

**Laporan UTS Grafkom**  
**Sky Life**



Kelompok 23

Anggota:

Nicholas Gunawan	c14200048
Azarya Dami	c14200090
Kenneth Poniman	c14200070

## Penjelasan singkat proyek UTS

Proyek revisi UTS kami merupakan hasil perbaikan dari proyek lama kami, dengan beberapa kesalahan sebagai berikut:

- Tema yang terlalu abstrak (tidak teratur).
- Kurangnya jenis object pada beberapa object gabungan.
- Posisi object yang saling bertabrakan atau menutupi object lain.
- Animasi yang kurang rapi/ kurang teratur.

Oleh karena itu, revisi proyek dan laporan ini kami buat dengan harapan bahwa beberapa kesalahan diatas dapat diperbaiki dan menjadi lebih baik ke depannya. Perbaikan kami lakukan pertama adalah dengan mengganti tema kami menjadi suatu tema yang lebih jelas, yaitu “Sky Life”, yang berisi kehidupan seseorang yang tinggal di atas suatu tanah yang mengawang di langit. Terdapat banyak object gabungan yang ada, antara lain orang (merupakan hasil proyek sebelumnya), kincir angin, balon udara, roket, dan beberapa object lain yang ditransformasikan. Transformasi menggunakan translasi, rotasi, dan scaling yang diatur sedemikian rupa sehingga menghasilkan transformasi yang diinginkan, seperti rotasi untuk gerakan kaki dan translasi untuk menggerakkan balon udara dengan button. Selain itu, kami juga menggunakan berbagai bentuk parametric quadric, seperti elliptic cone, box vertices, ellipsoid, hyperboloid paraboloid, dan sebagainya. Object object tersebut kami gunakan bersamaan sehingga membentuk object gabungan yang diinginkan.

Berikut kami sertakan rumus dan penjelasan singkat untuk masing masing object parametric quadric yang kami gunakan:

### Box vertices

Kode:

```
public void createBoxVertices(float x, float y, float z, float xlength, float ylength, float zlength)
{
    centerPosition.X = x;
    centerPosition.Y = y;
    centerPosition.Z = z;
    Vector3 temp_vector;

    //TITIK 1
    temp_vector.X = x - xlength / 2.0f;
    temp_vector.Y = y + ylength / 2.0f;
    temp_vector.Z = z - zlength / 2.0f;
    vertices.Add(temp_vector);
    //TITIK 2
    temp_vector.X = x + xlength / 2.0f;
    temp_vector.Y = y + ylength / 2.0f;
    temp_vector.Z = z - zlength / 2.0f;
```

```

        vertices.Add(temp_vector);
//TITIK 3
temp_vector.X = x - xlength / 2.0f;
temp_vector.Y = y - ylength / 2.0f;
temp_vector.Z = z - zlength / 2.0f;
        vertices.Add(temp_vector);
//TITIK 4
temp_vector.X = x + xlength / 2.0f;
temp_vector.Y = y - ylength / 2.0f;
temp_vector.Z = z - zlength / 2.0f;
        vertices.Add(temp_vector);
//TITIK 5
temp_vector.X = x - xlength / 2.0f;
temp_vector.Y = y + ylength / 2.0f;
temp_vector.Z = z + zlength / 2.0f;
        vertices.Add(temp_vector);
//TITIK 6
temp_vector.X = x + xlength / 2.0f;
temp_vector.Y = y + ylength / 2.0f;
temp_vector.Z = z + zlength / 2.0f;
        vertices.Add(temp_vector);
//TITIK 7
temp_vector.X = x - xlength / 2.0f;
temp_vector.Y = y - ylength / 2.0f;
temp_vector.Z = z + zlength / 2.0f;
        vertices.Add(temp_vector);
//TITIK 8
temp_vector.X = x + xlength / 2.0f;
temp_vector.Y = y - ylength / 2.0f;
temp_vector.Z = z + zlength / 2.0f;
        vertices.Add(temp_vector);

        indices = new List<uint>
        {
            //SEGITIGA DEPAN 1
            0,1,2,
            //SEGITIGA DEPAN 2
            1,2,3,
            //SEGITIGA ATAS 1
            0,4,5,

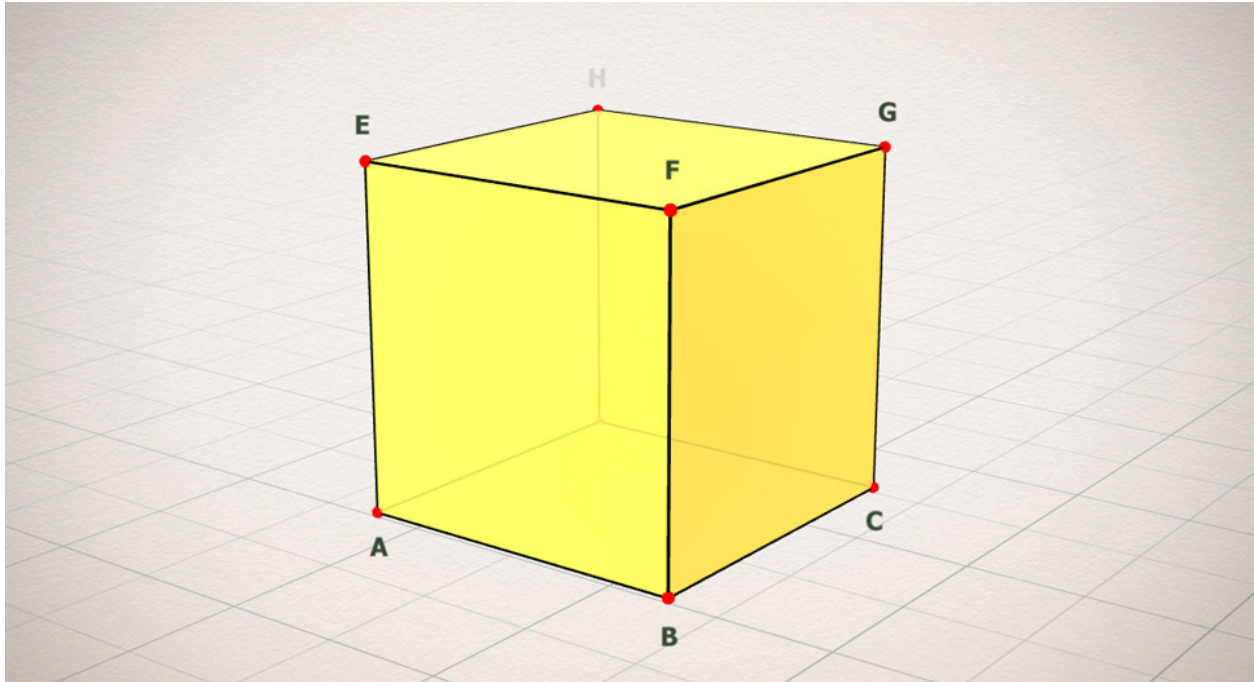
```

```

//SEGITIGA ATAS 2
0,1,5,
//SEGITIGA KANAN 1
1,3,5,
//SEGITIGA KANAN 2
3,5,7,
//SEGITIGA KIRI 1
0,2,4,
//SEGITIGA KIRI 2
2,4,6,
//SEGITIGA BELAKANG 1
4,5,6,
//SEGITIGA BELAKANG 2
5,6,7,
//SEGITIGA BAWAH 1
2,3,6,
//SEGITIGA BAWAH 2
3,6,7
};
}

```

Penjelasan: rumus tersebut merupakan rumus yang diambil dari rumus yang digunakan dosen di kelas Grafika Komputer, dengan cara pass parameter untuk panjang sisi di tiap koordinat dan posisi dari setiap titik 0 koordinat (x, y, dan z). Kemudian program akan memposisikan titik tengah tiap koordinat sesuai parameter yang ada. Setelah itu posisi x, y, dan z untuk tiap 8 titik akan diprogram dengan memperhitungkan panjang masing masing koordinat dan posisinya, dengan contoh posisi X titik 8 adalah hasil penjumlahan dari posisi koordinat x dan panjang koordinat x yang dibagi 2. Hasil ini didapatkan karena posisi titik ke 8 merupakan posisi dari tengah atas ke posisi akhir  $x / 2$ , sehingga merupakan titik yang berada di posisi bawah depan bagian kanan atau seperti titik B pada gambar berikut:



Cara menentukan posisi setiap titik adalah dengan melihat tanda pada masing masing koordinatnya, jika - maka adalah sisi kiri, bawah, atau belakang, sedangkan tanda + adalah sisi kanan, depan, atau atas, tergantung pada bagian koordinat (x merupakan kanan/ kiri, y merupakan atas/ bawah, dan z merupakan depan/ belakang). Langkah terakhir untuk membuat box vertices adalah

### Ellipsoid

Kode:

```
public void createEllipsoid(float radiusX, float radiusY, float radiusZ, float _x, float _y, float _z)
{
    _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 / 1000)
    {
        for (float v = -pi / 2; v <= pi / 2; v += pi / 1000)
        {
            temp_vector.X = _x + (float)Math.Cos(v) * (float)Math.Cos(u) * radiusX;
            temp_vector.Y = _y + (float)Math.Cos(v) * (float)Math.Sin(u) * radiusY;
            temp_vector.Z = _z + (float)Math.Sin(v) * radiusZ;
            vertices.Add(temp_vector);
        }
    }
}
```

```

    }
}

```

Penjelasan: rumus tersebut juga didapatkan dari rumus dosen Grafkomp. Langkah yang dilakukan adalah pass parameter posisi dan ukuran size. Kemudian melakukan nested loop dan didalam loop akan diatur posisi x, y, dan z tergantung dari parameter loop itu sendiri. Rumus tersebut sama dengan rumus modul Grafkomp, sebagai berikut:

Nama quadrik	$X(u,v)$	$Y(u,v)$	$Z(u,v)$	v-range	u-range
Elipsoid	$a \cos(v) \cos(u)$	$b \cos(v) \sin(u)$	$c \sin(v)$	$(-\pi/2, \pi/2)$	$(-\pi, \pi)$

Dimana loop untuk u dilakukan dari -pi hingga pi, loop v dilakukan dari -pi/2 hingga pi/2, posisi X merupakan hasil perkalian cos dan posisi Y merupakan hasil perkalian cos dan sin, serta posisi Z merupakan hasil perkalian sin. Kemudian, posisi tersebut dimasukkan ke vertices sehingga menghasilkan bola sempurna ataupun lonjong oval, tergantung dari parameter size yang dipakai.

### Elliptic cone

Kode:

```

public void createEllipticCone(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 / 1000)
    {
        for (float v = 0 ; v <= 2; v += pi / 1000)
        {
            temp_vector.X = ( _x + v * (float)Math.Cos(u) * radiusX);
            temp_vector.Y = ( _y + v * (float)Math.Sin(u) * radiusY);
            temp_vector.Z = dir*( z + v * radiusZ);
            vertices.Add(temp_vector);
        }
    }
}

```

Penjelasan: Rumus tersebut merupakan hasil dari pengembangan rumus ellipsoid dengan rumus yang ada di modul Grafkom. Langkah yang dilakukan mirip dengan ellipsoid, yaitu pass parameter, menentukan posisi x, y, dan z, serta memasukkan koordinat ke vertices. Rumus yang digunakan adalah sebagai berikut:

Nama quadrik	$X(u,v)$	$Y(u,v)$	$Z(u,v)$	v-range	u-range
Elliptic cone	$av \cos(u)$	$bv \sin(u)$	$cv$	Semua bil	$(-\pi, \pi)$

Nested loop dilakukan dari posisi u -pi ke pi dan posisi v yaitu semua bilangan dengan range yang diinginkan. Kelompok kami menggunakan range dari 0 ke 2 untuk v. Posisi X merupakan perkalian cos, posisi Y adalah perkalian sin, dan posisi Z merupakan perkalian posisi v. Hasil yang dihasilkan adalah kerucut yang menghadap belakang atau depan sesuai dengan parameter dir (1 untuk belakang, -1 untuk depan) dan dapat ditransformasikan sesuai keinginan.

### Hyperboloid paraboloid

Kode:

```
public void createHyperboloidParaboloid(float radiusX, float radiusY, float radiusZ, float _x,
float _y, float _z)
{
    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 / 1000)
    {
        for (float v = 0; v <= 3 / 2; v += pi / 1000)
        {
            temp_vector.X = _x + (float)Math.Tan(u) * radiusX * v;
            temp_vector.Y = _y + (float)(1/Math.Cos(u)) * radiusY * v;
            temp_vector.Z = _z + v * v;
            vertices.Add(temp_vector);
        }
    }
}
```

Penjelasan: Rumus tersebut merupakan rumus hyperboloid paraboloid yang didapatkan dari pengembangan rumus di modul Grafkom. Langkah yang dilakukan yaitu pass parameter yang ada, menentukan posisi, dan memasukkan koordinat ke vertices. Rumus yang digunakan adalah sebagai berikut:

Nama quadrik	$X(u,v)$	$Y(u,v)$	$Z(u,v)$	v-range	u-range
Hyperboloid paraboloid	$av \tan(u)$	$bv \sec(u)$	$v^2$	$v \geq 0$	$(-\pi, \pi)$

Nested loop dilakukan dari posisi u dari  $-\pi$  ke  $\pi$  dan posisi v yaitu semua bilangan yang diinginkan (bilangan real). Kelompok kami menggunakan range dari 0 ke  $3/2$  untuk v. Posisi X merupakan perkalian tan, posisi y merupakan perkalian secan atau  $1/\cos$ , dan posisi Z adalah hasil kuadrat dari v. Hasilnya adalah object menyerupai huruf X dan dapat ditransformasikan sesuai keinginan.

## Star

Kode:

```
public void createStar(float radiusX, float radiusY, float radiusZ, float _x, float _y, float _z)
{
    centerPosition.X = _x;
    centerPosition.Y = _y;
    centerPosition.Z = _z;
    float pi = (float)Math.PI;
    Vector3 temp_vector;
    for (float u = -pi / 2; u <= pi / 2; u += pi / 10)
    {
        for (float v = -pi / 2; v <= pi / 2; v += pi / 10)
        {
            temp_vector.X = _x + (float)Math.Tan(v) * (float)Math.Cos(u) * radiusX / 1000000;
            temp_vector.Y = _y + (float)Math.Tan(v) * (float)Math.Sin(u) * radiusY / 1000000;
            temp_vector.Z = _z + (float)(1 / Math.Cos(v)) * radiusZ / 1000000;
            vertices.Add(temp_vector);
        }
    }
}
```



Penjelasan: Rumus tersebut merupakan rumus star/ bintang 3 dimensi yang didapatkan dari pengembangan rumus Hyperboloid 2 sisi pada sisi pertama. Langkah yang dilakukan yaitu pass parameter yang ada, menentukan posisi, dan memasukkan koordinat ke vertices. Nested loop dilakukan dari posisi u ke  $-\pi/2$  hingga  $\pi/2$  dan posisi v dari  $-\pi/2$  ke  $\pi/2$  juga, sama dengan u. Posisi X merupakan perkalian tan dan cos, posisi Y merupakan perkalian sin dengan tan, dan posisi Z merupakan perkalian secan atau  $1/\cos$ . Hasilnya adalah bentuk yang menyerupai matahari/ bintang dengan ruas ruas yang saling terpisah.

### Kurva bezier

Kode:

```
public List<float> createCurveBezier()
{
    List<float> _verticesBezier = new List<float>();
    List<int> pascal = getRow(indexs - 1);
    pascal = pascal.ToArray();
    for (float t = 0; t <= 1.0f; t += 0.01f)
    {
        Vector3 p = getP(indexs, t);
        _verticesBezier.Add(p.X);
        _verticesBezier.Add(p.Y);
        _verticesBezier.Add(p.Z);
    }
    return _verticesBezier;
}
```

Penjelasan: kode kurva bezier merupakan kode yang didapat dari kelas grafkom dan diberi tambahan koordinat Z untuk posisi. Langkah yang dilakukan adalah menginisialisasi List dengan function lain yang ada, kemudian memasukkan posisi X, Y, dan Z pada vector3 yang nantinya akan direturn. Hasilnya adalah kurva bezier yang sesuai posisi X, Y, dan Z yang ditentukan.

### Torus

Kode:

```
public void createTorus(float radius1, float radius2, float _x, float _y, float _z)
{
    centerPosition.X = _x;
    centerPosition.Y = _y;
    centerPosition.Z = _z;

    float pi = (float)Math.PI;
    Vector3 temp_vector;
```

```

        for (float u = 0; u <= 2 * pi; u += pi / 700)
        {
            for (float v = 0; v <= 2 * pi; v += pi / 700)
            {
                temp_vector.X = x + (radius1 + radius2 * (float)Math.Cos(v)) *
(float)Math.Cos(u);
                temp_vector.Y = y + (radius1 + radius2 * (float)Math.Cos(v)) * (float)Math.Sin(u);
                temp_vector.Z = z + radius2 * (float)Math.Sin(v);
                vertices.Add(temp_vector);
            }
        }
    }
}

```

Penjelasan: rumus torus diatas merupakan hasil modifikasi dari ellipsoid yang diberikan lubang di tengahnya sesuai dengan parameter radius1 dan radius2. Kemudian langkah berikutnya adalah melakukan nested loop dengan posisi u dari 0 ke 2pi dan posisi v dari 0 ke 2pi. Kemudian posisi X, Y, dan Z akan dimasukkan sesuai dengan posisi dan radius yang diberikan. Rumusnya mirip dengan ellipsoid, hanya ada tambahan radius2 sebagai radius sisi dalam/ lubang. Sehingga yang berisi ellipsoid hanya dari posisi angka radius2 hingga radius1 saja dan sisanya berupa lubang. Hasilnya adalah donat atau torus, yaitu lingkaran dengan lubang di bagian tengah.

### Elliptic paraboloid

Kode:

```

public void createEllipticParaboloid(float radiusX, float radiusY, float radiusZ, float x, float y,
float z)
{
    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 / 1000)
    {
        for (float v = 0; v <= 3 / 2; v += pi / 1000)
        {
            temp_vector.X = x + (float)Math.Cos(u) * radiusX * v;
            temp_vector.Y = y + (float)Math.Sin(u) * radiusY * v;
            temp_vector.Z = -1 * ( z + v * v * radiusZ);
            vertices.Add(temp_vector);
        }
    }
}

```

```

    }
}

```

Penjelasan: rumus elliptic paraboloid di atas merupakan rumus yang dihasilkan dari rumus di modul Grafkom. Langkah yang dilakukan adalah inisialisasi parameter posisi dan ukuran, kemudian looping dan memasukkan posisi X, Y, dan Z. Rumus yang digunakan adalah:

Nama quadrik	$X(u,v)$	$Y(u,v)$	$Z(u,v)$	v-range	u-range
Elliptic paraboloid	$av \cos(u)$	$bv \sin(u)$	$v^2$	$v \geq 0$	$(-\pi, \pi)$

Nested loop dilakukan dari v dengan range semua bilangan yang diinginkan dan u dengan range dari -pi hingga pi. Posisi X merupakan perkalian cos, posisi Y merupakan perkalian sin, dan posisi Z adalah hasil kuadrat dari v. Hasilnya adalah bentuk menyerupai sepatu yang menghadap bawah.

Bagian individu masing masing anggota adalah sebagai berikut:

**Nicholas Gunawan - c14200048**

### Object

Bagian individu saya adalah membuat object gabungan orang yang merupakan gabungan dari 5 jenis parametric quadric, yaitu box vertices, ellipsoid, dan elliptic cone serta torus dan elliptic paraboloid untuk gelang tangan kanan dan 1 kurva bezier. Object orang ini merupakan object interaktif yang bisa dikontrol dan terdiri dari anggota tubuh lengkap dengan topi dari elliptic cone yang dirotasikan. Object gabungan orang tersebut adalah:



Object tersebut terdiri dari:

Muka, dengan ukuran 0.25 di semua koordinat dan posisi tepat di sumbu X dan Z serta 0.5 di sumbu Y serta warna krem kulit, kode:

```

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);

```

Hasilnya adalah:



Badan, dengan ukuran 0.5 di semua sisi dan posisi tepat di ketiga sumbu serta warna biru, kode:

```
object3d[0] = new Asset3D(new Vector3(0, 0, 1.0f));  
object3d[0].createBoxVertices(0f, 0, 0, 0.5f, 0.5f, 0.5f);
```

Hasilnya adalah:



Mata kanan, dengan ukuran 0.05 di semua sisi dan posisi 0.1 dari sumbu X, 0.6 dari sumbu Y dan warna putih, dan 0.25 dari sumbu Z, kode:

```
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, 0.25f);
```

Hasilnya adalah:



Mata kiri, dengan ukuran 0.05 di semua sisi dan posisi mirip dengan mata kanan, hanya berbeda pada posisi X, yaitu -0.1 dari sumbu X dengan warna putih, kode:

```
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, 0.25f);
```

Hasilnya adalah:



Mulut, dengan kurva bezier dan posisi yang merupakan gabungan dari 4 titik dengan warna merah, kode:

```
mouth.createCurve(-0.15f, 0.45f, 0.25f);  
mouth.createCurve(-0.1f, 0.4f, 0.25f);  
mouth.createCurve(0.1f, 0.4f, 0.25f);  
mouth.createCurve(0.15f, 0.45f, 0.25f);
```

Hasilnya adalah:



Alis kanan, dengan kurva bezier dan posisi gabungan dari 4 titik dengan warna hitam, kode:

```
eyebrow2.createCurve(0.025f, 0.7f, 0.25f);  
eyebrow2.createCurve(0.075f, 0.75f, 0.25f);  
eyebrow2.createCurve(0.125f, 0.75f, 0.25f);  
eyebrow2.createCurve(0.175f, 0.7f, 0.25f);
```

Hasilnya adalah:



Alis kiri, dengan kurva bezier dan posisi gabungan 4 titik dengan warna hitam, kode:

```
eyebrow.createCurve(-0.175f, 0.7f, 0.25f);  
eyebrow.createCurve(-0.125f, 0.75f, 0.25f);  
eyebrow.createCurve(-0.075f, 0.75f, 0.25f);  
eyebrow.createCurve(-0.025f, 0.7f, 0.25f);
```

Hasilnya adalah:



Topi menggunakan elliptic cone dengan ukuran 0.125 di semua sisi dan posisi tepat di sumbu X, 1.4 dari sumbu Y dan -0.7 dari sumbu Z dengan direction 1 dan warna ungu, kode:

```
object3dNicho[12] = new Asset3D(new Vector3(0.5f, 0.0f, 0.5f));  
object3dNicho[12].createEllipticCone(0.125f, 0.125f, 0.15f, 0, 1.4f, -0.7f, 1);
```

Topi ini dirotasi 90 derajat terhadap sumbu X dan ditranslasikan sehingga berada tepat di atas kepala player dan berbentuk kerucut, kode:

```
temp3 = temp3 * Matrix4.CreateRotationX(MathHelper.DegreesToRadians(90f));  
temp3 = temp3 * Matrix4.CreateTranslation(0, 0.3f, -1.35f);
```

Hasilnya adalah:



Gelang, dibuat dari torus dengan ukuran 0.065 dari sumbu X, 0.015 dari sumbu Y, dan 0.3 dari sumbu Z dengan posisi 0.3, -0.2, dan 0.25 serta warna kuning keemasan. Kode:

```
object3dNicho[15] = new Asset3D(new Vector3(1f, 1f, 0));  
object3dNicho[15].createTorus(0.065f, 0.015f, 0.3f, -0.2f, 0.25f);
```

Gelang tersebut dirotasi 90 derajat dari sumbu X dan ditranslasikan sehingga berada di tangan kanan player, kode:

```
temp1 = temp1 * Matrix4.CreateRotationX(MathHelper.DegreesToRadians(90f));  
temp1 = temp1 * Matrix4.CreateTranslation(0, -0.095f, 0.25f);
```

Hasilnya adalah:



Tangan kanan, dibuat dengan ukuran 0.05, 0.35, dan 0.15 sedangkan posisi menggunakan 0.33, -0.15, dan 0 dan warna krem, kode:

```
object3dNicho[3] = new Asset3D(new Vector3(0.9f, 0.5f, 0.4f));  
object3dNicho[3].createEllipsoid(0.05f, 0.35f, 0.15f, 0.3f, -0.15f, 0);
```

Hasilnya adalah:



Tangan kiri, dengan ukuran dan posisi serta warna yang sama dengan tangan kanan, hanya berbeda di posisi x, yaitu -0.3, kode:

```
object3dNicho[2] = new Asset3D(new Vector3(0.9f, 0.5f, 0.4f));  
object3dNicho[2].createEllipsoid(0.05f, 0.35f, 0.15f, -0.3f, -0.15f, 0);
```

Hasilnya adalah:



Kaki kanan, dengan ukuran 0.05, 0.35, dan 0.15 serta posisi 0.1, -0.5, dan 0, warna krem kulit, kode:

```
object3dNicho[4] = new Asset3D(new Vector3(0.9f, 0.5f, 0.4f));
object3dNicho[4].createEllipsoid(0.05f, 0.3f, 0.15f, 0.1f, -0.5f, 0);
```

Hasilnya adalah:



Kaki kiri, ukuran dan posisi serta warna sama dengan kaki kanan hanya berbeda di posisi x, yaitu -0.1, kode:

```
object3dNicho[5] = new Asset3D(new Vector3(0.9f, 0.5f, 0.4f));
object3dNicho[5].createEllipsoid(0.05f, 0.3f, 0.15f, -0.1f, -0.5f, 0);
```

Hasilnya adalah:



Bola mata kanan, menggunakan ukuran 0.01 di semua sisi dan posisi menggunakan 0.1, 0.6, dan 0.3 dengan warna hitam, kode:

```
object3dNicho[11] = new Asset3D(new Vector3(0f, 0f, 0f));
object3dNicho[11].createEllipsoid(0.01f, 0.01f, 0.01f, 0.1f, 0.6f, 0.3f);
```

Hasilnya adalah:



Bola mata kiri, menggunakan posisi dan ukuran serta warna yang sama dengan bola mata kanan dengan posisi X -0.1, kode:

```
object3dNicho[10] = new Asset3D(new Vector3(0f, 0f, 0f));
object3dNicho[10].createEllipsoid(0.01f, 0.01f, 0.01f, -0.1f, 0.6f, 0.3f);
```

Hasilnya adalah:



Sepatu kanan, menggunakan elliptic paraboloid dengan ukuran 0.125 di semua sisi dan posisi menggunakan 1.25, 0.025, dan 0.02 dan warna hitam, kode:

```
object3dNicho[13] = new Asset3D(new Vector3(0, 0, 0));
object3dNicho[13].createEllipticParaboloid(0.125f, 0.125f, 0.125f, 1.25f, 0.025f, 0.2f);
```

Sepatu kemudian diputar 270 derajat terhadap sumbu X dan ditranslasikan serta diatur skalanya sehingga berada di bawah kaki kanan, kode:

```
temp02 = temp02 * Matrix4.CreateRotationX(MathHelper.DegreesToRadians(270f));
temp02 = temp02 * Matrix4.CreateTranslation(-1.15f, -0.45f, 0.125f);
temp02 = temp02 * Matrix4.CreateScale(1, 1.05f, 1);
```

Hasilnya adalah:



Sepatu kiri, menggunakan posisi dan ukuran serta warna yang sama dengan sepatu kanan, posisi X yang digunakan adalah 1, kode:

```
object3dNicho[14] = new Asset3D(new Vector3(0, 0, 0));  
object3dNicho[14].createEllipticParaboloid(0.125f, 0.125f, 0.125f, 1, 0.025f, 0.2f);
```

Hasilnya adalah:



Motif S pada baju, menggunakan kurva bezier dengan 6 titik dan warna pink, kode:

```
shirt.createCurve(0.1f, 0.25f, 0.26f);  
shirt.createCurve(-0.1f, 0.25f, 0.26f);  
shirt.createCurve(-0.1f, 0.1f, 0.26f);  
shirt.createCurve(0.1f, 0.1f, 0.26f);  
shirt.createCurve(0.1f, -0.05f, 0.26f);  
shirt.createCurve(-0.1f, -0.05f, 0.26f);
```

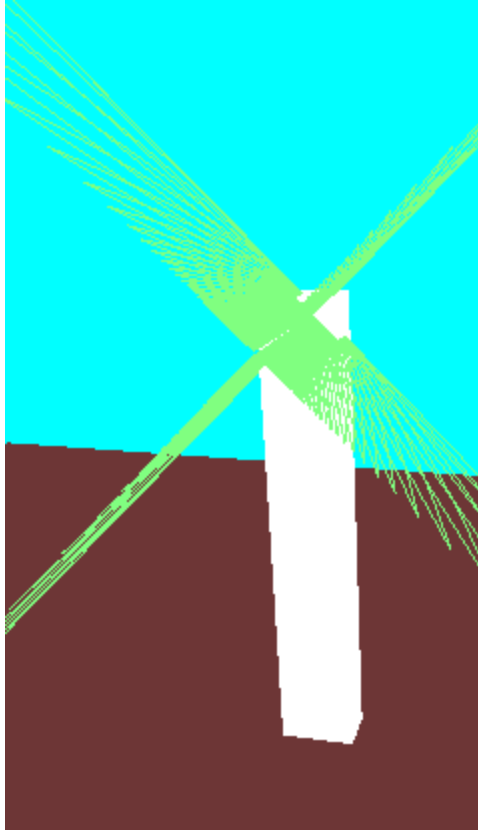
Hasilnya adalah:



Selain object orang/ player, terdapat juga object gabungan berupa kincir angin dan matahari.

Kincir angin merupakan gabungan 2 object, yaitu box vertices dan hyperboloid paraboloid.

Object gabungan kincir angin tersebut adalah:



Object tersebut terdiri dari:

Tiang kincir angin, dibentuk dari box vertices dengan ukuran -1.6, 0.35, dan 0.5 serta posisi 0.5, 3.5, dan 0.25 dan warna putih, kode:

```
object3dNicho[6] = new Asset3D(new Vector3(1f, 1.0f, 1.0f));  
object3dNicho[6].createBoxVertices(-1.6f, 0.35f, 0.5f, 0.5f, 3.5f, 0.25f);
```

Hasilnya adalah:

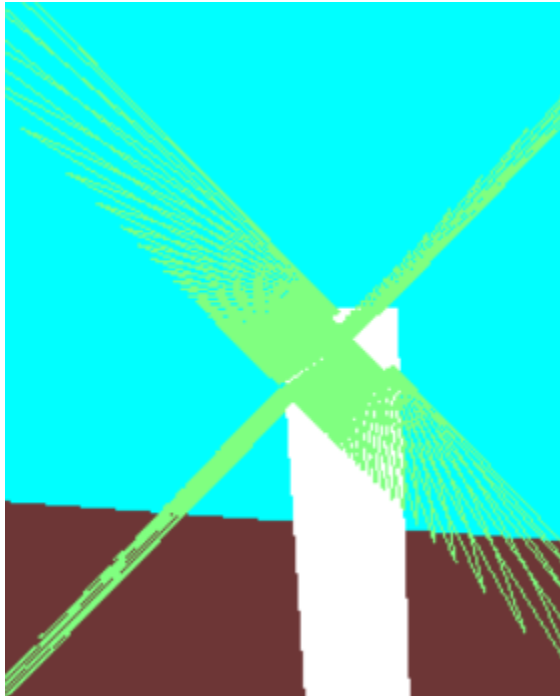


Baling baling, dibentuk dari hyperboloid paraboloid dengan ukuran 0.015 di semua sisi, serta posisi -1.5, 2, dan 0.5 dengan warna hijau muda, kode:

```
object3dNicho[17] = new Asset3D(new Vector3(0.5f, 1.0f, 0.5f));  
object3dNicho[17].createHyperboloidParaboloid(0.015f, 0.015f, 0.015f, -1.5f, 2f, 0.5f);
```



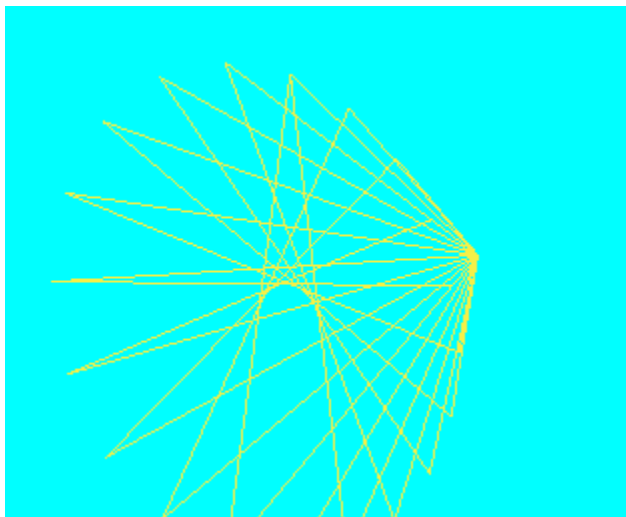
Hasilnya adalah:



Object terakhir yang saya buat adalah matahari/ bintang yang merupakan object 3D star dengan ukuran 0.125 di semua sisi dan posisi 5.5, 14.5, dan 0, kode:

```
object3dNicho[16] = new Asset3D(new Vector3(0.960f, 0.941f, 0.278f));  
object3dNicho[16].createStar(0.125f, 0.125f, 0.125f, 5.5f, 14.5f, 0f);
```

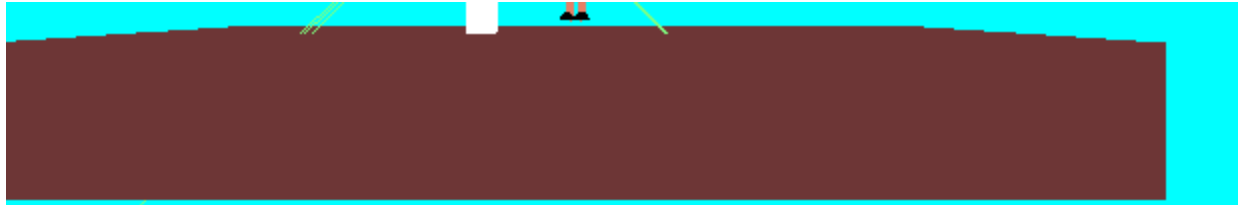
Hasilnya adalah:



Untuk environment, saya menggunakan tanah/ land dengan kode:

```
object3dNicho[7] = new Asset3D(new Vector3(0.427f, 0.211f, 0.211f));  
object3dNicho[7].createBoxVertices(0, -2f, 0, 15.0f, 2.0f, 8.0f);
```

Hasilnya adalah:



Jenis parametric quadric dan object yang saya gunakan dapat dilihat pada tabel berikut:

Parametric quadric/ object	Nomor index	Nama
Box vertices	0, 6, 7	Badan, tiang kincir, land
Ellipsoid	1, 2, 3, 4, 5, 8, 9, 10, 11, 18	Muka, kedua mata, kedua tangan, kedua kaki, kedua bola mata, hidung
Elliptic cone	12	Topi
Kurva bezier	- (object2d tunggal)	Motif baju, kedua alis, mulut
Elliptic paraboloid	13, 14	Kedua sepatu
Torus	15	Gelang tangan
Hyperboloid paraboloid	17	Baling baling kincir

Dapat disimpulkan bahwa saya menggunakan 26 object dan 7 jenis bentuk parametric quadric/object untuk pembuatan bagian individu saya pada proyek.

### Transformasi

Untuk transformasi bagian saya, saya menggunakan 11 variabel Matrix4 untuk menyimpan transformasi matriks dan menggunakan time untuk parameter render. Kegunaan dan transformasi yang ada untuk ke 11 variabel temp tersebut adalah sebagai berikut:

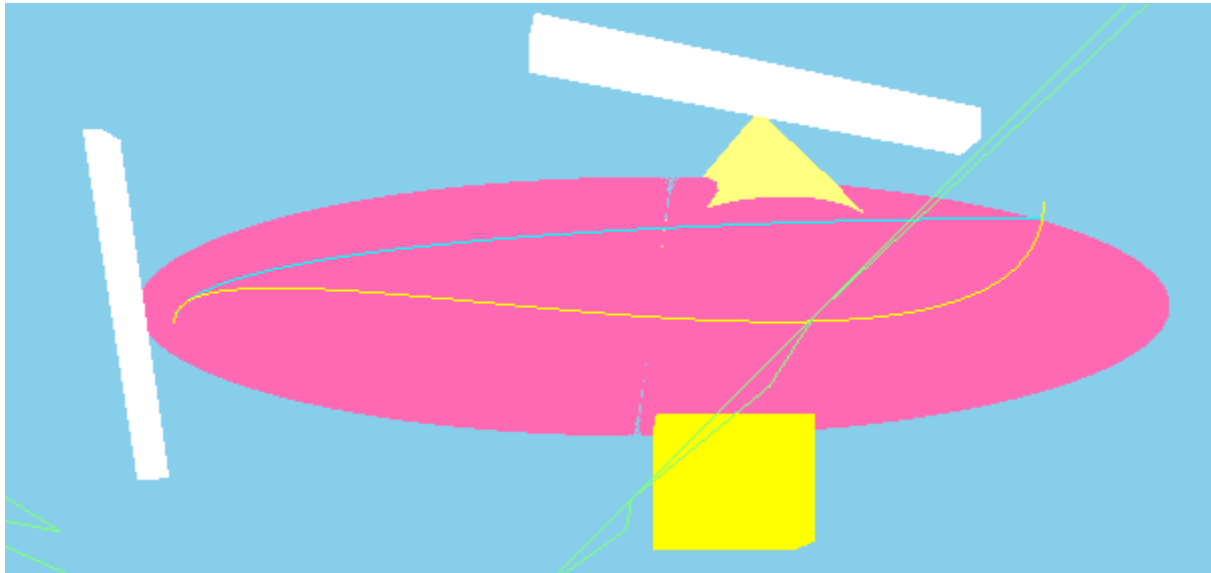
Variabel temp	Nomor index object	Kegunaan	Transformasi penyesuaian	Transformasi pengulangan	Trigger event
temp	0, 1, 2, 3, 8, 9, 10, 11, 18	Penggerak beberapa anggota tubuh	-	Translasi	User click up, down, left, right
temp0	13	Penggerak sepatu kanan agar sejajar	Rotasi sumbu X, translasi, scaling	Translasi, rotasi sumbu X	User click up, down, left, right

		dengan kaki kanan			
temp02	14	Penggerak sepatu kiri agar sejajar dengan kaki kiri	Rotasi sumbu X, translasi, scaling	Translasi, rotasi sumbu X	User click up, down, left, right
temp1	15	Pengatur posisi gelang	Rotasi sumbu X, translasi	Translasi	User click up, down, left, right
temp2	7	Untuk land agar statis	-	-	-
temp3	12	Pengatur posisi topi cone	Rotasi sumbu X, translasi	Translasi	User click up, down, left, right
temp4	16	Pengatur gerak matahari/ star	-	Rotasi sumbu Z, scaling	User click CapsLock (rotasi) atau click + atau - (scaling)
temp5	5	Penggerak kaki kiri	-	Rotasi sumbu X, translasi	User click up, down, left, right
temp6	4	Penggerak kaki kanan	-	Rotasi sumbu X, translasi,	User click up, down, left, right
temp7	17	Pengatur baling baling	-	-	-
temp8	6	Pengatur tiang kincir	-	-	-

Dapat disimpulkan bahwa saya menggunakan semua jenis transformasi, yaitu rotasi, scaling, dan translasi sesuai kebutuhan dan bersifat interaktif, yaitu dapat digerakkan dengan tombol pada KeyBoard. Selain untuk animasi, transformasi juga saya gunakan untuk penyesuaian seperti untuk rotasi agar sesuai dan juga untuk translasi serta scaling. Untuk animasi saya juga menggunakan scaling, translasi, dan rotasi sesuai kebutuhan.


**Azarya Dami - c14200090**






## Object



Saya telah membuat sebuah object berkarakter atau berbentuk seperti balon udara bertenaga mesin dan baling-baling. Object gabungan yang saya buat untuk membentuk balon udara ini terdiri dari beberapa jenis parametric quadric yaitu 1 ellipsoid untuk badan balon udara, 3 box vertices untuk kabin pilot, baling-baling belakang dan baling-baling atas, serta 1 elliptic cone untuk perekat baling-baling atas dari badan balon udara. Terdapat juga kurva yang berada pada objek ellipsoid tersebut. Selain itu, juga terdapat objek terpisah yang saya buat menyerupai bulan menggunakan ellipsoid.

Pembuatan Objek tersebut dirincikan sebagai berikut:

No.	Nama Parametric / Objek dan Kurva	Source Code
1.	Ellipsoid → Badan balon udara 	<pre>//badan pesawat _object3dSutan[1] = new Asset3D(new Vector3(1.0f, 0.411f, 0.705f)); _object3dSutan[1].createEllipsoid(3.5f, 0.8f, 0.9f, 0.3f, 5.2f, -0.5f);</pre> <p>Membuat bentuk ellipsoid dengan 3 parameter awal untuk menentukan ukuran balon udara dengan <math>x = 3.5f</math>, <math>y = 0.8f</math>, dan <math>z = 0.3f</math> sedangkan 3 parameter akhir untuk menentukan posisi badan balon udara lebih ke atas (posisi terbang) dengan <math>y = 5.2f</math>. Warna rgb dari badan balon pesawat yaitu <math>1.0f</math>, <math>0.41f</math>, dan <math>0.705f</math> sehingga menghasilkan warna pink (merah muda).</p>

2.	Box Vertices → Baling-baling belakang 	<pre>//baling" belakang _object3dSutan[0] = new Asset3D(new Vector3(1.0f, 1.0f, 1.0f)); _object3dSutan[0].createBoxVertices(-3.0f, 5.0f, 0.0f, 0.1f, 2.0f, 0.2f);</pre> <p>Membuat bentuk box vertices atau segi empat dengan 3 parameter awal untuk menentukan posisi baling-baling yang melekat pada bagian belakang badan balon udara. 3 parameter terakhir merupakan ukuran baling-baling. Warna rgb yang diberikan menghasilkan warna saat di render putih.</p>
3.	Box Vertices → Kabin Pilot 	<pre>//kabin pilot _object3dSutan[2] = new Asset3D(new Vector3(1.0f, 1.0f, 0.0f)); _object3dSutan[2].createBoxVertices(0.8f, 4.0f, -0.3f, 1.0f, 0.8f, 0.5f);</pre> <p>Kabin pilot dibuat berbentuk balok segi empat dan diletakkan menempel pada bagian bawah balon udara. Letak <math>y = 4.0f</math> yang berada sedikit di bawah badan balon udara. Warna yang dihasilkan adalah warna kuning (<i>light yellow</i>).</p>
4.	Elliptic Cone → Perekat baling-baling atas. 	<pre>//Perekat baling-baling atas _object3dSutan[3] = new Asset3D(new Vector3(1.5f, 1.0f, 0.5f)); _object3dSutan[3].createEllipticCone(0.4f, 0.4f, 0.5f, 1, 1.0f, -6.2f, 1);</pre> <p>Perekat baling-baling atas berbentuk kerucut dengan ukuran diameter <math>x</math> dan <math>y</math> adalah <math>0.4f</math> sedangkan <math>z</math> berukuran <math>0.5f</math>. Karena posisinya telah diubah menggunakan parameter ke terakhir yaitu <math>1</math>. <math>1</math> tersebut akan menentukan arah dari kerucut yang dibuat. Hal ini menyebabkan posisi dan letak yang harus disesuaikan. Warna rgb yang dihasilkan juga berwarna kuning (<i>egg yellow</i>).</p>
5.	Elliptic Paraboloid → Scope / teropong 	<pre>//scope _object3dSutan[4] = new Asset3D(new Vector3(0.58f, 0.29411f, 0.0f)); _object3dSutan[4].createEllipticParaboloid(0.1f, 0.1f, -1.0f, -0.3f, 5.0f, 0.0f);</pre> <p>Scope atau teropong berbentuk lonjong dan dibuat dengan rumus elliptic paraboloid. Memiliki ukuran <math>x</math> dan <math>y</math> yang sangat kecil yaitu <math>0.1f</math> dan diletakkan pada badan balon udara dengan <math>y = 5.0f</math>, <math>x = -0.3f</math> (agak ke kiri), dan <math>z = -1.0f</math> (lebih masuk/mundur ke dalam badan balon udara). Warna yang dihasilkan adalah warna coklat tua hampir mendekati warna hitam.</p>
6.	Baling-baling atas 	<pre>//baling-baling atas _object3dSutan[6] = new Asset3D(new Vector3(1.0f, 1.0f, 1.0f)); _object3dSutan[6].createBoxVertices(1.0f, 6.4f, 0.0f, 3.0f, 0.2f, 0.2f);</pre> <p>Membuat baling-baling atas dengan box vertices sehingga membentuk segi empat memanjang dengan ukuran yang telah disesuaikan serta berada pada posisi yang melekat tepat pada perekat baling-baling atas. Warna yang dihasilkan warna putih.</p>

7.

## Kurva Beazier



```
//SUTAN
//les1
Asset2D moon = new Asset2D(
    new float[1080],
    new uint[] { },
    new float[] { 0, 0, 1.0f });

Asset2D moonBezier = new Asset2D(
    new float[] { },
    new uint[] { },
    new float[] { 1.0f, 1.0f, 0f });

//les2
Asset2D les = new Asset2D(
    new float[1080],
    new uint[] { },
    new float[] { 0, 1.0f, 1.0f });

Asset2D lesBezier = new Asset2D(
    new float[] { },
    new uint[] { },
    new float[] { 0, 1.0f, 1.0f });
```


Membuat kurva menggunakan constructor dari class Asset2D. Kurva yang dibuat ada 2 macam dengan warna yang berbeda yaitu kuning dan biru muda (*light blue*).

```
//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, 0.4f);
moon.createCurve(-2.5f, 5.5f, 0.4f);
moon.createCurve(2.5f, 4.0f, 0.4f);
moon.createCurve(2.5f, 5.5f, 0.4f);

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, 0.4f);
les.createCurve(-2.5f, 5.4f, 0.4f);
les.createCurve(2.5f, 5.4f, 0.4f);
les.createCurve(2.5f, 5.4f, 0.4f);
```

Pada saat di load kurva akan di-*create* dengan ukuran x, y, z yang disesuaikan antara kedua kurva sehingga saling bertemu membentuk dekorasi les yang saling membantu.

8.	Ellipsoid → bulan 	<pre>//bulan _object3dSutan[5] = new Asset3D(new Vector3(0.81f, 0.81f, 0.81f)); _object3dSutan[5].createEllipsoid(2.5f, 2.5f, 2.5f, -15.3f, -23.0f, -3.5f);</pre> <p>Membentuk bulan menggunakan ellipsoid dengan diameter yang sama yaitu 2.5f pada x, y, dan z. Letak bulan berada di bawah dari objek yang lain sehingga berada pada -23.0f sumbu y. Warna yang dihasilkan adalah warna abu-abu.</p>
----	--	---

### Transformasi

Variabel temp	Nomor index object	Kegunaan	Transformasi penyesuaian	Transformasi pengulangan	Trigger event
temp9	0	Penggerak baling-baling belakang untuk bergerak berputar dan berpindah tempat	Rotasi sumbu X dan translasi	Translasi, rotasi sumbu X	User click F1 untuk bergerak ke arah kiri (mundur) dan F2 untuk bergerak ke arah kanan (maju)
temp10	6	Penggerak baling-baling atas untuk bergerak berputar dan berpindah tempat	Rotasi sumbu Y dan translasi	Translasi, rotasi sumbu Y	User click F1 untuk bergerak ke arah kiri (mundur) dan F2 untuk bergerak ke arah kanan(maju)
temp11	3	Perekat baling-baling atas	-	Translasi	User click F1 untuk bergerak ke arah kiri (mundur) dan F2 untuk bergerak ke arah kanan(maju)
temp14	4	Penggerak scope pada balon udara	-	Translasi	User click F1 untuk bergerak ke arah kiri (mundur) dan F2 untuk bergerak ke arah kanan(maju)
temp15	1	Badan	-	Rotasi sumbu	User click F1

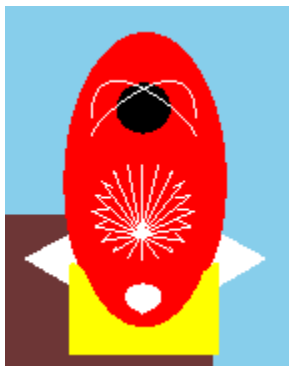
		Pesawat		Z, scaling	untuk bergerak ke arah kiri (mundur) dan F2 untuk bergerak ke arah kanan(maju)
temp19	5	bulan	-	Rotasi sumbu Z dan scaling	User click CapsLock (rotasi) atau click + (memperbesar) atau - (memperkecil) scaling)

Kesimpulan dari objek yang saya buat telah memenuhi minimal objek parametric kuadrik yang telah ditentukan yaitu saya menggunakan 4 macam parametric kuadrik dan 1 jenis kurva total saya menggunakan 9 objek yang terdiri dari 8 objek gabungan (balon udara) dan 1 jenis objek ellipsoid yaitu bulan yang terpisah. Selain itu, telah menggunakan rotasi pada sumbu X dan Y, translasi, serta scaling.

## **Kenneth Poniman - c14200070**

### **Object**

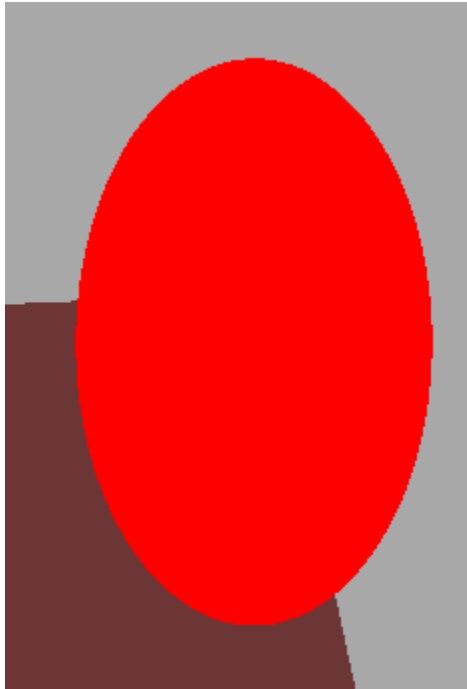
Bagian individu saya adalah membuat object gabungan roket yang merupakan gabungan dari 4 jenis parametric quadric, yaitu box vertices, ellipsoid, dan elliptic cone serta star dan 1 kurva bezier. Object roket ini merupakan object interaktif yang bisa dikontrol dan terdiri dari sayap roket, jendela roket, dan beberapa gambar yang dapat dirotasikan. Object gabungan roket tersebut adalah:



Untuk Source Code nya:

```
//tengah roket
_object3dKenneth[0] = new Asset3D(new Vector3(1.0f, 0.0f, 0.0f));
_object3dKenneth[0].createEllipsoid(0.5f, 0.9f, 0.5f, 7.0f, 1.0f, 0.0f);
```





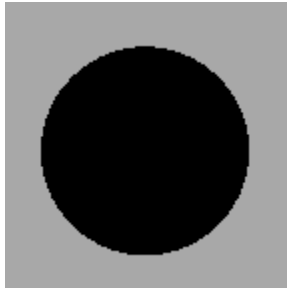
Membuat bentuk ellipsoid dengan 3 parameter awal untuk menentukan ukuran balon udara dengan  $x = 0.5f$ ,  $y = 0.9f$ , dan  $z = 0.5f$  sedangkan 3 parameter akhir untuk menentukan posisi badan balon udara lebih ke atas (posisi terbang) dengan  $x=7.0f$ ,  $y = 1.0f$ . Warna rgb dari badan balon pesawat yaitu  $1.0f$ ,  $0.0f$ , dan  $0.0f$  sehingga menghasilkan warna merah.

```
//bawah roket
_object3dKenneth[1] = new Asset3D(new Vector3(1.0f, 1.0f, 0.0f));
_object3dKenneth[1].createBoxVertices(7.0f, 0.3f, 0.0f, 0.9f, 0.5f, 0.5f);
```



Membuat bentuk Box vertice 3 parameter awal posisi dan 3 parameter akhir ukuran sama dengan tengah roket hanya berubah x dan y untuk warna menghasilkan warna kuning/

```
//jendela bulat
_object3dKenneth[2] = new Asset3D(new Vector3(0.0f, 0.0f, 0.0f));
_object3dKenneth[2].createEllipsoid(0.2f, 0.2f, 0.2f, 7.0f, 1.5f, 0.25f);
```



Elipsoid dengan 3 parameter awal ukuran dan 3 parameter akhir posisi dan memiliki warna hitam

```
//sirip depan
degr1 = MathHelper.DegreesToRadians(180f);
degr3 = MathHelper.DegreesToRadians(90f);
_object3dKenneth[3] = new Asset3D(new Vector3(1.0f, 1.0f, 1.0f));
_object3dKenneth[3].createEllipticCone(0.125f, 0.125f, 0.25f, 7.0f, 0.5f, -0.6f, 1);

//sirip belakang
_object3dKenneth[4] = new Asset3D(new Vector3(1.0f, 1.0f, 1.0f));
_object3dKenneth[4].createEllipticCone(0.125f, 0.125f, 0.25f, 7.0f, -0.5f, -0.6f, 1);

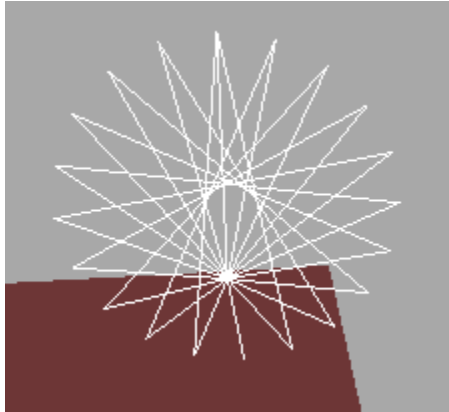
//Sirip kiri
_object3dKenneth[5] = new Asset3D(new Vector3(1.0f, 1.0f, 1.0f));
_object3dKenneth[5].createEllipticCone(0.125f, 0.125f, 0.25f, 0.0f, 0.5f, 6.25f, 1);

//Sirip kanan
_object3dKenneth[6] = new Asset3D(new Vector3(1.0f, 1.0f, 1.0f));
_object3dKenneth[6].createEllipticCone(0.125f, 0.125f, 0.25f, 0.0f, 0.5f, -7.75f, 1);
```



Sayap nya sama semua aku pakai elliptic cone hanya disini aku merubah posisi menghadapnya menggunakan temp, degr1, dan degr3. Dan 3 parameter awal ukuran, 3 parameter akhir posisi. Dan warna nya sama semua putih

```
//star
_object3dKenneth[7] = new Asset3D(new Vector3(1.0f, 1.0f, 1.0f));
_object3dKenneth[7].createStar(0.0125f, 0.0125f, 0.0125f, 7.0f, 1.0f, 0.75f);
```



Star 3 parameter awal untuk ukuran dan 3 parameter akhir untuk posisi. Warna nya putih dan ini untuk hiasan

```
//KENNETH
kurva.load(Constants.path + "shader.vert", Constants.path + "shader.frag");
kurvaBezier.load(Constants.path + "shader.vert", Constants.path + "shader.frag");
kurva.createCurve(6.7f, 1.4f, 0.4f);
kurva.createCurve(6.7f, 1.5f, 0.4f);
kurva.createCurve(7.3f, 2.0f, 0.4f);
kurva.createCurve(7.3f, 1.5f, 0.4f);

kurva2.load(Constants.path + "shader.vert", Constants.path + "shader.frag");
kurva2Bezier.load(Constants.path + "shader.vert", Constants.path + "shader.frag");
kurva2.createCurve(7.3f, 1.4f, 0.4f);
kurva2.createCurve(7.3f, 1.5f, 0.4f);
kurva2.createCurve(6.7f, 2.0f, 0.4f);
kurva2.createCurve(6.7f, 1.5f, 0.4f);
```



Kurva untuk hiasan

Jenis parametric quadric dan object yang saya gunakan dapat dilihat pada tabel berikut:

Parametric quadric/ object	Nomor index	Nama
----------------------------	-------------	------

Box vertices	1	Bagian Bawah roket
Elipsoid	0,2	Bagian tengah roket dan jendela roket
Elliptic cone	3,4,5,6	Sayap roket
Kurva bezier	- (object2d tunggal)	hiasan
Star	7	hiasan

### Transformasi

Variabel temp	Nomor index object	Kegunaan	Transformasi penyesuaian	Transformasi pengulangan	Trigger event
temp16	4	Sirip belakang. Untuk mengubah posisi ke arah depan, rotasi, dan bergerak ke atas dan bawah	Rotasi Sumbu Y dan translasi	Translasi, rotasi sumbu Y	User click U dan P
temp17	0,1,2,7, kurva	Bagian tengah roket, bawah roket, jendela roket, dan hiasan. Untuk bergerak ke atas dan bawah	translasi	Translasi	User click U dan P
temp18	3	Sirip depan Untuk mengubah posisi ke arah kebelakang, rotasi, dan bergerak ke	Rotasi sumbu Y dan translasi	Translasi, rotasi sumbu Y	User click U dan P

		atas dan bawah			
temp20	5	Sirip kiri Untuk mengubah posisi ke arah kiri, rotasi, dan bergerak ke atas dan bawah	Rotasi sumbu Y dan translasi	Translasi, rotasi sumbu Y	User click U dan P
temp21	6	Sirip kanan Untuk mengubah posisi ke arah kanan, rotasi, dan bergerak ke atas dan bawah	Rotasi sumbu Y dan translasi	Translasi, rotasi sumbu Y	User click U dan P