

**Laporan UTS Grafkom
Fun Walking Around**



Kelompok 23

Anggota Kelompok:

Nicholas Gunawan - c14200048

Azarya Dami - c14200090

Kenneth Poniman - c14200070

Kelas Grafkom A
Informatika

Penjelasan singkat proyek:

Tema proyek kami adalah fun walking around, dimana diceritakan terdapat seseorang yang berjalan mengelilingi environment baru nan unik dimana terdapat banyak object yang memiliki gerakan unik. User dapat berinteraksi dengan player melalui keyboard dan mouse untuk mendapatkan kombinasi gerakan yang diinginkan. Object yang digunakan terdiri dari environment land, player/ orang dan beberapa obstacles dan objects yang bersifat interaktif maupun statis. Parametric quadric yang digunakan adalah box, ellipsoid, elliptic cone, elliptic paraboloid. Transisi yang digunakan untuk animasi adalah rotation, scaling, dan translation. User juga dapat mengontrol player/ orang untuk berjalan dan juga dapat mengontrol camera serta scaling object.

Cerita yang kami gunakan adalah seseorang berada di dunia fantasi di mana terdapat banyak objek fantasi yang berterbangan dan berputar/ spin. Orang tersebut bisa dikontrol oleh user menggunakan keyboard atau mouse. Objek yang ada memiliki banyak variasi gerakan dan ada juga objek yang bisa berubah skala dalam kontrol user. User dapat bergerak maju mundur atau ke kanan dan ke kiri.

Warna background yang digunakan adalah biru langit/ biru muda dengan parameter RGBA 0, 0.9, 1, dan 0 membentuk warna biru muda.

Berikut adalah bagian individu masing masing anggota:

Bagian individu:

[Nicholas G](#)

[Azarya Dami](#)

[Kenneth Poniman](#)

Nicholas G - c14200048

Bagian saya adalah membuat 1 object orang dari 2 jenis parametric quadric, yaitu box dan ellipsoid serta kurva bezier, environment berupa jalanan atau land, obstacle berupa spinning objects yang terdiri dari 3 objek singular dan 1 object gabungan dari 2 jenis parametric quadric, yaitu box dan elliptic cone. Selain itu, terdapat sebuah object dari ellipsoid yang bisa discale dengan menekan angka 1-9 dan akan berubah skala sesuai angka yang ditekan (1 sampai 9). Untuk jenis parametric quadric yang saya gunakan adalah box, ellipsoid, elliptic cone, dan elliptic paraboloid (4 jenis) serta 3 kurva bezier.

Sedangkan, untuk animasi saya menggunakan scaling, rotating, dan translation. Scaling digunakan menggunakan `Matrix4.CreateScale()` untuk setiap sumbu, yaitu sumbu Y, X, dan sumbu Z. Scaling digunakan pada object ellipsoid yang bisa discale dengan menekan angka 1 sampai 9 di keyboard. Rotation banyak digunakan pada obstacle, baik singular maupun gabungan masing masing pada sumbu Y, X, dan Z dengan time dan degree tertentu yang telah diatur menggunakan `MathHelper.DegreesToRadians()`. Rotating dilakukan dengan mengalikan `Matrix4` dengan `Matrix4.CreateRotation()` sehingga menghasilkan spinning object dengan pola tertentu. Translation adalah memindahkan object dari 1 titik ke titik lain dan pada hal ini dilakukan pada object orang ketika menekan tombol up, left, down, dan right sehingga terlihat seperti sedang bergerak.

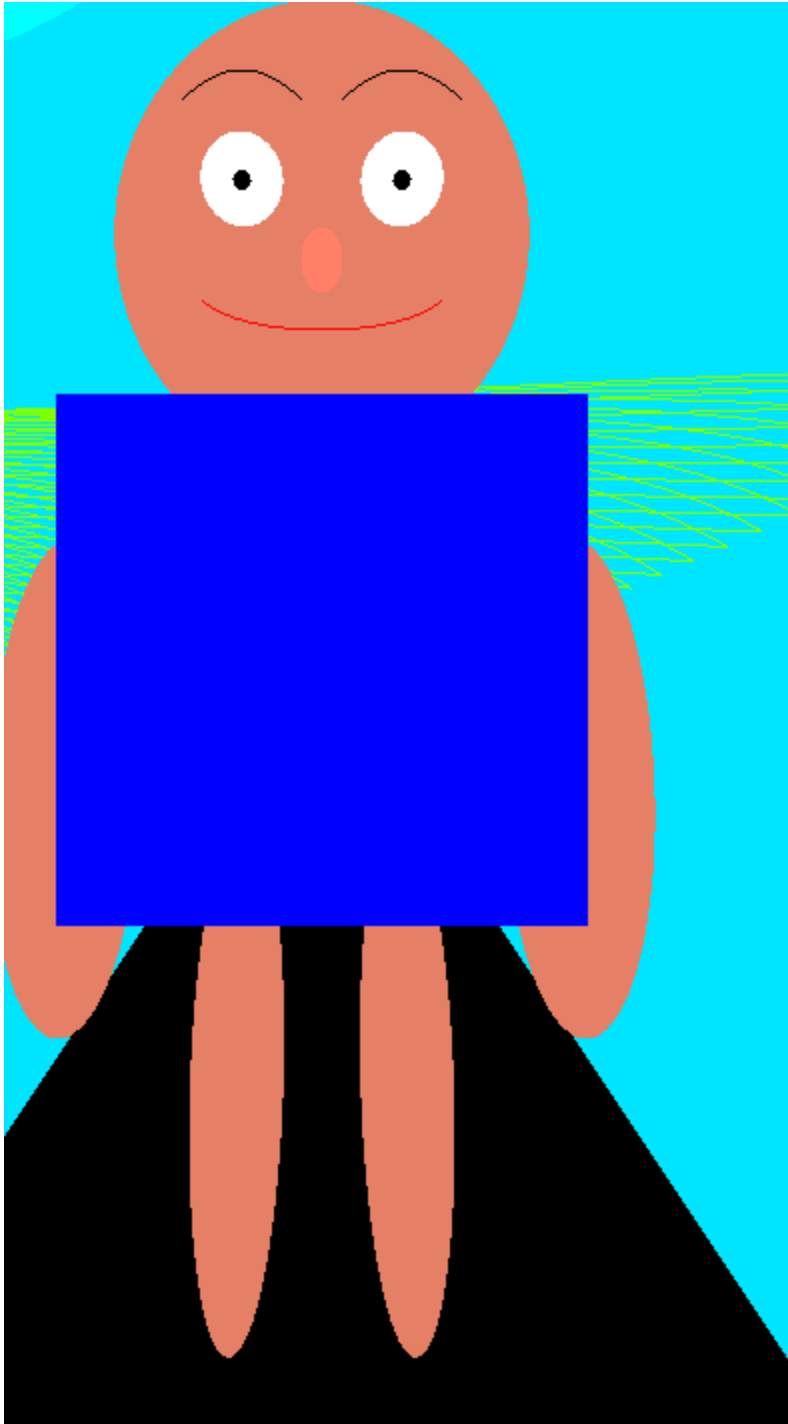
Selain menggunakan animasi dan objects, saya juga mengatur camera sehingga dapat zoom in, zoom out, move camera, dan mengatur angle dengan gerakan mouse. Jika mouse digerakkan di atas mousepad atau permukaan, maka angle camera berubah mengikuti gerakan mouse. Selain itu, gerakan kamera juga dapat diatur menggunakan button A, W, S, dan D yaitu ke kiri, maju, ke kanan, dan mundur. Selain itu, dengan menekan tombol N maka posisi camera dapat diubah.

Untuk pengaturan warna juga dapat dilakukan dengan parameter pada object3d, dimana menggunakan Vector3 yang akan ditranslasikan kepada shader.frag melalui SetVector() sehingga warna dapat diatur sesuka hati menggunakan formula RGB float. Posisi dan ukuran object serta arah elliptic cone (ke kanan atau kiri) dapat diatur pada parameter create() masing masing jenis object.

Tujuan saya menggunakan metode interaktif (penekanan keyboard dan penggunaan mouse) supaya user dapat lebih berinteraksi dengan program dan tidak hanya sekedar menonton gerakan program saja. User dapat mengatur camera, ukuran object, posisi, dan angle camera melalui metode interaktif tersebut.

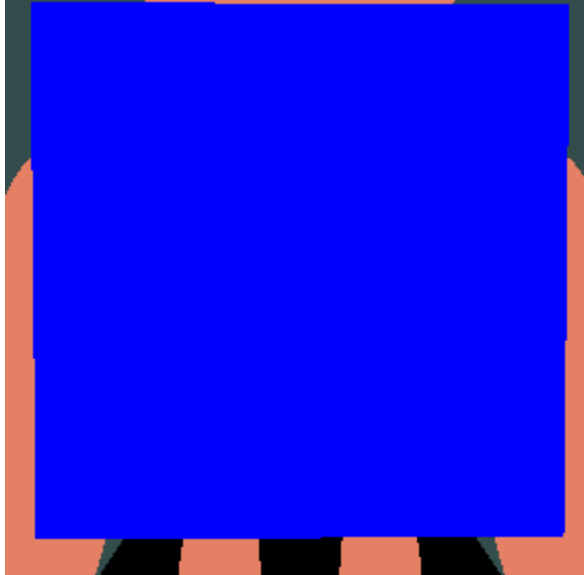
Penjelasan singkat tiap object dan animasi yang saya buat adalah sebagai berikut:

Object:



Object 1: player/ orang, terdiri dari 2 jenis parametric quadric, yaitu box vertices dan ellipsoid serta kurva bezier. Box vertices digunakan pada badan dengan menggunakan ukuran panjang yang berbeda sehingga membentuk balok. Kode yang digunakan untuk membentuk badan player adalah: `object3d[0] = new Asset3D(new Vector3(0, 0, 1.0f));`
`object3d[0].createBoxVertices(0, 0, 0, 0.5f, 0.5f, 0.5f);`

Dimana Asset3D adalah class yang berisi rumus parametric quadric dan method yang digunakan serta method render dan load. Vector3(0,0,1.0f) digunakan untuk parameter color sehingga menghasilkan badan yang mengenakan baju berwarna biru tua. CreateBoxVertices(0, 0, 0, 0.5f, 0.5f, 0.5f) digunakan untuk menghasilkan balok dengan panjang 0.5f di semua sisi sehingga berbentuk kubus serta berada di koordinat X, Y, dan Z pada titik 0 atau tepat di ujung. Hasil yang dihasilkan adalah sebagai berikut:



Ellipsoid digunakan pada kedua kaki, tangan, mata, dan bola mata. Ellipsoid pada tangan kanan menggunakan kode sebagai berikut:

```
object3d[3] = new Asset3D(new Vector3(0.9f, 0.5f, 0.4f));  
object3d[3].createEllipsoid(0.05f, 0.3f, 0.25f, 0.3f, -0.15f, 0);
```

Warna yang digunakan adalah warna kulit/ krem dengan parameter red 0.9f, green 0.5f, dan blue 0.4f. Parameter ellipsoid adalah createEllipsoid(radiusX, radiusY, radiusZ, _x, _y, _z). Radius yang digunakan adalah 0.05f di sumbu X, 0.3f di sumbu Y, dan 0.25f di sumbu Z. sehingga ellipsoid lebih bersifat oval dimana akan lebih condong ke sumbu Y sehingga membentuk tangan, seperti berikut:



Seperti dapat dilihat dari gambar di atas, tangan akan berbentuk oval yang melonjong ke sumbu Y sehingga lebih terlihat seperti tangan kanan.

Sama seperti tangan kanan, tangan kiri juga dibuat dengan konsep serupa dengan kode sebagai berikut:

```
object3d[2] = new Asset3D(new Vector3(0.9f, 0.5f, 0.4f));
object3d[2].createEllipsoid(0.05f, 0.3f, 0.25f, -0.3f, -0.15f, 0);
```

Warna yang digunakan sama seperti tangan kanan, yaitu warna kulit/ krem. Posisi dan ukuran serupa dengan tangan kanan hanya berbeda di posisi x atau parameter `_x` yaitu `-0.3f`, dimana merupakan kebalikan atau negatif dari posisi x tangan kanan yaitu `0.3f` sehingga menghasilkan bentuk yang serupa di posisi berbeda, seperti berikut:



Kaki kanan dan kaki kiri juga dibuat menggunakan ellipsoid. Kaki kanan dibuat menggunakan cara yang sama dengan tangan, yaitu ellipsoid yang diameter y dipanjangkan dan diameter x nya dikecilkan sehingga berbentuk melonjong dengan kode berikut:

```
object3d[4] = new Asset3D(new Vector3(0.9f, 0.5f, 0.4f));
object3d[4].createEllipsoid(0.05f, 0.3f, 0.25f, 0.1f, -0.5f, 0);
```

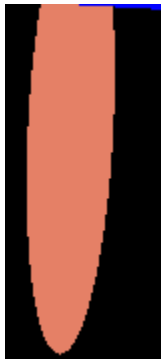
Warna yang digunakan adalah warna krem kulit, sama seperti warna tangan. Diameter yang digunakan adalah 0.05 di sumbu X, 0.3 di sumbu Y, dan 0.25 di sumbu Z. Hasil yang dihasilkan adalah sebagai berikut:



Kaki kiri dibuat dengan formula yang sama dengan kaki kanan, hanya diganti di bagian posisi x, yaitu negatif dari posisi Y, dengan kode sebagai berikut:

```
object3d[5] = new Asset3D(new Vector3(0.9f, 0.5f, 0.4f));  
object3d[5].createEllipsoid(0.05f, 0.3f, 0.25f, -0.1f, -0.5f, 0);
```

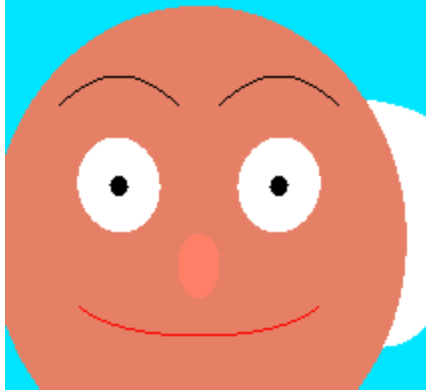
Sumbu X diganti dari 0.1 menjadi -0.1 sehingga berpindah ke sebelah kiri menjadi kaki kiri.



Untuk muka, digunakan ellipsoid dengan sisi yang berukuran sama, dengan program sebagai berikut:

```
object3d[1] = new Asset3D(new Vector3(0.9f, 0.5f, 0.4f));  
object3d[1].createEllipsoid(0.25f, 0.25f, 0.25f, 0, 0.5f, 0);
```

Warna yang digunakan, sama seperti tangan yaitu warna krem kulit. Posisi yang digunakan adalah tepat di sumbu X, 0.5 dari sumbu Y, dan tepat di sumbu Z. Ukuran yang digunakan adalah 0.25, 0.25, dan 0.25 sehingga berbentuk tepat seperti bola, seperti dapat dilihat pada gambar berikut:



Pada muka, terdapat 5 ellipsoid yang digunakan yaitu 2 mata dan 2 bola mata serta 1 untuk hidung. Mata kanan menggunakan ellipsoid dengan kode sebagai berikut:

```
object3d[9] = new Asset3D(new Vector3(1.0f, 1.0f, 1.0f));  
object3d[9].createEllipsoid(0.05f, 0.05f, 0.05f, 0.1f, 0.6f, 0);
```

Warna yang digunakan putih, yaitu menggunakan RGB 1 karena putih merupakan warna dasar RGBA bersama dengan hitam. Diameter yang digunakan adalah 0.05 di semua koordinat sehingga menghasilkan bola yang teratur. Posisi dimasukkan di 0.1, 0.6, dan tepat di sumbu Z. Mata kanan dihasilkan sebagai berikut:



Mata kiri memiliki formula yang serupa, hanya berbeda di posisi x, sebagai berikut:

```
object3d[8] = new Asset3D(new Vector3(1.0f, 1.0f, 1.0f));  
object3d[8].createEllipsoid(0.05f, 0.05f, 0.05f, -0.1f, 0.6f, 0);
```

Warna dan ukuran serta posisi y dan z yang digunakan sama persis dengan mata kanan. Posisi x dirubah, yaitu menggunakan negatif dari posisi x mata kanan, dengan hasil seperti berikut:



Di dalam masing masing mata terdapat bola mata, yang dibuat dengan ukuran yang lebih kecil dan posisi yang berada di dalam mata. Bola mata kanan dibuat dengan kode sebagai berikut:

```
object3d[11] = new Asset3D(new Vector3(0f, 0f, 0f));  
object3d[11].createEllipsoid(0.01f, 0.01f, 0.01f, 0.1f, 0.6f, 0);
```

Warna yang digunakan adalah hitam dengan RGB 0 karena hitam adalah warna dasar. Untuk diameter digunakan 0.01 untuk semua sisi sehingga dihasilkan bola sempurna. Posisi diletakkan di dalam posisi mata, yaitu 0.1, 0.6, dan tepat di sumbu Z. Hasil yang dihasilkan adalah sebagai berikut:



Bola mata kiri juga dibuat dengan formula serupa dengan ukuran dan warna serta posisi y dan z yang sama dengan bola mata kanan. Kode yang digunakan adalah sebagai berikut:

```
object3d[10] = new Asset3D(new Vector3(0f, 0f, 0f));
object3d[10].createEllipsoid(0.01f, 0.01f, 0.01f, -0.1f, 0.6f, 0);
```

Yang dibedakan adalah posisi x, yaitu -0.1 atau negatif dari posisi x bola mata kanan, seperti dilihat pada gambar berikut:



Hidung dibuat dengan formula yang mirip dengan tangan, yaitu sebagai berikut:

```
object3d[18] = new Asset3D(new Vector3(1.0f, 0.5f, 0.4f));
object3d[18].createEllipsoid(0.025f, 0.0375f, 0.025f, 0f, 0.5f, 0);
```

Warna yang digunakan adalah warna krem yang sedikit diterangkan agar tidak menyatu dengan warna kulit, dengan menambahkan nilai red dari 0.5 menjadi 1. Ukuran yang digunakan sedikit memanjang di bagian koordinat Y, dengan panjang 0.025 dari sumbu X, 0.0375 dari sumbu Y, dan 0.025 dari sumbu Z membentuk lonjong. Posisi diletakkan pada sumbu X, 0.5 dari sumbu Y, dan pada sumbu Z sehingga tepat berada di tengah tengah kedua mata, sebagai berikut:



Kurva bezier saya gunakan untuk kedua alis dan mulut, dimana setiap kurva bezier mengandung 2 object (1 kurva standard dan 1 untuk bezier menggunakan _verticesTemp). Untuk bagian mulut, saya menggunakan kurva bezier merah dengan koordinat sebagai berikut:

```
mouth.load(Constants.path + "shader.vert", Constants.path + "shader.frag");
mouth2.load(Constants.path + "shader.vert", Constants.path + "shader.frag");
mouth.createCurve(-0.15f, 0.45f);
mouth.createCurve(-0.1f, 0.4f);
mouth.createCurve(0.1f, 0.4f);
mouth.createCurve(0.15f, 0.45f);
```

Dimana parameter pertama adalah posisi di sumbu X dan parameter kedua adalah posisi di sumbu Y. Kemudian, mulut dirender sebagai berikut:

```
if (mouth.getVertices())
{
    List<float> _verticesTemp = mouth.createCurveBezier();
    mouth2.setVertices( _verticesTemp.ToArray());
    mouth2.load(Constants.path + "shader.vert", Constants.path + "shader.frag");
    mouth2.render( time, temp, 3, camera.GetViewMatrix(),
    camera.GetProjectionMatrix());
}
```

```
mouth.render( time, temp, 2, camera.GetViewMatrix(),
camera.GetProjectionMatrix());
```

Parameter 2 pada mouth digunakan untuk menggambar kurva biasa, sedangkan parameter 3 pada mouth2 digunakan untuk menggambar kurva bezier. Hasil yang dihasilkan adalah sebagai berikut:



Alis kanan dibuat menggunakan kurva bezier dengan warna dasar hitam. Alis kanan diletakkan di atas mata kanan dengan koordinat sebagai berikut:

```
eyebrow2.load(Constants.path + "shader.vert", Constants.path + "shader.frag");
eyebrowBezier2.load(Constants.path + "shader.vert", Constants.path + "shader.frag");
eyebrow2.createCurve(0.025f, 0.7f);
eyebrow2.createCurve(0.075f, 0.75f);
eyebrow2.createCurve(0.125f, 0.75f);
eyebrow2.createCurve(0.175f, 0.7f);
```

Kemudian, alis kanan dirender sebagai berikut:

```
if (eyebrow2.getVertices())
{
```

```
List<float> verticesTemp = eyebrow2.createCurveBezier();
eyebrowBezier2.setVertices( verticesTemp.ToArray());
eyebrowBezier2.load(Constants.path + "shader.vert", Constants.path + "shader.frag");
eyebrowBezier2.render( time, temp, 3, camera.GetViewMatrix(),
camera.GetProjectionMatrix());
```

```
}
```

```
eyebrow2.render( time, temp, 2, camera.GetViewMatrix(),
camera.GetProjectionMatrix());
```

Hasil yang dihasilkan adalah sebagai berikut:



Alis kiri digambarkan dengan formula yang serupa dengan alis kanan, hanya berbeda di bagian posisi X saja, dengan koordinat sebagai berikut:

```
eyebrow.load(Constants.path + "shader.vert", Constants.path + "shader.frag");
eyebrowBezier.load(Constants.path + "shader.vert", Constants.path + "shader.frag");
eyebrow.createCurve(-0.175f, 0.7f);
eyebrow.createCurve(-0.125f, 0.75f);
eyebrow.createCurve(-0.075f, 0.75f);
eyebrow.createCurve(-0.025f, 0.7f);
```

Kemudian, dirender sebagai berikut:

```

if (eyebrow.getVertices())
{

    List<float> _verticesTemp = eyebrow.createCurveBezier();
    eyebrowBezier.setVertices( _verticesTemp.ToArray());
    eyebrowBezier.load(Constants.path + "shader.vert", Constants.path + "shader.frag");
    eyebrowBezier.render( time, temp, 3, camera.GetViewMatrix(),
    camera.GetProjectionMatrix());

}

eyebrow.render( time, temp, 2, camera.GetViewMatrix(),
camera.GetProjectionMatrix());

```

Hasilnya adalah sebagai berikut:



Total jumlah object3D untuk player adalah 17 object, yaitu 1 hidung, 4 untuk alis (2 kurva bezier), 2 untuk mulut (1 kurva bezier), 2 mata, 2 bola mata, 2 kaki, 2 tangan, kepala, dan badan. Selain itu, untuk environment, kami menggunakan object road yang merupakan box vertices dengan sisi Z yang panjang sehingga terlihat seperti tak terbatas. Kode yang digunakan adalah sebagai berikut:

```

_object3d[7] = new Asset3D(new Vector3(0, 0, 0));
_object3d[7].createBoxVertices(0, -2f, 0, 2.0f, 1.0f, 200.0f);

```

Warna yang digunakan adalah hitam dengan parameter 0 di RGB. Posisi digunakan adalah tepat di sumbu X, -2 dari sumbu Y, dan tepat di sumbu Z, sedangkan ukuran yang digunakan adalah 2 untuk sumbu X, 1 untuk sumbu Y, dan 200 untuk sumbu Z, sehingga object akan memanjang ke depan maupun ke belakang sehingga tampak seolah olah seperti jalanan/ road. Hasilnya adalah sebagai berikut:



Terdapat juga object lain yang digunakan, yaitu 1 object gabungan dan 4 object tunggal. Object gabungan merupakan gabungan 2 jenis parametric quadric, yaitu 1 box vertices dan 2 elliptic cone. Box vertices digunakan sebagai penampung dan dibuat dengan kode dibawah ini:

```
object3d[12] = new Asset3D(new Vector3(0.5f, 0.0f, 0.5f));  
object3d[12].createBoxVertices(1.5f, 0.025f, 0.01f, 0.5f, 0.5f, 0.5f);
```

Warna yang digunakan adalah ungu dengan posisi 1.5 dari sumbu X, 0.025 dari sumbu Y, dan 0.01 dari sumbu Z. Ukuran yang digunakan sama, yaitu 0.5 dari semua koordinat sehingga membentuk sebuah kubus.

Selain itu, terdapat 2 elliptic cone yang digunakan sebagai “hammer”, 1 berada di bagian belakang dan 1 berada di bagian depan. Untuk yang berada di bagian depan, animasi yang digunakan adalah:

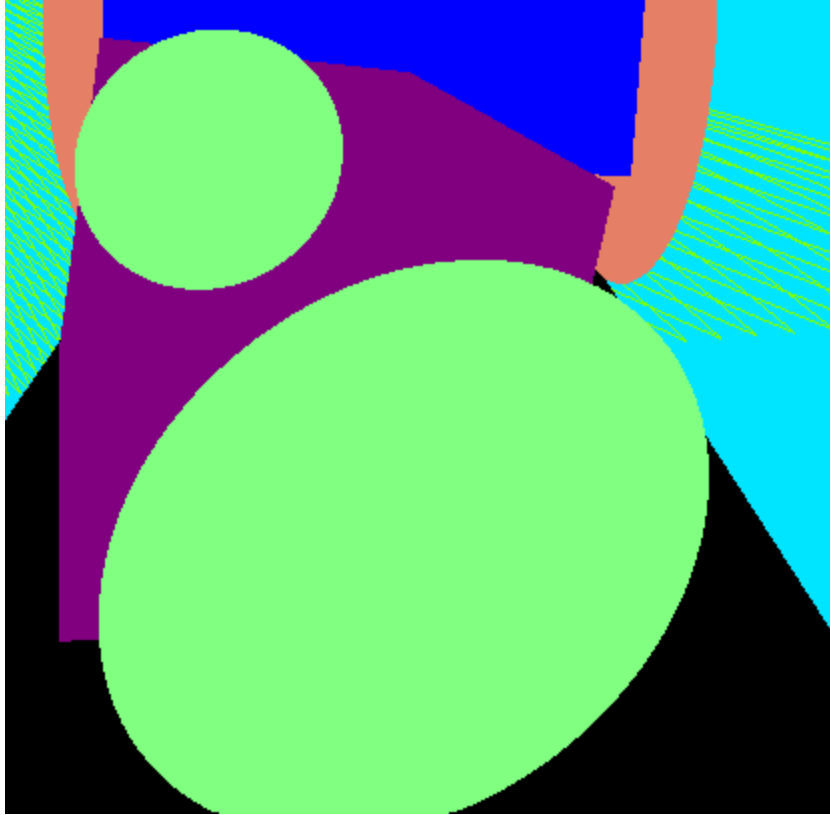
```
object3d[13] = new Asset3D(new Vector3(0.5f, 1.0f, 0.5f));  
object3d[13].createEllipticCone(0.125f, 0.125f, 0.125f, 1.5f, 0.025f, 0.2f, -1);
```

Warna yang digunakan adalah hijau muda dengan posisi 1.5 di sumbu X, 0.025 di sumbu Y, dan 0.2 di sumbu Z dengan arah -1 atau mengarah ke luar. Ukuran yang digunakan sama semua, yaitu 0.125 di semua koordinat.

Untuk hammer bagian belakang, kode yang digunakan mirip, hanya berbeda di bagian direction (parameter paling belakang), seperti berikut:

```
object3d[14] = new Asset3D(new Vector3(0.5f, 1.0f, 0.5f));  
object3d[14].createEllipticCone(0.125f, 0.125f, 0.125f, 1.5f, 0.025f, 0.2f, 1);
```

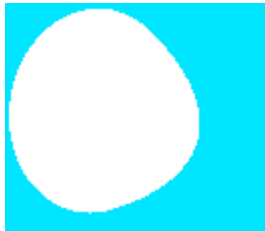
Perbedaannya adalah hammer mengarah ke dalam sehingga keseluruhan obstacle menghasilkan sebuah hammer dengan gagang hammer, sebagai berikut:



Selain object gabungan, terdapat juga beberapa object tunggal, yaitu object yang diberi nama UFO yang merupakan elliptic paraboloid. Kode yang digunakan adalah sebagai berikut:

```
object3d[15] = new Asset3D(new Vector3(1f, 1f, 1f));  
object3d[15].createEllipticParaboloid(0.125f, 0.125f, 0.125f, -2f, 0.15f, -0.25f);
```

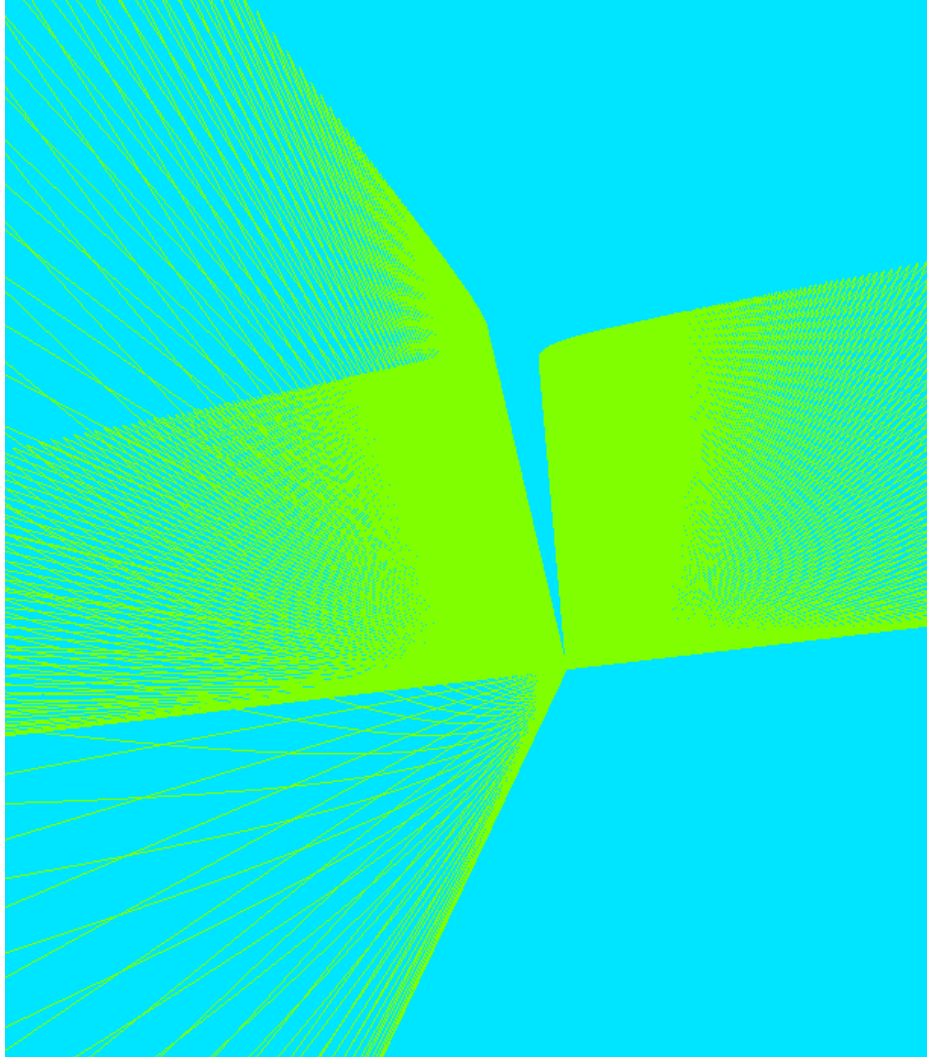
Warna yang digunakan adalah putih dengan parameter 1 pada RGBA. Ukuran yang digunakan sama, yaitu 0.125 pada semua koordinat. Posisi yang digunakan adalah -2 di sumbu X, 0.15 di sumbu Y, dan -0.25 di sumbu Z. Hasil yang dihasilkan adalah sebagai berikut:



Selain itu, terdapat object singular lain yaitu yang diberi nama banner yang merupakan bentuk parametric quadric jenis hyperboloid paraboloid. Kode yang digunakan adalah sebagai berikut:

```
object3d[17] = new Asset3D(new Vector3(0.5f, 1.0f, 0.0f));  
object3d[17].createHyperboloidParaboloid(0.125f, 0.125f, 0.125f, -1.5f, 0.1f, -0.25f);
```

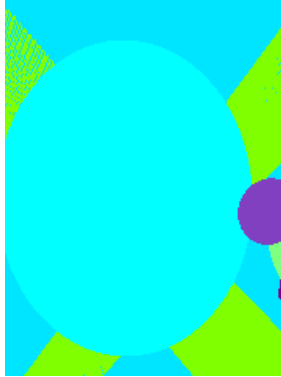
Warna yang digunakan adalah hijau muda dan menggunakan ukuran 0.125 di semua koordinat, serta posisi -1.5 di sumbu X, 0.1 di sumbu Y, dan -0.25 di sumbu Z. Hasil yang dihasilkan adalah sebagai berikut:



Terdapat juga object tunggal lain yaitu bentuk hyperboloid 2 sisi yang memutar mengelilingi sumbu Y. Object ini dibuat dengan menggunakan kode sebagai berikut:

```
object3d[6] = new Asset3D(new Vector3(0f, 1.0f, 1.0f));  
object3d[6].createHyperBolloid2Sheet(0.025f, 0.025f, 0.025f, 1.75f, 0.1f, 0.5f);
```

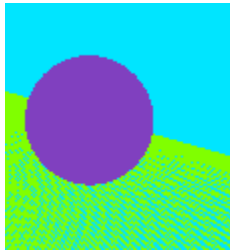
Warna yang digunakan adalah biru muda dengan ukuran yang sama yaitu 0.025 di semua koordinat, sedangkan posisi yang digunakan adalah 1.75 dari sumbu X, 0.1 dari sumbu Y, dan 0.5 dari sumbu Z. Hasil yang dihasilkan adalah sebagai berikut:



Selain itu, terdapat object tunggal yang diberi nama scaling object, karena skala object ini bisa diatur dengan menekan tombol angka 1 sampai 9 di keyboard user. Jika ditekan angka 2, maka skala object akan membesar 2 kali lipat di semua koordinat terhadap semua sisi, begitu seterusnya untuk angka angka lain. Object ini dibuat dengan menggunakan kode:

```
object3d[16] = new Asset3D(new Vector3(0.5f, 0.25f, 0.75f));  
object3d[16].createEllipsoid(0.125f, 0.125f, 0.125f, 1.5f, 0.1f, -0.25f);
```

Warna yang digunakan adalah warna ungu dengan ukuran yang sama yaitu 0.125 di semua koordinat pada skala awal sehingga berbentuk bola sempurna. Object ditempatkan di posisi 1.5 dari sumbu X, 0.1 dari sumbu Y, dan -0.25 dari sumbu Z dan berada di sekitar daerah banner object. Hasil yang dihasilkan adalah sebagai berikut:



Contoh gambar ketika user menekan angka 4:



Dapat dilihat, object berpindah menjadi lebih maju ke depan. Hal ini disebabkan karena object membesar ke semua sisi dengan skala yang sama, yaitu 4.0. Object juga mengecil ke semua sisi jika skala diperkecil, sehingga object terlihat menjauh dari kamera pusat. Untuk bentuk yang digunakan dan posisi object, dapat dilihat pada tabel berikut:

Bentuk parametric quadric/ kurva	Nomor index object yang menggunakan	Nama object yang menggunakan
Box vertices	0, 7, 12	Badan player, road, kotak hammer (gagang)
Elipsoid	1, 2, 3, 4, 5, 8, 9, 10, 11, 16, 18	Muka, mata, bola mata, hidung, kaki, tangan, scaling object
Elliptic cone	13, 14	Hammer luar dan hammer dalam
Hyperboloid 2 sisi	6	Object hyperboloid
Elliptic paraboloid	15	UFO
Hyperboloid paraboloid	17	Banner
Kurva bezier	- (6 object2d tunggal)	Mulut, alis kanan, dan alis kiri

Sehingga total bentuk parametric yang digunakan adalah 6 jenis dengan 1 kurva bezier dan total object3d yang digunakan adalah 19 object dengan 6 object2d untuk kurva bezier dengan total keseluruhan object 25 buah.

Animasi dan transformasi:

Animasi di object object saya menggunakan Matrix4 yang diubah ubah sesuai kebutuhan (scaling, rotation, dan translation) dengan time yang berbeda beda pula. Saya menggunakan 4 variabel untuk time dan 8 variabel Matrix4 untuk menyimpan transisi object, karena terdapat 8 jenis kombinasi gerakan berbeda, sebagai berikut:

1. Temp digunakan untuk gerakan player, diatur pada bagian OnUpdateFrame(), dimana ketika user menekan tombol up, down, left, atau right maka temp akan ditranslasikan dengan contoh sebagai berikut (penekanan tombol up): `temp = temp * Matrix4.CreateTranslation(0.0f, 0f, 0.05f);`, sehingga ketika user menekan tombol up, maka posisi player beserta dengan object object tubuh akan bergerak maju ke depan dari sumbu Z sebanyak 0.05, begitu juga untuk penekanan tombol lain, maka player akan bergerak ke kanan, kiri, atau belakang sebanyak 0.05. Temp digunakan oleh semua object tubuh dari player kecuali kedua kaki karena adanya transformasi tambahan, yaitu kedua tangan, muka, kedua mata dan bola mata, kedua alis, badan, dan mulut.
2. Temp2 digunakan untuk object yang stabil atau tidak bergerak, seperti road atau jalan yang digunakan. Temp2 tidak diberikan transformasi apapun sehingga tidak terpengaruh oleh penekanan tombol ataupun time dan biasa digunakan dengan time 0 sehingga tidak melakukan transformasi. Temp2 hanya digunakan pada road.

- Temp3 merupakan rotasi kombinasi melalui 3 sumbu secara bergantian, yaitu dari sumbu Y, dilanjutkan ke sumbu Z, dan ke sumbu X secara berulang dengan sudut putar 1, sebagai berikut:

```
temp3 = Matrix4.Identity;
degr += MathHelper.DegreesToRadians(1f);
temp3 = temp3 * Matrix4.CreateRotationY(degr);
temp3 = temp3 * Matrix4.CreateRotationZ(degr);
temp3 = temp3 * Matrix4.CreateRotationX(degr);
```

Degr merupakan sudut yang digunakan dan menggunakan MathHelper untuk konversi sudut ke radians sebanyak 1. Kemudian melakukan rotasi melalui 3 sumbu seperti disebutkan di atas. Temp3 digunakan pada object UFO dan object gabungan hammer dengan cara dimasukkan ke parameter ketiga object tunggal penyusun hammer, yaitu 2 elliptic cone dan 1 box vertices.

- Temp4 merupakan scaling yang didasarkan pada penekanan tombol angka 1 hingga 9 di keyboard user. Skala akan diperbesar atau diperkecil sesuai angka yang ditekan user, misalnya jika user menekan angka 9, maka object akan memperbesar skala X, Y, dan Z menjadi 9 kali lipat ukuran normal (membesar ke semua arah). Sebagai contoh, jika user menekan tombol 8, maka kode berikut akan dijalankan:

```
if (KeyboardState.IsKeyReleased(Keys.D8))
{
    time4 += 18f;
    temp4 = Matrix4.Identity;
    temp4 = temp4 * Matrix4.CreateScale(8.0f);
}
```

Jika tombol 8 ditekan, maka time akan bertambah 18 dan temp4 akan diskala ke angka 8, begitu juga dengan penekanan angka lain. Temp4 merubah skala sehingga object terlihat lebih dekat dengan kamera, karena membesar ke semua arah koordinat. Temp4 digunakan di scaling object untuk mengubak skala object tersebut.

- Temp5 digunakan untuk pergerakan kaki kiri ketika player berjalan ke depan atau ke belakang. Temp6 digunakan untuk pergerakan kaki kanan dengan formula yang serupa dengan temp5. Pergerakan kedua kaki ini diatur dengan function walkingBackwards() dan walking() menggunakan variabel turn sebagai counter, sebagai berikut:

```
public async void walking()
{
    turn ++;
    if(turn % 2 == 0){
        degr2 = MathHelper.DegreesToRadians(0.5f);
        Matrix4 temporary = temp5;
        Matrix4 temporary2 = temp6;
        temp5 = temp5 * Matrix4.CreateRotationX(degr);
```

```

        await Task.Delay(100);
        temp5 = temporary;
        temp6 = temp6 * Matrix4.CreateRotationX(-degr);
        await Task.Delay(100);

```

```

        temp6 = temporary2;
    }
    else
    {
        degr2 = MathHelper.DegreesToRadians(0.5f);
        Matrix4 temporary = temp5;
        Matrix4 temporary2 = temp6;
        temp5 = temp5 * Matrix4.CreateRotationX(-degr);
        await Task.Delay(100);
        temp5 = temporary;
        temp6 = temp6 * Matrix4.CreateRotationX(degr);
        await Task.Delay(100);

```

```

        temp6 = temporary2;
    }

```

```

    }
    public async void walkingBackwards()
    {
        turn2++;
        if (turn2 % 2 == 0)
        {
            degr2 = MathHelper.DegreesToRadians(0.05f);
            Matrix4 temporary = temp5;
            Matrix4 temporary2 = temp6;
            temp5 = temp5 * Matrix4.CreateRotationX(degr);
            await Task.Delay(100);
            temp5 = temporary;
            temp6 = temp6 * Matrix4.CreateRotationX(-degr);
            await Task.Delay(100);

            temp6 = temporary2;
        }
    }

```

```

else
{
    degr2 = MathHelper.DegreesToRadians(0.05f);
    Matrix4 temporary = temp5;
    Matrix4 temporary2 = temp6;
    temp5 = temp5 * Matrix4.CreateRotationX(-degr);
    await Task.Delay(100);
    temp5 = temporary;
    temp6 = temp6 * Matrix4.CreateRotationX(degr);
    await Task.Delay(100);

    temp6 = temporary2;
}
}

```

Dimana akan mengatur pergerakan kedua kaki saat berjalan dengan menggunakan setengah putaran rotasi. Kemudian pergerakan temp5 dan temp 6 akan diatur dengan menggunakan turn, dimana pergerakan akan berbeda setiap turn genap atau ganjil (jika kaki kanan ke depan, maka kaki kiri ke belakang, dan sebaliknya). Degree dan delay diberikan untuk setiap turn agar menghasilkan gerakan seperti berjalan. Delay digunakan agar perubahan posisi rotasi dapat terlihat oleh user. Selain itu, di setiap penekanan tombol up, down, left, atau right, temp5 dan temp6 akan ikut ditranslasikan, sama seperti temp sehingga pergerakan temp5 dan temp6 akan sejajar dengan pergerakan object tubuh lain. Karena adanya delay di setiap turn, maka pergerakan maju/ mundur yang beruntun akan merusak gerakan rotasi kaki, sehingga kekurangannya adalah pergerakan tidak dapat dilakukan terus menerus.

6. Temp7 merupakan pergerakan rotasi memutar sumbu X, dengan kode sebagai berikut:

```

temp7 = Matrix4.Identity;
temp7 = temp7 * Matrix4.CreateRotationX(degr);

```

Sehingga object akan memutar sumbu X terus menerus dengan kecepatan stabil, karena temp7 direset setiap eksekusi kode sehingga speed tidak terus bertambah dan cenderung stabil. Temp7 digunakan pada object banner.

7. Temp8 adalah Matrix4 untuk transformasi yang terakhir dari bagian saya yang merupakan rotasi melalui sumbu Y dan digunakan dengan kode berikut:

```

temp8 = Matrix4.Identity;
temp8 = temp8 * Matrix4.CreateRotationY(degr);

```

Sama seperti temp7, temp8 akan direset setiap eksekusi kode dan memiliki speed yang stabil. Temp8 digunakan pada object yang berbentuk hyperboloid 2 sisi. Didapatkan bahwa daftar transformasi pada bagian saya adalah:

Transformasi	Matrix4 temp yang menggunakan	Object yang menggunakan
Rotasi	Temp3, temp5, temp6, temp7, temp8	Kaki kanan, kaki kiri, UFO, object gabungan hammer, hyperboloid 2 sisi, banner
Translasi	Temp, temp5, temp6	Semua object pembentuk player (kaki, tangan, mata, bola mata, mulut, alis, badan, wajah)
Scaling	Temp4	Scaling object
None	Temp2	Road

Selain itu, terdapat 4 variabel time yang digunakan untuk mengatur timing dan speed dari tiap transformasi/ animasi, yaitu:

```
time = 0;
time2 = 18f;
time3 += 20f;
time4 += 18f;
```

Time tersebut diubah di masing masing method dan menggunakan time yang berbeda untuk tiap kebutuhan transformasi, dapat dilihat pada tabel berikut:

Variabel time	Nilai standard (kondisi diam)	Object yang menggunakan
_time	0	Semua object tubuh player
_time2	18	Road dan object hyperboloid
_time3	20	Banner dan object hammer
_time4	18	Scaling object

Sehingga dapat disimpulkan, bahwa 3 jenis transformasi dasar (scaling, rotation, translation) sudah digunakan di semua bagian saya dengan menggunakan 8 variabel Matrix4 temporary yang digunakan bersamaan dengan 4 variabel time untuk menghasilkan animasi yang diinginkan.

Azarya Dami - c14200090

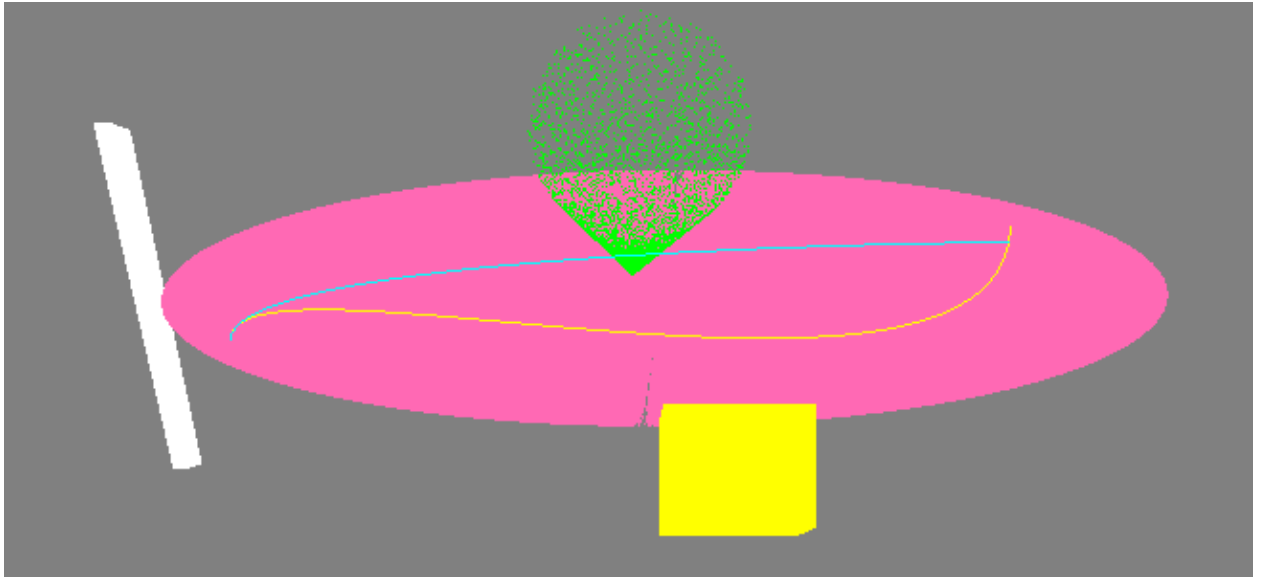
Saya membuat sebuah objek 3D berupa balon udara dengan bentuk seperti sebuah balon udara yang digerakkan menggunakan tenaga mesin untuk menggerakkan baling-baling di belakang dan di atas balon tersebut. Terdapat juga sebuah kabin pilot dan pemancar sinyal dari balon tersebut, sebagai media komunikasi. Terdapat 3 jenis objek quadric yang saya gunakan untuk membentuk balon ini yaitu ellipsoid, elliptic cone, dan elliptic paraboloid. Sedangkan saya juga menggunakan 1 jenis objek 3D yaitu box vertices (segi empat) untuk memberi bentuk yang lebih baik. Dan yang terakhir saya menggunakan 2 lengkungan kurva untuk memberi dekorasi logo pada balon udara tersebut.

Untuk animasi dari object 3D saya menggunakan translasi, rotasi dan scaling saya menggunakan translasi untuk memindahkan objek 3D balon udara ke depan dan belakang, rotasi digunakan dalam perputaran baling-baling balon, dan scalling untuk memperbesar body dari balon tersebut. Disini saya juga menggambar kurva pada body balon untuk menambah estetika balon seperti logo biasanya pada balon.

Pada proses rotasi, objek baling-baling yang menggunakan persamaan quadric elliptic paraboloid (baling-baling atas) dan box vertices (baling-baling belakang). Ketika render pertama kali kedua baling-baling tersebut berada di belakang dan di atas balon dan berputar atau berotasi baik terhadap sumbu Y maupun sumbu X. kecepatan rotasi juga diatur dengan variabel waktu yang di *increment* dengan perkalian $1500f * \text{args.Time}$. Dengan kecepatan 1500f kali ini diharapkan menampilkan pergerakan rotasi baling-baling yang sesuai.

Pada proses transformasi translasi terdapat *event* perpindahan balon terbang untuk maju dan mundur. Proses ini membutuhkan inputan *user* yaitu menekan tombol F1 untuk bergerak ke belakang dan F2 untuk bergerak maju.

Proses scale terjadi pada badan balon yang berbentuk seperti ellips yang akan mengalami scale dan semakin membesar dan posisi nya akan lebih maju ke arah camera sebesar 1.5f dari ukuran awalnya dengan menekan tombol 5 pada *keyboard*.



```
//SUTAN
//kincir1
_object3d[22] = new Asset3D(new Vector3(0.1f, 1.0f, 0.0f));
_object3d[22].createEllipticParaboloid(0.2f, 0.1f, 0.1f, 0.0f, 6.0f, 0.0f);

//kincir2
_object3d[23] = new Asset3D(new Vector3(0.1f, 1.0f, 0.0f));
_object3d[23].createEllipticParaboloid(0.2f, 0.1f, 0.1f, 0.0f, 6.0f, 0.0f);

//kincir3
_object3d[24] = new Asset3D(new Vector3(0.0f, 1.0f, 0.0f));
_object3d[24].createEllipticParaboloid(0.2f, 0.1f, 0.1f, 0.0f, 6.0f, 0.0f);

//baling" belakang
_object3d[25] = new Asset3D(new Vector3(0.0f, 1.0f, 0.0f));
_object3d[25].createBoxVertices(-3.0f, 5.0f, 0.0f, 0.1f, 2.0f, 0.2f);

//balon udara
_object3d[26] = new Asset3D(new Vector3(1.0f, 0.1f, 0.0f));
_object3d[26].createEllipsoid(3.5f, 0.8f, 0.9f, 0.3f, 5.2f, -0.5f);

//kabin pilot
_object3d[27] = new Asset3D(new Vector3(1.0f, 1.0f, 0.0f));
_object3d[27].createBoxVertices(0.8f, 4.0f, -0.3f, 1.0f, 0.8f, 0.5f);

//pemancar jaringan
_object3d[28] = new Asset3D(new Vector3(0.0f, 1.0f, 0.0f));
_object3d[28].createEllipticCone(0.5f, 0.5f, 1.5f, 0.1f, 5.2f, 0.0f, 1);
```

Berikut objek-objek yang saya buat untuk menjadi sebuah objek 3D balon udara:

- a. **Ellipsoid**: Ellipsoid digunakan untuk memberi bentuk pada bagian badan balon udara.

```
//badan pesawat
_object3d[26] = new Asset3D(new Vector3(1.0f, 0.1f, 0.0f));
_object3d[26].createEllipsoid(3.5f, 0.8f, 0.9f, 0.3f, 5.2f, -0.5f);
```

Pada *array of object* Objek 3D yang ke-26 saya membuat badan dari balon berbentuk ellips dengan letak 0.3f sumbu X, 5.2f sumbu Y, dan -0.5f sumbu Z. Hal ini menunjukkan posisi balon yang berada lebih di atas atau sedang terbang.

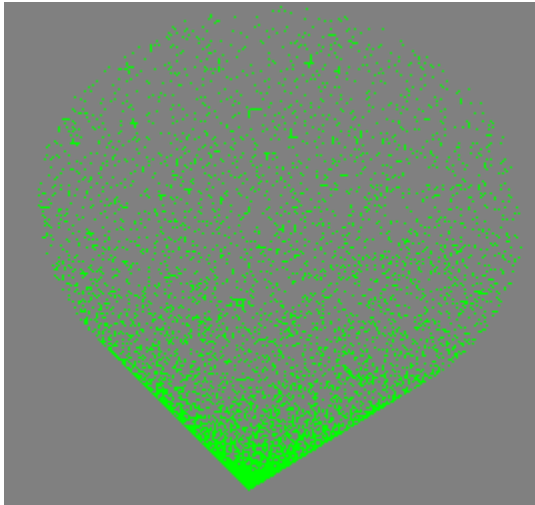


- b. **Elliptic Cone**: Digunakan untuk memberi bentuk seperti halnya pemancar sinyal di daratan namun pembentuk objeknya bukan menggunakan benda padat atau triangleFan sehingga terlihat seperti jaring dan tidak padat yang bertujuan agar dapat menyesuaikan diri ketika balon udara terbang di kondisi cuaca hujan atau angin kencang. Ketika hujan air tidak tertampung pada pemancar sinyal tersebut dan ketika kondisi angin kencang akan mengalir lebih baik arah anginnya karena tidak terhambat pemancar sinyal ini.

```
//pemancar jaringan
_object3d[28] = new Asset3D(new Vector3(0.0f, 1.0f, 0.0f));
_object3d[28].createElipticCone(0.5f, 0.5f, 1.5f, 0.1f, 5.2f, 0.0f, 1);
```

Pada rumus createElipticCone() ini saya memodifikasi dengan menambah 1 parameter sehingga total memiliki 7 parameter yaitu 3 parameter awal untuk menentukan ukuran dari objek elliptic cone tersebut, 3 parameter akhir untuk menentukan letak objek pada sumbu X,Y, dan Z, serta 1 parameter terakhir yang bisa bernilai +1 atau -1 untuk menentukan arah hadapan dari elliptic cone tersebut ke kiri atau ke kanan dengan mengalikan/multiply pada vektor Z.

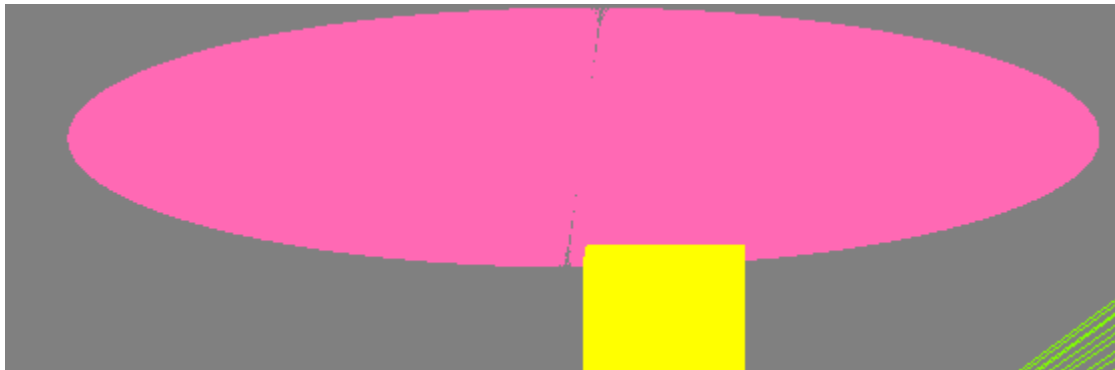
```
5 references
public void createElipticCone(float radiusX, float radiusY, float radiusZ, float _x, float _y, float _z, float dir)
{
    _centerPosition.X = _x;
    _centerPosition.Y = _y;
    _centerPosition.Z = _z;
    float pi = (float)Math.PI;
    Vector3 temp_vector;
    for (float u = -pi; u <= pi; u += pi / 4500)
    {
        for (float v = 0; v <= 1; v += pi / 4500)
        {
            temp_vector.X = _x + (float)Math.Cos(u) * radiusX * v;
            temp_vector.Y = _y + (float)Math.Sin(u) * radiusY * v;
            temp_vector.Z = dir * (_z + v * radiusZ);
            vertices.Add(temp_vector);
        }
    }
}
```



- c. **Box Vertices (Segi Empat):** digunakan untuk membuat kabin pilot dan *passenger* dari balon udara tersebut. Kemudian untuk bagian baling-baling balon yang digunakan menggerakkan balon.

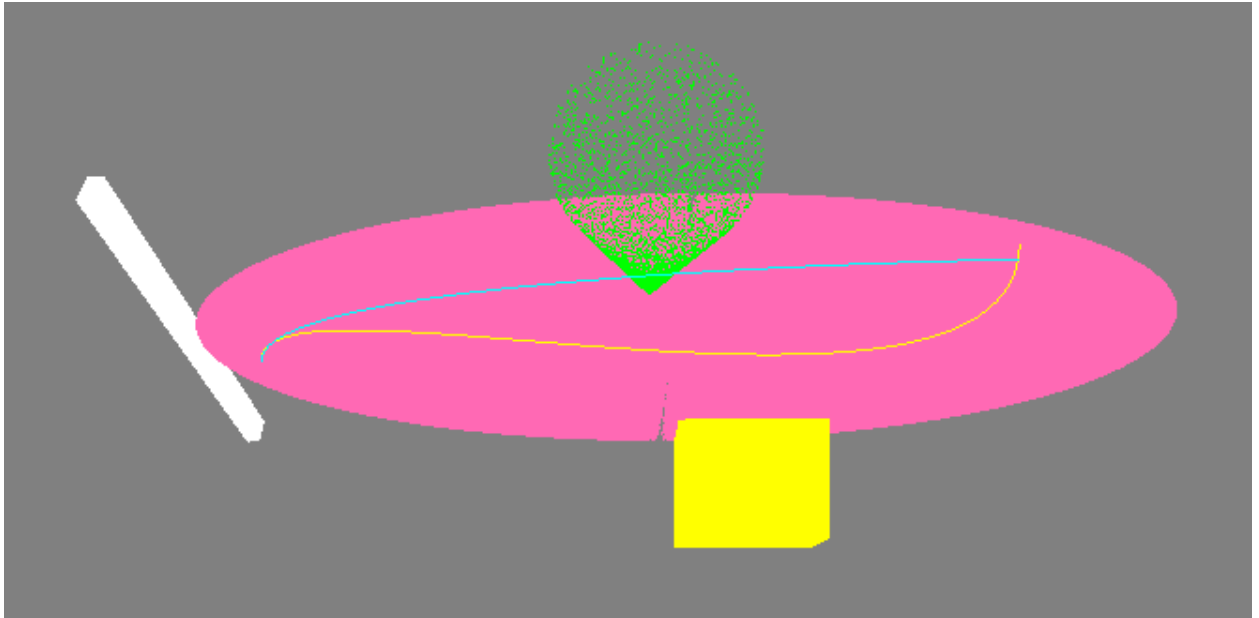
```
//kabin pilot
_object3d[27] = new Asset3D(new Vector3(1.0f, 1.0f, 0.0f));
_object3d[27].createBoxVertices(0.8f, 4.0f, -0.3f, 1.0f, 0.8f, 0.5f);
```

Posisi kabin pilot di set tepat di bawah balon udara dengan ukuran yang sesuai dengan ukuran balon udara.




```
//baling" belakang
_object3d[25] = new Asset3D(new Vector3(1.0f, 1.0f, 1.0f));
_object3d[25].createBoxVertices(-3.0f, 5.0f, 0.0f, 0.1f, 2.0f, 0.2f);
```

Baling-baling belakang juga dibuat dan ketika di render dirotasikan terhadap sumbu X.



d. Kurva Beazier.

Saya menggunakan konsep kurva beazier untuk membuat lekukan garis/les pada balon udara.

```
//SUTAN
moon.load(Constants.path + "shader.vert", Constants.path + "shader.frag");
moonBezier.load(Constants.path + "shader.vert", Constants.path + "shader.frag");

moon.createCurve(-2.5f, 4.8f);
moon.createCurve(-2.5f, 5.5f);
moon.createCurve(2.5f, 4.0f);
moon.createCurve(2.5f, 5.5f);

les.load(Constants.path + "shader.vert", Constants.path + "shader.frag");
lesBezier.load(Constants.path + "shader.vert", Constants.path + "shader.frag");
les.createCurve(-2.5f, 4.8f);
les.createCurve(-2.5f, 5.4f);
les.createCurve(2.5f, 5.4f);
les.createCurve(2.5f, 5.4f);
```

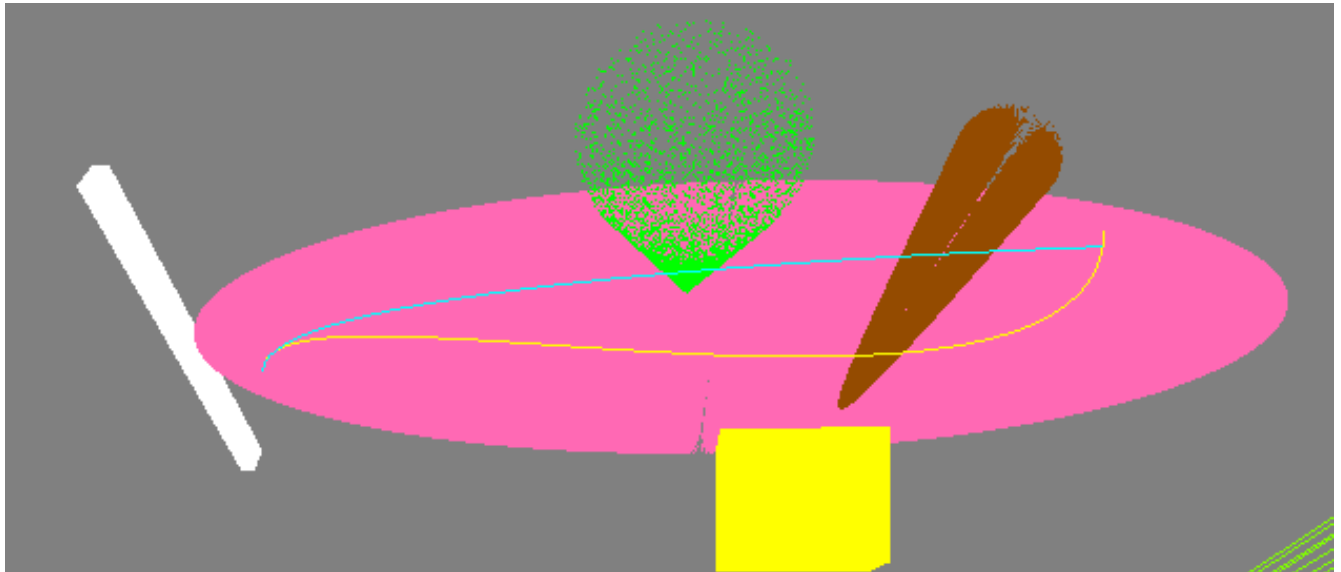
e. Elliptic Paraboloid

Membuat objek 3D berbentuk teropong/scope yang digunakan oleh *passenger* yang ingin menikmati kondisi di luar ruangan *passenger* yang sempit. Ruang ini

hanya bisa dikunjungi ketika balon udara telah berada dalam kondisi stabil di udara.

```
//scope
_object3d[31] = new Asset3D(new Vector3(0.58f, 0.29411f, 0.0f));
_object3d[31].createEllipticParaboloid(0.1f, 0.1f, -0.1f, 1.0f, 4.5f, 0.0f);
```

Berikut hasil akhir dari objek balon udara yang saya buat.

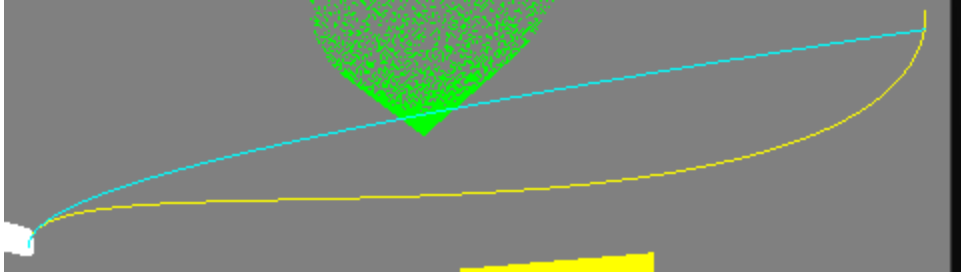


Berikut adalah code inisialisasi posisi dari kurva yang saya gambarkan baik kurva satu yang disebut moon dan kurva dua yaitu les.

```
if (moon.getVertices())
{
    List<float> _verticesTemp = moon.createCurveBezier();
    moonBezier.setVertices(_verticesTemp.ToArray());
    moonBezier.load(Constants.path + "shader.vert", Constants.path + "shader.frag");
    moonBezier.render(_time, temp14, 3, _camera.GetViewMatrix(), _camera.GetProjectionMatrix());
}
moon.render(_time, temp14, 3, _camera.GetViewMatrix(), _camera.GetProjectionMatrix());

if (les.getVertices())
{
    List<float> _verticesTemp = les.createCurveBezier();
    lesBezier.setVertices(_verticesTemp.ToArray());
    lesBezier.load(Constants.path + "shader.vert", Constants.path + "shader.frag");
    lesBezier.render(_time, temp14, 3, _camera.GetViewMatrix(), _camera.GetProjectionMatrix());
}
les.render(_time, temp14, 3, _camera.GetViewMatrix(), _camera.GetProjectionMatrix());
```

Di dalam method onRenderFrame() saya menggambar 2 kurva tersebut.



Berikut adalah penjelasan animasi (transformasi) dari setiap objek yang saya buat yaitu rotasi, translasi dan scaling.

a. Rotasi.

```
//Rotasi kincir belakang
temp9 = temp9 * Matrix4.CreateTranslation(new Vector3(0.0f, -5.0f, 0.0f));
temp9 = temp9 * Matrix4.CreateRotationX(MathHelper.DegreesToRadians((float)_time10));
temp9 = temp9 * Matrix4.CreateTranslation(new Vector3(0.0f, 5.0f, 0.0f));

_object3d[25].render(3, _time, temp9, _camera.GetViewMatrix(), _camera.GetProjectionMatrix());
```

Rotasi kincir belakang saya lakukan dengan menggunakan fungsi `CreateRotationX()`. Di dalamnya saya memberi parameter `_time10` yaitu *increment* waktu ketika baling-baling tersebut mulai berotasi yaitu `_time10 += 1500 * args.Time`.

```
_time10 += 1500f * args.Time;
```

b. Translasi.

Digunakan ketika saya menekan inputan *keyboard* F1 untuk mundur dan F2 untuk maju. Sehingga code nya sebagai berikut:

```

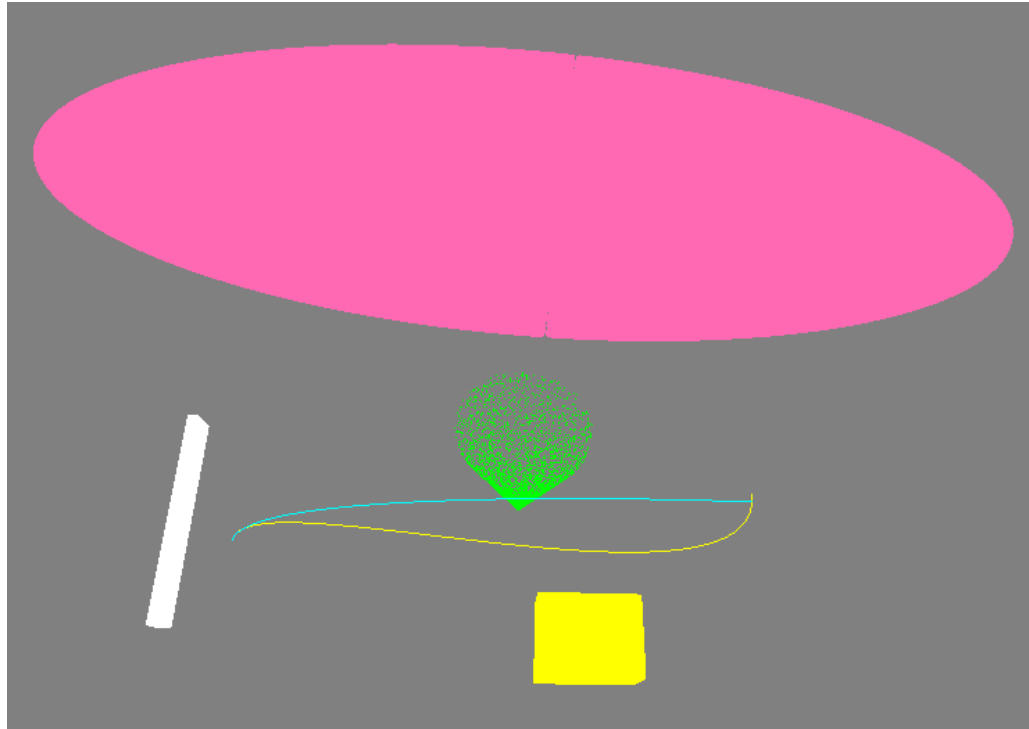
//SUTAN
if (KeyboardState.IsKeyDown(Keys.F2))
{
    _time += 20f * args.Time;
    _time10 += 1000f * args.Time;
    temp9 = temp9 * Matrix4.CreateTranslation(new Vector3(0.0f, -5.0f, 0.0f));
    temp9 = temp9 * Matrix4.CreateRotationX(MathHelper.DegreesToRadians((float)_time));
    temp9 = temp9 * Matrix4.CreateTranslation(0.05f, 0f, 0.0f);
    temp9 = temp9 * Matrix4.CreateTranslation(new Vector3(0.0f, 5.0f, 0.0f));
    temp14 = temp14 * Matrix4.CreateTranslation(0.05f, 0, 0.0f);
    temp15 = temp15 * Matrix4.CreateTranslation(0.05f, 0, 0.0f);
    temp13 = temp13 * Matrix4.CreateTranslation(new Vector3(0.05f, 0, 0.0f));
}

if (KeyboardState.IsKeyDown(Keys.F1))
{
    _time += 20f * args.Time;
    _time10 += 10000f * args.Time;
    temp12 = temp12 * Matrix4.CreateTranslation(-0.05f, 0f, 0.0f);
    temp14 = temp14 * Matrix4.CreateTranslation(-0.05f, 0f, 0.0f);
    temp15 = temp15 * Matrix4.CreateTranslation(-0.05f, 0f, 0.0f);
    temp9 = temp9 * Matrix4.CreateTranslation(new Vector3(0.0f, -5.0f, 0.0f));
    temp9 = temp9 * Matrix4.CreateRotationX(MathHelper.DegreesToRadians((float)_time));
    temp9 = temp9 * Matrix4.CreateTranslation(-0.05f, 0f, 0.0f);
    temp9 = temp9 * Matrix4.CreateTranslation(new Vector3(0.0f, 5.0f, 0.0f));
    temp13 = temp13 * Matrix4.CreateTranslation(new Vector3(-0.05f, 0, 0.0f));
}

```

Seperti yang kita lihat bahwa untuk menggerakkan translasi ke sumbu X sesuai keinginan maka temp9 yang digunakan untuk merender baling-baling perlu saya translasikan terlebih dahulu kemudian dirotasi, setelah itu di translasikan sebanyak 0.05f pada sumbu X untuk bergerak ke kanan dan -0.05f pada sumbu X untuk bergerak ke kiri. Dan terakhir temp9 nya saya translasikan lagi kembali ke posisi awal. Hal ini membuat perputaran rotasi dan translasi dari baling-baling dan objek balon udara lainnya lebih *smooth*.

c. Scale.



Digunakan ketika *user* ingin memberikan atau melihat perpisahan antar objek dari balon udara dengan lebih *detail* sehingga balon udaranya di scale sebesar 1.5f ke semua sumbu ketika *user* menekan inputan *keyboard* 5.

```
if (KeyboardState.IsKeyReleased(Keys.D5))
{
    _time4 += 18f;
    temp4 = Matrix4.Identity;
    temp4 = temp4 * Matrix4.CreateScale(5.0f);

    temp15 = Matrix4.Identity;
    temp15 = temp15 * Matrix4.CreateScale(2.0f);
}
```

Temp15 merupakan temp variabel yang digunakan sementara memperlihatkan efek scale pada objek balon udara.

Saya membuat objek 3D berupa roket. Disini saya menggunakan kurva hanya untuk dekorasi dan untuk memperlihatkan rotasi terhadap sumbu X. Di objek ini juga saya menggunakan parametric quadric yaitu ellipsoid(untuk membentuk roket dan jendela roket) dan ellipticCone(untuk sirip dari roket). Saya juga menggunakan boxVertice untuk bagian bawah roket. Untuk animasi saya menggunakan rotasi untuk menggerakkan kurva dan translasi untuk menaikkan atau menurunkan roket keatas dan kebawah.

Pada proses tranlasi saya menggunakan tombol page down untuk menurunkan roket dan page up menaikkan roket.

Ini adalah kode bentuk dari badan roket

```
//tengah roket
_object3d[19] = new Asset3D(new Vector3(1.0f, 0.0f, 0.0f));
_object3d[19].createEllipsoid(0.5f, 0.9f, 0.5f, 5.0f, 1.0f, 0.0f);
```

Kode untuk bawah roket

```
//tengah roket
_object3d[19] = new Asset3D(new Vector3(1.0f, 0.0f, 0.0f));
_object3d[19].createEllipsoid(0.5f, 0.9f, 0.5f, 5.0f, 1.0f, 0.0f);
```

Kode untuk jendela roket

```
_object3d[21] = new Asset3D(new Vector3(1.0f, 1.0f, 1.0f));
_object3d[21].createEllipsoid(0.2f, 0.2f, 0.2f, 5.0f, 1.5f, 0.0f);
```

Kode untuk sirip

```
//sirip depan
degr1 = MathHelper.DegreesToRadians(180f);
_object3d[29] = new Asset3D(new Vector3(1.0f, 1.0f, 1.0f));
_object3d[29].createEllipticCone(0.125f, 0.125f, 0.25f, 5.0f, 0.3f, -0.7f, 1);

//sirip belakang
_object3d[30] = new Asset3D(new Vector3(1.0f, 1.0f, 1.0f));
_object3d[30].createEllipticCone(0.125f, 0.125f, 0.25f, 5.0f, -0.3f, -0.7f, 1);
```

Dan untuk bentuk dari kurva

```
kurva.load(Constants.path + "shader.vert", Constants.path + "shader.frag");
kurvaBezier.load(Constants.path + "shader.vert", Constants.path + "shader.frag");
kurva.createCurve(5.5f, 0.4f);
kurva.createCurve(5.3f, 0.8f);
kurva.createCurve(5.1f, 0.0f);
kurva.createCurve(4.9f, 1.5f);

kurva2.load(Constants.path + "shader.vert", Constants.path + "shader.frag");
kurva2Bezier.load(Constants.path + "shader.vert", Constants.path + "shader.frag");
kurva2.createCurve(4.9f, 0.4f);
kurva2.createCurve(5.1f, 1.5f);
kurva2.createCurve(5.3f, 1.5f);
kurva2.createCurve(5.5f, 1.5f);
```

Untuk rotasi

```
//KENNETH
_time4 += 100f * args.Time;
temp18 = temp18 * Matrix4.CreateTranslation(-5.0f, 0.0f, 0.0f);
temp18 = temp18 * Matrix4.CreateRotationY(MathHelper.DegreesToRadians((float)_time4));
temp18 = temp18 * Matrix4.CreateTranslation(5.0f, 0.0f, 0.0f);

_object3d[29].render(3, _time, temp17, _camera.GetViewMatrix(), _camera.GetProjectionMatrix());
_object3d[30].render(3, _time, temp16, _camera.GetViewMatrix(), _camera.GetProjectionMatrix());
_object3d[20].render(3, _time, temp17, _camera.GetViewMatrix(), _camera.GetProjectionMatrix());
_object3d[19].render(3, _time, temp17, _camera.GetViewMatrix(), _camera.GetProjectionMatrix());
_object3d[21].render(3, _time, temp17, _camera.GetViewMatrix(), _camera.GetProjectionMatrix());
```

Translasi

```
if (KeyboardState.IsKeyDown(Keys.PageUp))
{
    _time += 20f * args.Time;
    temp16 = temp16 * Matrix4.CreateTranslation(0.0f, 0.05f, 0.0f);
    temp17 = temp17 * Matrix4.CreateTranslation(0.0f, 0.05f, 0.0f);
    temp18 = temp18 * Matrix4.CreateTranslation(0.0f, 0.05f, 0.0f);
}
if (KeyboardState.IsKeyDown(Keys.PageDown))
{
    _time += 20f * args.Time;
    temp16 = temp16 * Matrix4.CreateTranslation(0.0f, -0.05f, 0.0f);
    temp17 = temp17 * Matrix4.CreateTranslation(0.0f, -0.05f, 0.0f);
    temp18 = temp18 * Matrix4.CreateTranslation(0.0f, -0.05f, 0.0f);
}
}
```

