sebuah fungsi JavaScript yang akan digunakan untuk melakukan *re-rendering* tersebut dengan menerima nilai-nilai terkait, seperti misalnya besaran translasi yang dilakukan, sesuai dengan keadaan yang ada pada saat itu. Di dalam proyek yang dibuat, fungsi ini misalnya adalah fungsi *drawScene*.
4. Program JavaScript juga tentunya perlu menggunakan *event callback* yang sesuai dengan interaksi antara pengguna dan objek interaktif yang disediakan, dan dalam hal ini fungsi *event* tersebut adalah memperbarui nilai-nilai yang ada sesuai dengan interaksi yang dilakukan terhadap objek tertentu.
5. Ada dua cara untuk memberikan pembaruan nilai tersebut kepada WebGL. Pertama, nilai tersebut dihitung terlebih dahulu untuk setiap titik yang ada pada JavaScript. Tentunya, hal ini cukup merepotkan, terlebih ketika jumlah titik yang digambarkan sangat banyak.
6. Oleh karena itu, maka lebih baik jika besaran translasi juga diberikan kepada *vertex processor* melalui *buffer* seperti biasa, lalu melakukan perhitungan titik yang baru pada GLSL sesuai dengan besaran tersebut. Hal ini akan jauh memudahkan pemrogram, karena granularitas logika yang digunakan di dalam kode program akan menjadi lebih rendah.

WebGL: 2D Rotation

1. Rotasi merupakan perubahan orientasi sudut pada suatu objek dari posisinya semula.
2. Pada WebGL, rotasi menggunakan prinsip lingkaran satuan yang memiliki jari-jari (mirip seperti *clip space*).
3. Perkalian posisi setiap simpul dengan posisi suatu titik yang ada pada lingkaran satuan akan menyebabkan pergeseran titik tersebut sesuai dengan besaran sudut yang terbentuk. Hal ini tentu tidak menyebabkan perubahan ukuran dari objek yang dirotasi.
4. Ada dua cara untuk menerapkan rotasi yang interaktif pada WebGL dengan menggunakan komponen-komponen HTML. Cara pertama adalah dengan menggunakan pustaka *GMAN Unit Circle* yang telah memberikan dukungan terhadap pemrograman JavaScript melalui JQuery. Cara kedua adalah cara yang lebih primitif, yaitu menggunakan *slider* biasa untuk mengganti besaran sudut putar yang akan diterapkan pada objek. Pada pendekatan kedua ini, maka nilai yang diberikan kepada WebGL harus diubah terlebih dahulu ke dalam satuan radian dengan menggunakan konsep trigonometri.

WebGL: 2D Scaling

1. *Scaling* merupakan transformasi yang digunakan untuk mengubah ukuran dari suatu benda.
2. Pada WebGL, *scaling* dilakukan dengan mengubah posisi dari titik-titik yang akan digambarkan pada kanvas.
3. Perkalian titik dengan nilai yang negatif akan menyebabkan objek yang digambarkan menjadi terbalik.

Proyek Tutorial WebGL

1. Berikut merupakan [tautan](#) dari *repository* yang digunakan untuk menampung hal-hal yang berkaitan dengan proyek WebGL ini.
2. Video singkat mengenai penjelasan dari proyek ini dapat diakses melalui tautan [YouTube](#) berikut. Mohon maaf apabila durasi videonya melebihi batas yang diberikan.
3. *Repository* tersebut berisikan beberapa proyek yang dilakukan berdasarkan *tutorial* WebGL, meliputi *WebGL Fundamentals*, *WebGL How It Works*, dan *WebGL Image Processing*. Di dalam *repository* ini juga terdapat rujukan kepada *repository* lain yang berisikan proyek WebGL lalu yang pernah dikerjakan. Selain itu, terdapat tambahan berupa proyek-proyek terkait dengan tugas ini, yaitu *Translation*, *Rotation*, dan *Scaling*.
4. Teknik-teknik serta konsep yang digunakan dalam pembuatan proyek-proyek tersebut telah dijelaskan pada bagian-bagian di atas.