TUGAS 2
3D WEBGL ARTICULATED MODEL

Diajukan untuk memenuhi nilai tugas 3
Mata Kuliah IF3260 Grafika Komputer

Dosen Pengampu: Dr. Judhi Santoso, M.Sc.

Dibuat Oleh:

Ilham Prasetyo Wibowo	13520013
Maharani Ayu Putri Irawan	13520019
Muhammad Akmal Arifin	13520037
Yohana Golkaria Nainggolan	13520053

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

2023

BAB I
PENDAHULUAN

Dalam tugas ini, dibuat sebuah aplikasi berbasis web untuk melakukan manipulasi pada *articulated model* yang juga dibuat. Kode dibuat dalam bahasa Javascript dan menggunakan API WebGL. WebGL merupakan API Javascript untuk me-*render* grafika 2 dan 3 dimensi menggunakan browser. Kami menggunakan browser Google Chrome dengan mengaktifkan WebGL developer extension yang telah tersedia.

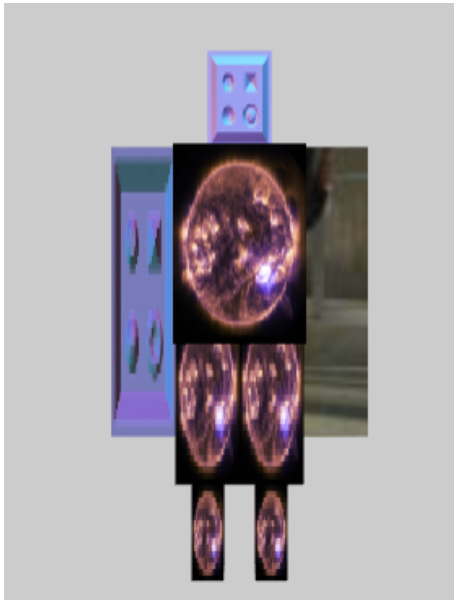
Kode sumber yang telah dibuat, dapat diakses melalui https://github.com/ilhamwibowo/IF3260_Tugas3_K01_G07. Untuk menjalankan program, diperlukan perangkat dengan browser yang mendukung WebGL developer extension. *Clone repository* pada direktori lokal, lalu buka jalankan *file* index.html dengan LiveServer, yang dapat diunduh sebagai *extension* VS Code.

BAB II
DESKRIPSI DAN FITUR

A. Articulated Objects

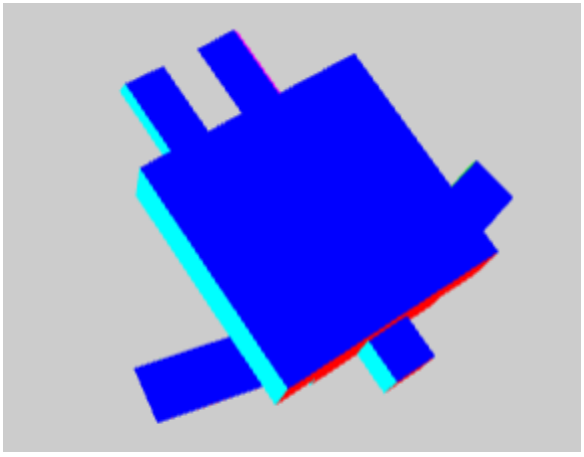
Terdapat 4 *articulated model* yang dibuat, yakni:

1. Human
   Dibuat oleh: Ilham Prasetyo Wibowo (13520013)



Gambar 1. Human

2. Turtle
   Dibuat oleh: Maharani Ayu Putri Irawan (13520019)

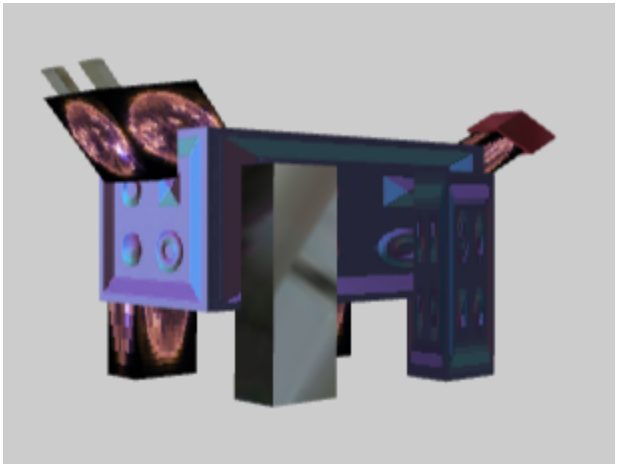


Gambar 2. Turtle

3. Snow Golem
   Dibuat oleh: Muhammad Akmal Arifin (13520037)

Gambar 3. Snow Golem

4. Cat
   Dibuat oleh: Yohana Golkaria Nainggolan (13520053)



Gambar 4. Cat

B. Input & Output

Ada 2 fungsi input dan output yang dibuat, yakni *load* dan *save*.

1. Load
   Fungsi *load* ada 3, yaitu:
   a. Load Models
      Fungsi *load model* digunakan untuk menggambar objek 3 dimensi pada *canvas* dengan *name*, *vertices, colors, indices, normals, tangents, bitangents, textureCoord*, *textureMode*, dan *children* yang didefinisikan pada file berformat JSON dengan struktur:

Gambar 5. Struktur file masukan berformat JSON

Terdapat 5 file objek yang sudah tersedia pada subfolder `src/modules/models`, yakni *cube* serta keempat *articulated object* pada poin sebelumnya. Tampilan awal ketika membuka website menunjukkan model *human*. Model-model lain dapat di-*load* dengan:

a. Mengklik Choose File lalu memilih file model yang tersedia
b. Mengklik Load

Gambar 6. Fungsionalitas Load

Ketika event klik dideteksi pada tombol load, dilakukan pembacaan file yang dibaca dari input file, lalu dibuat Object3D dengan *name*, *vertices, colors, indices, normals, tangents, bitangents, textureCoord*, *textureMode*, dan *children* hasil bacaan file.

b. Load Components

Fungsi *load model* digunakan untuk menggambar bagian objek 3 dimensi yang terpilih pada *canvas* dengan *name*, *vertices, colors, indices, normals, tangents, bitangents, textureCoord*, *textureMode*, dan *children* yang didefinisikan pada file berformat JSON dengan struktur:

Gambar 7. Fungsionalitas Load Component

Komponen dapat di-*load* dengan:

c. Mengklik Choose File lalu memilih file komponen yang tersedia
d. Mengklik Load

Gambar 8. Fungsionalitas Load

Ketika event klik dideteksi pada tombol load, dilakukan pembacaan file yang dibaca dari input file, lalu dibuat Object3D dengan *name*, *vertices, colors, indices, normals, tangents, bitangents, textureCoord*, *textureMode*, dan *children* hasil bacaan file.

2. Save

Untuk menyimpan Object3D pada *canvas* beserta transformasi yang telah dilakukan, dapat digunakan fungsionalitas *save*. Untuk melakukan penyimpanan, langkah yang dapat dilakukan sebagai berikut:

a. Untuk menyimpan dengan nama file tertentu, masukkan nama file yang diinginkan (tanpa ekstensi .json) pada input teks di sebelah kiri tombol Save. Bila tidak dimasukkan, nama file default adalah articulatedmodel.json.
b. Klik tombol Save. Object3D akan tersimpan di Downloads/

Gambar 9. Fungsionalitas Save

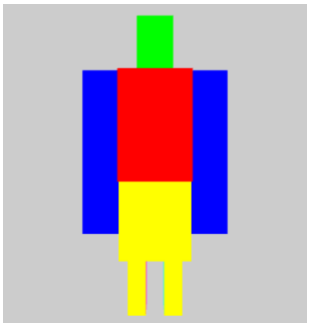
Fungsi Save diimplementasikan dengan terlebih dahulu mengaplikasikan matriks transformasi pada seluruh *vertices*, *normals, tangents,* dan *bitangents*. Lalu, dibuat file JSON dan diisikan *name*, *vertices, colors, indices, normals, tangents, bitangents, textureCoord*, *textureMode*, dan *children* Object3D tersebut. Terakhir, file disimpan dengan Blob dan *create URL* untuk di download.

C. Proyeksi

Terdapat 3 jenis proyeksi yang diimplementasikan, berikut dicobakan pada *hollow cube*.

1. Perspective

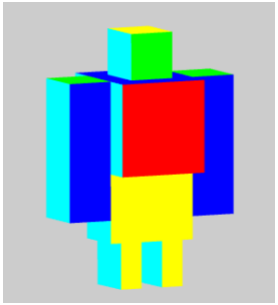
   Merupakan jenis proyeksi default. Titik yang terletak jauh (menuju sumbu z-) terlihat mengecil. Canvas juga dibatasi pada x = {-1, +1}, y = {-1, +1}, dan z = {-1, +1}. Objek akan di-clip saat melebihi 0.1 di depan dan 100 di belakang. Berikut merupakan hasil perspective projection.



Gambar 10. Perspective projection pada human dengan scale sumbu z

2. Orthographic

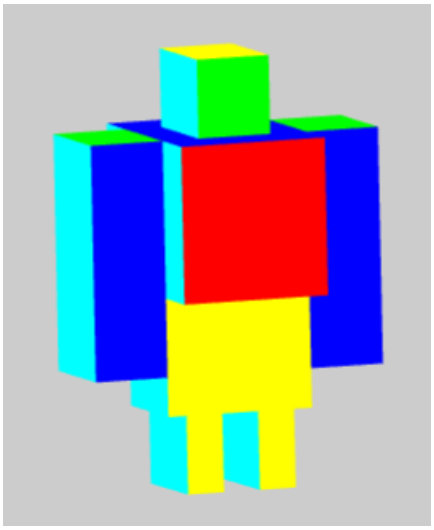
   Pada proyeksi ini, sisi-sisi 3D ditampakkan sebagai sejumlah sisi-sisi 2D. Berikut merupakan hasil orthographic projection.



Gambar 11. Orthographic projection human dan rotation x-y

3. Oblique

   Proyeksi oblique diimplementasikan dengan melakukan proyeksi orthographic dan melakukan shear lalu sedikit ditranslasikan ke sumbu x, y, dan z. Berikut merupakan hollow cube yang sama dengan poin sebelumnya direpresentasikan dengan proyeksi oblique.



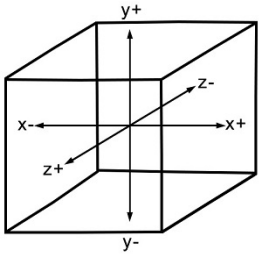
Gambar 12. Oblique projection pada hollow cube Gambar 8

D. Transformasi

Ada 3 jenis transformasi yang diimplementasikan pada ketiga sumbu, x, y, dan z.

1. Translasi

   Menggeser posisi Object3D, baik ke arah sumbu x, y, maupun z dengan slider masing-masing sumbu. Pergeseran ke kanan menandakan sumbu positif merujuk pada gambar berikut.



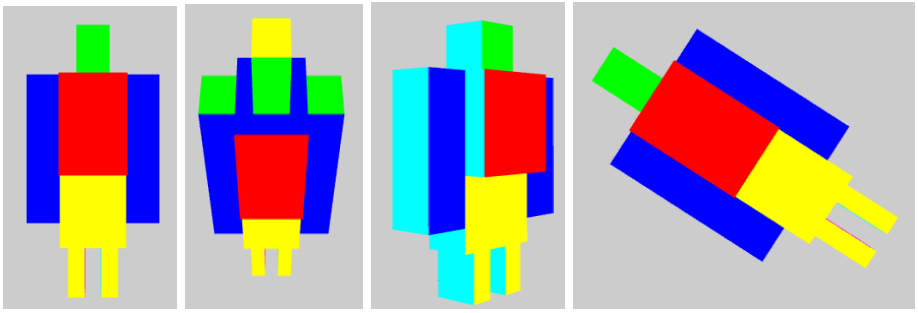
Gambar 13. Sumbu acuan (Sumber: TutorialsPoint)

Berikut contoh posisi hasil translasi ke arah sumbu x+, y+, dan z- dengan human yang di-*reset default*.

Gambar 14. Hasil translasi sesuai slider

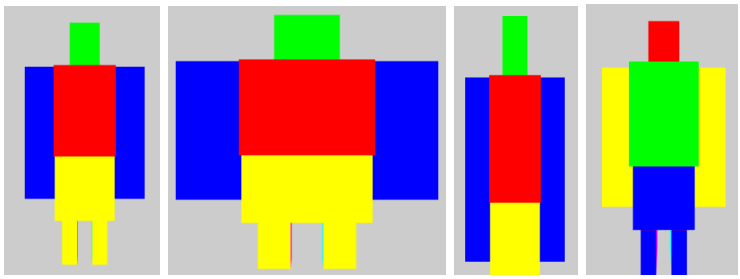
Translasi diimplementasikan dengan mengalikan matriks proyeksi dengan matriks translasi dengan nilai selisih slider dengan nilai slider sebelumnya.

2. Rotasi
   Rotasi digunakan untuk merotasi Object3D pada sumbu x, y, dan z. Slider ke kanan merepresentasikan nilai positif. Berikut merupakan contoh rotasi hollow object masing-masing ke sumbu x, y, dan z.



Gambar 15. Perbandingan hollow object awal, setelah rotasi sumbu x, y, dan z.

   Rotasi diimplementasikan dengan mengalikan matriks proyeksi dengan matriks rotasi masing-masing sumbu dengan nilai selisih slider dengan nilai slider sebelumnya.
3. Scaling
   Scaling digunakan untuk melebar/rampingkan objek, meninggikan/memendekkan objek, serta menebalkan/menipiskan objek (secara berurutan pada sumbu x, y, dan z). Untuk melakukan scaling, dapat menggeser slider masing-masing sumbu. Untuk membesarkan, geser ke arah luar. Semakin tengah, objek akan semakin kecil. Berikut merupakan contoh hasil scaling.



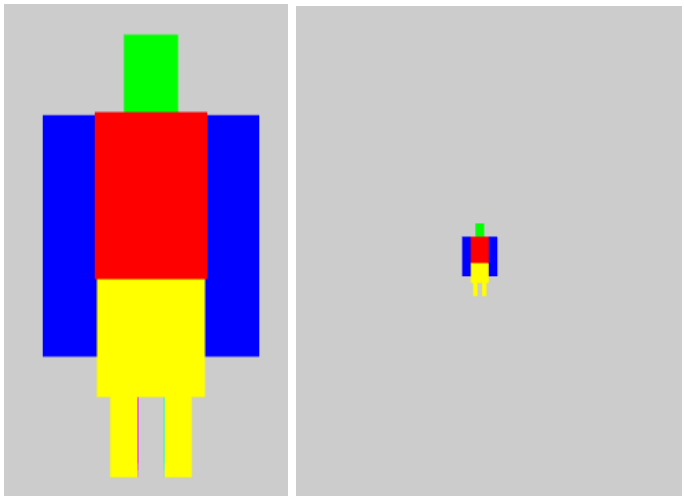
Gambar 16. Perbandingan hollow object awal, setelah scaling sumbu x, y, dan z

   Scaling diimplementasikan dengan mengalikan matriks proyeksi dengan matriks scaling masing-masing sumbu dengan nilai proporsi slider saat ini dengan pembacaan slider sebelumnya.

E. Kamera

Kamera digunakan untuk “melihat” objek pada jarak dan sudut tertentu. Oleh karena itu, terdapat 2 transformasi kamera, yakni perubahan jarak dan sudut.

1. Camera View Radius
   Untuk mengubah jarak pandang kamera, digunakan slider Camera View Radius. Pergeseran ke kanan memperjauh jarak kamera dan sebaliknya. Berikut merupakan perbandingan view awal dengan menjauhkan jarak pandang kamera.

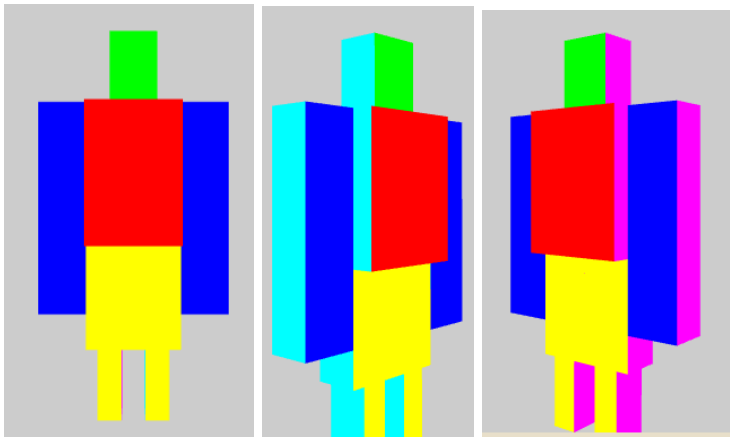


Gambar 17. Jarak kamera default dan setelah diperjauh

Perubahan jarak kamera diimplementasikan dengan membuat transformasi matriks tersendiri untuk menentukan posisi kamera dengan rotasi pada sumbu y, lalu mentranslasikan matriks itu pada sumbu z sejauh radius yang bernilai selisih slider saat itu dengan nilai slider sebelumnya.

2. Camera View Angle

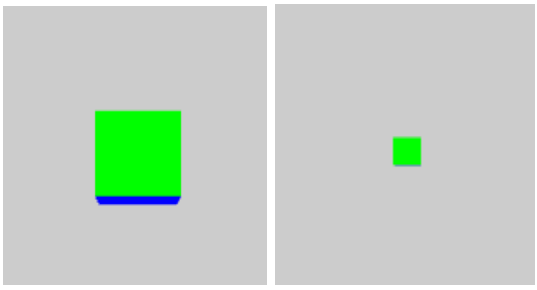
   Untuk mengubah sudut pandang kamera, digunakan slider Camera View Angle. Pergeseran ke kanan menggeser kamera memutari titik 0,0,0 ke arah kanan, dan sebaliknya. Selain itu, pergeseran angle dapat juga dilakukan dengan menggunakan tombol *arrow right* dan *arrow left* pada keyboard. *Arrow right* digunakan untuk menggeser kamera ke kiri, dan sebaliknya untuk *arrow left*. Berikut merupakan perbandingan awal, setelah digeser ke kanan, dan setelah digeser ke kiri.



Gambar 18. Posisi kamera awal, setelah digeser ke kanan, dan setelah digeser ke kiri

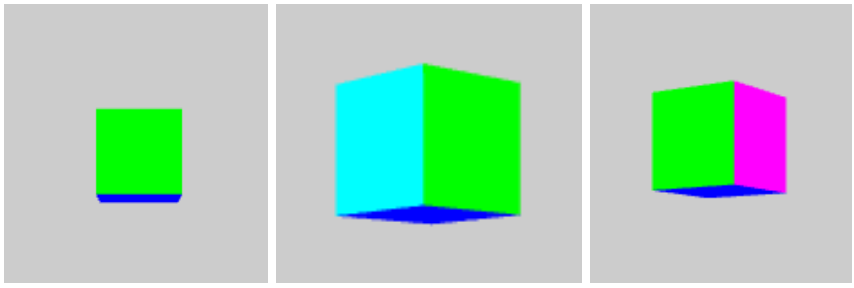
Perubahan angle kamera diimplementasikan dengan merotasikan matriks transformasi kamera pada sumbu y dengan nilai sudut dalam radius sesuai nilai slider saat itu. Nilai slider dibatasi antara -360 derajat hingga 360 derajat.

3. Camera View Radius Object
   Untuk mengubah jarak pandang kamera terhadap bagian objek yang dipilih, digunakan slider Camera View Radius Object. Pergeseran ke kanan memperjauh jarak kamera dan sebaliknya. Berikut merupakan perbandingan view awal dengan menjauhkan jarak pandang kamera.



Gambar 19. Jarak kamera default dan setelah diperjauh

   Perubahan jarak kamera diimplementasikan dengan membuat transformasi matriks tersendiri untuk menentukan posisi kamera dengan rotasi pada sumbu y, lalu mentranslasikan matriks itu pada sumbu z sejauh radius yang bernilai selisih slider saat itu dengan nilai slider sebelumnya.
4. Camera View Angle Object
   Untuk mengubah sudut pandang kamera terhadap bagian objek yang dipilih, digunakan slider Camera View Angle Object. Pergeseran ke kanan menggeser kamera memutari titik 0,0,0 ke arah kanan, dan sebaliknya. Selain itu, pergeseran angle dapat juga dilakukan dengan menggunakan tombol *arrow right* dan *arrow left* pada keyboard. *Arrow right* digunakan untuk menggeser kamera ke kiri, dan sebaliknya untuk *arrow left*. Berikut merupakan perbandingan awal, setelah digeser ke kanan, dan setelah digeser ke kiri.



Gambar 20. Posisi kamera awal, setelah digeser ke kanan, dan setelah digeser ke kiri

   Perubahan angle kamera diimplementasikan dengan merotasikan matriks transformasi kamera pada sumbu y dengan nilai sudut dalam radius sesuai nilai slider saat itu. Nilai slider dibatasi antara -360 derajat hingga 360 derajat.

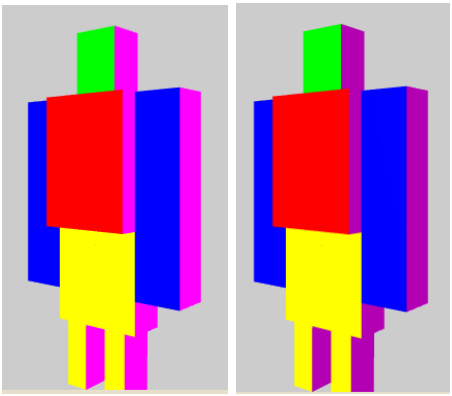
F. Reset

Reset digunakan untuk mengembalikan *state* objek ke awal sebelum dilakukannya transformasi. Untuk melakukan reset, pengguna dapat mengklik tombol Set Default.

Fungsionalitas reset diimplementasikan dengan me-reset matriks transformasi, serta mengembalikan nilai slider ke nilai default dan menginisialisasi nilai variabel kamera dengan nilai default.

G. Shading

Shading diimplementasikan menggunakan model Phong, dengan mempertimbangkan tiga refleksi yaitu ambient, diffuse, dan specular. Perhitungan warna shading dilakukan di *fragment shader.* Warna *ambient, diffuse,* dan *specular* didefinisikan di dalam kode *fragment shader.* Berikut merupakan tampilan Hollow Cube sebelum dan sesudah shading diaktifkan.



Gambar 21. Objek Hollow Cube sebelum dan sesudah ditambahkan shading.

Pewarnaan *shading* dipengaruhi oleh normal dari tiap sisi. Untuk *diffuse,* dihitung intensitas cahaya dengan dot product antara normal dan arah datang cahaya yang kemudian dikalikan dengan warna bidang dan warna *diffuse.* Kemudian *ambient light* juga ditambahkan untuk menambahkan sedikit cahaya pada setiap sisi, warna *ambient light* diaplikasikan pada seluruh warna rgb sesuai masukan. Kemudian specular dihitung dengan memangkatkan dot antara normal dengan *halfway vector,* dengan koefisien *shininess.* Shading bisa diaktifkan atau dinonaktifkan dengan mengklik *checkbox* "Enable shading" pada *website.*

H. Animasi

Animasi diimplementasikan sebagai modul mandiri yang terpisah dari *articulated model*. Seluruh model yang tersedia bisa menggunakan animasi manapun yang telah dibuat. Disediakan 4 animasi, berpindah secara melingkar, pergerakan berenang, berputar mengelilingi pusat, serta efek membesar dan mengecil (*bouncy).* Animasi yang ada didefinisikan dalam file berformat JSON dengan struktur sebagai berikut,

```
{
    "numKeyframes": 12,
    "keyframes": [
        {
            "translation": [1.153, 0.16, 0],
            "rotation": [0, 0, 0],
            "scale": [1, 1, 1]
        },
        {
            "translation": [1.421, 0.32, 0],
            "rotation": [0, 0, 0],
            "scale": [1, 1, 1]
        },
```


Gambar 22. Contoh format file animasi

Animasi dapat berjalan pada model dengan terlebih dahulu melakukan *load* animasi. Pada panel kanan, klik Choose File pada loader animasi (bagian tengah). Pilih file animasi pada folder animation/ lalu klik Open, kemudian klik Load. Untuk menjalankan animasi, klik Play. Untuk menghentikannya, klik tombol Stop, yang tadinya merupakan tombol Play. Untuk memulai ulang animasi dari state sebelumnya, klik Resume. Untuk menghentikannya, klik Pause. Untuk memulai ulang dari state awal, klik Play.

Animasi dimainkan dalam rate FPS tertentu yang dihitung dengan 1000 / timeBetweenFrames. Nilai batas timeBetweenFrames adalah 0…100. Pergerakan objek sesuai transformasi animasi diimplementasikan dengan mengalikan matriks transformasi model dengan matriks transformasi animasi per-*frame*. Untuk melihat tampilan animasi, dapat juga dilakukan dengan menggunakan *frame modifier* berikut.

Gambar 23. Frame modifier

Untuk mengubah *rate frame per second,* dapat digunakan *slider* berikut.

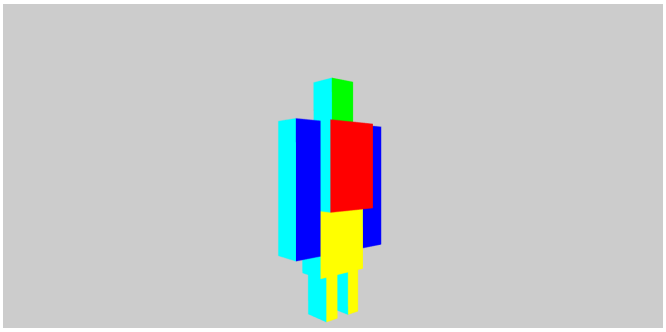
Gambar 24. Slider Time Between Frames

I. Tekstur

Tekstur pada objek model dapat diatur dengan memilih pengatur tekstur pada bagian pengaturan model. Pemilihan tekstur dibagi menjadi dua, *Select texture* mengatur tekstur yang ada pada bagian objek tertentu dan *select texture s* mengatur tekstur yang ada pada bagian objek tertentu beserta objek anakannya.

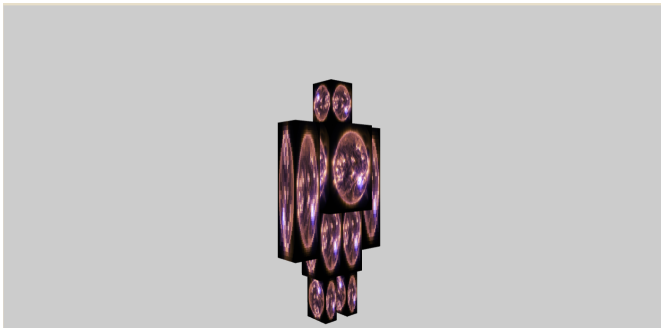
Pilih jenis texture yang ingin ditampilkan pada object yang terlihat di canvas.

- None: Tanpa texture, hanya warna dasar



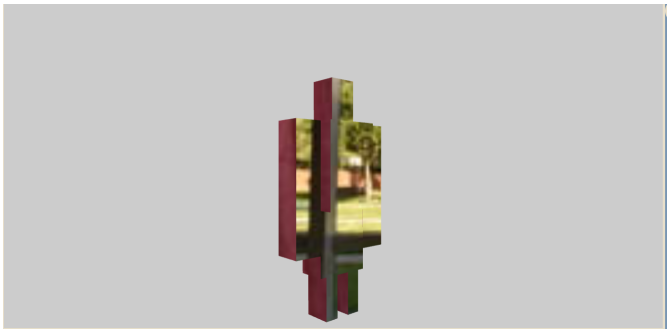
Gambar 25. Tanpa tekstur

- Image: Menampilkan gambar sebagai texture object



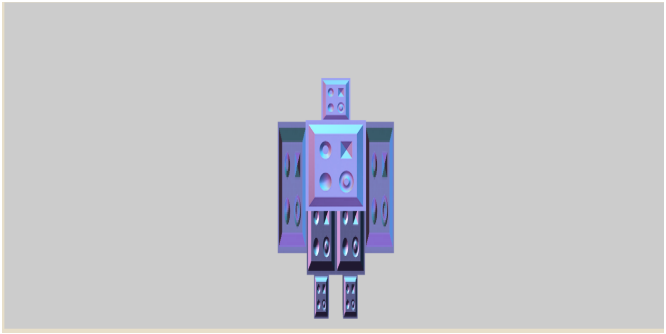
Gambar 26. Image Texture Mapping

- Environment: Membuat object dapat menampilkan pantulan dari lingkungan yang ada disekitar object



Gambar 27. Environment Mapping

- Bump: Membuat object memiliki texture yang tidak rata



Gambar 28. Bump Mapping

J. Bantuan

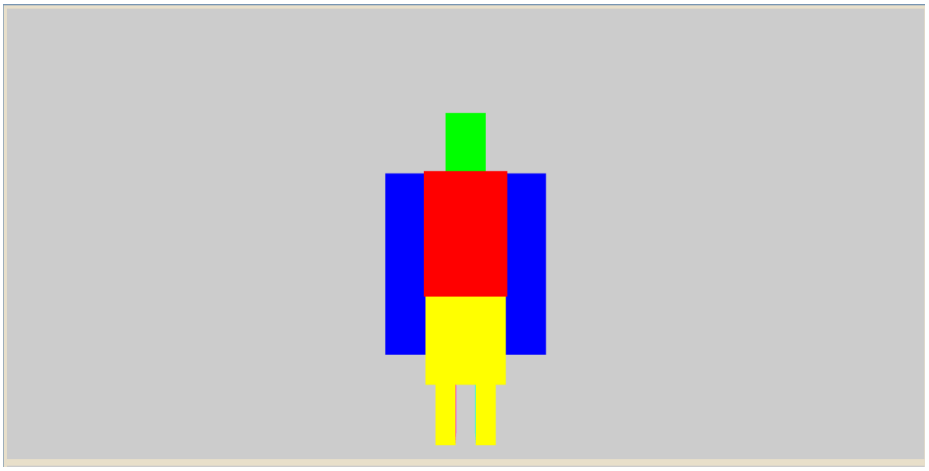
Bantuan dapat diakses dengan mengklik tombol Help. Bantuan ini menampilkan deskripsi fungsionalitas yang tersedia pada website. Bantuan ditampilkan dalam bentuk modal pop-up.

K. Kanvas

Terdapat 2 kanvas yang ditampilkan pada *website,* yaitu kanvas untuk seluruh model, dan kanvas yang hanya menampilkan subtree dari model.

1. Kanvas Model

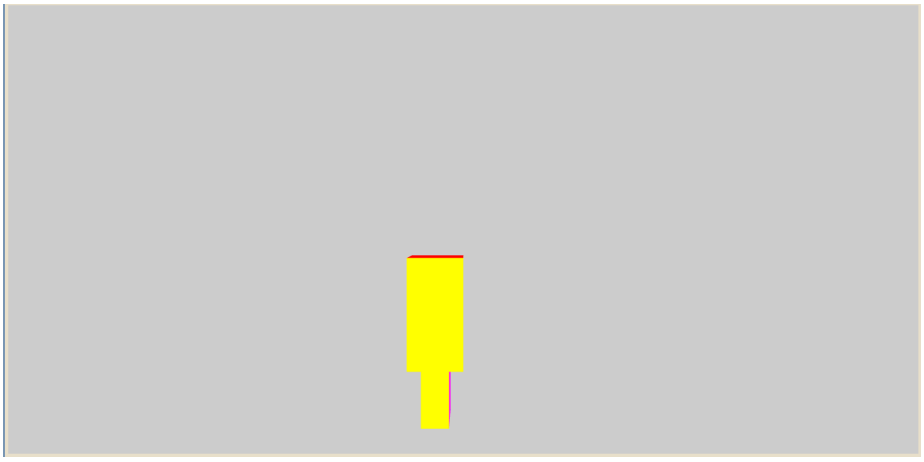
   Kanvas menampilkan articulated model yang telah di load. Pada canvas ini bisa dilakukan transformasi tehadap model dan mengubah jarak kamera dan rotasi kamera.



Gambar 29. Kanvas Model

2. Kanvas Subtree

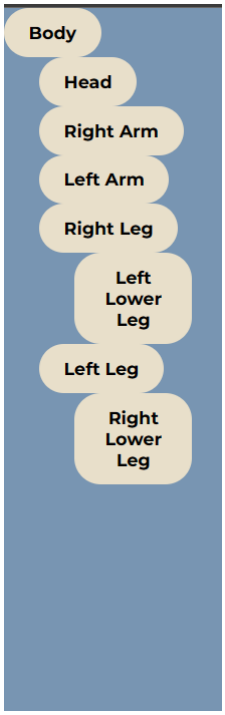
   Kanvas menampilkan subtree yang diklik oleh pengguna dalam *component tree.* Kanvas ini memiliki kontrol kamera yang berbeda dari kanvas model.



Gambar 30. Kanvas Komponen

L. Component Tree

Pada website terdapat sebuah *tree* yang berisikan tombol-tombol merepresentasikan *articulated model* serta *children*-nya. Ketika salah satu tombol diklik, pengguna bisa memanipulasi *subtree* dari objek yang dipilih, serta akan tampil pada kanvas *subtree.*



Gambar 31. Component Tree