LAPORAN PEMBUATAN OBJEK

**Untuk Memenuhi Nilai Ujian Tengah Semester**

**Mata Kuliah Grafika Komputer Semester Ganjil Tahun Ajaran 2021**

**Disusun oleh :**

**Wilson Otto Sasongko - C14190017**

**Denzel Daniel D'Assante Tangsaputra - C14200160**

**Andreas Hariyanto - C14190198**

**Dosen:**

**LILIANA, S.T., M.Eng., Ph.D.**

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS TEKNOLOGI INDUSTRI**

**UNIVERSITAS KRISTEN PETRA**

**SURABAYA**

# Pendahuluan

Untuk memenuhi nilai Ujian Tengah Semester mata kuliah Grafika Komputer, kelompok kami menemukan ide tema yaitu “*Latest Technologies from the Myssilia Clan*” atau “Teknologi Terbaru dari Klan *Myssilia*”. Kelompok kami akan membuat objek dengan menggunakan C#.

# Pembuatan Objek

## Battle Spaceship (oleh: Wilson)

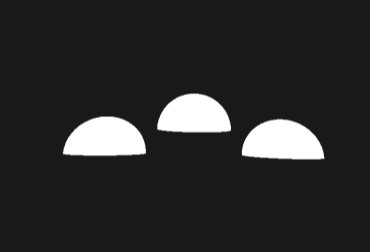
Battle Spaceship ini terdiri dari bagian objek:

1. Setengah Bola (Potongan Bola)

Objek dibawah ini digunakan pada bagian di depan badan untuk kaca sebagai penglihatan. Potongan bola tersebut dapat dibuat oleh fungsi createFractionedEllipsoid() dan di *translate* dan di *rotate* hingga menghasilkan posisi yang diinginkan.



Terdapat tiga potongan bola yang berfungsi sebagai tembok belakang spaceship. Objek tersebut dapat dibuat oleh createFractionedEllipsoid() dan di *translate* dan di *rotate* hingga menghasilkan posisi yang diinginkan.

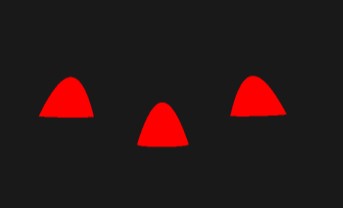


Terdapat tiga potongan bola berwarna putih yang berfungsi sebagai tembok depan dari spaceship, menghubungkan jendela dan tembok samping. Potongan bola tersebut dapat dibuat oleh fungsi createFractionedEllipsoid() dengan tambahan parameter *startPoint*, sehingga penggambaran bola dimulai dari titik tertentu dari bola. Dengan tambahan parameter tersebut, terbuat sebuah potongan bola yang berlubang. Ketiga potongan bola tersebut akan di *translate* dan di *rotate* hingga menghasilkan posisi yang diinginkan.



1. Elliptic Paraboloid

Terdapat tiga *elliptic paraboloid* yang berfungsi sebagai pendorong spaceship saat berpetualang ke luar angkasa. Ketiga *elliptic paraboloid* ini berfungsi sebagai sisi samping dari *ion thruster* sehingga tidak terlihat menempel di ujung spaceship tanpa penyangga. *Elliptic paraboloid* tersebut berwarna merah dan dapat dibuat dengan fungsi createEllipticParaboloid().

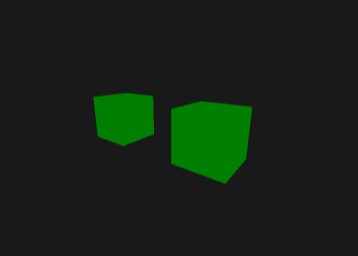


*Elliptic paraboloid* yang berikutnya berfungsi sebagai sisi belakang dari *ion thruster* supaya warna putih dari tembok belakang spaceship tidak terlihat. *Elliptic paraboloid* tersebut dapat dibuat dengan fungsi createEllipticParaboloid().



1. Balok

Objek dibawah ini digunakan sebagai koridor untuk menghubungkan sisi kiri, kanan, dan tengah dari Battle Spaceship. Posisi kedua koridor tersebut berada di antara tabung sisi masing - masing. Karena sebuah kubus adalah sebuah balok yang memiliki panjang ketiga rusuk yang sama, kubus ini dapat dibuat oleh fungsi createCuboid() dengan parameter panjang rusuk X, Y, dan Z yang sama. Penempatan kedua koridor tersebut adalah bagian dari sebuah for loop yang menempatkan empat pasangan objek lainnya, sehingga untuk memudahkan pembuatan objek, digunakan for loop, dan posisi kubus yang pertama akan dirotasi 180 derajat berpusat pada titik tengah spaceship hingga mendapatkan posisi yang diinginkan.



Objek dibawah ini terletak pada bagian atas badan utama Battle Spaceship untuk penempatan senjata. Kubus warna hitam ini juga dapat dibuat oleh fungsi createCuboid() dengan parameter panjang rusuk X, Y, dan Z yang sama.



Terdapat empat balok yang diletakkan di sisi kiri dan kanan sebagai bagian dari kaki Battle Spaceship sebagai penyambung kaki kerucut dan badan utama spaceship. Balok ini dapat dibuat oleh fungsi createCuboid(). Keempat balok tersebut di *translate* dan di *rotate* hingga mendapatkan posisi yang diinginkan.



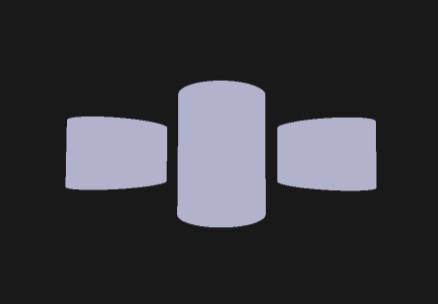
1. Kerucut

Terdapat empat kerucut yang berfungsi sebagai penyangga spaceship saat dalam keadaan mendarat. Keempat kerucut tersebut juga berfungsi sebagai *ion thruster* supaya spaceship memiliki alat pergerakan cadangan. Kerucut ini dapat dibuat oleh fungsi createEllipticCone().

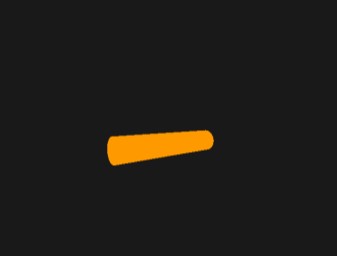


1. Tabung

Terdapat tiga tabung berwarna abu - abu yang berfungsi sebagai badan utama dari spaceship. Tabung - tabung tersebut dapat dibuat oleh fungsi createCylinder().



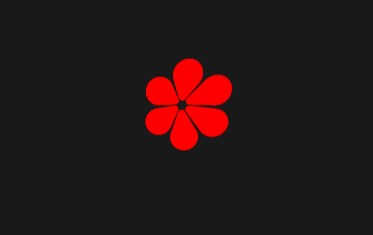
Terdapat sebuah tabung berwarna orange yang terletak pada bagian atas spaceship. Tabung tersebut dihubungkan dengan kubus berwarna hitam. Tabung ini juga dibuat oleh fungsi createCylinder() dan di *translate* dan di *rotate* hingga menjadi bentuk sebagai berikut.



Terdapat sebuah tabung yang warna hitam yang terletak di bagian bawah badan utama spaceship. Tabung tersebut berfungsi sebagai *lift* untuk mengakses spaceship. Tabung ini dibuat oleh fungsi createCylinder().



Terdapat enam tabung berwarna merah yang diletakkan pada tabung berwarna orange. Tabung - tabung tersebut berfungsi sebagai *barrel* dari senjata spaceship. Tabung - tabung tersebut dapat dibuat dengan fungsi createCylinder(). Penempatan keenam tabung dilakukan dengan for loop yang me-*rotate* sebuah tabung menurut titik tengah tabung berwarna orange.

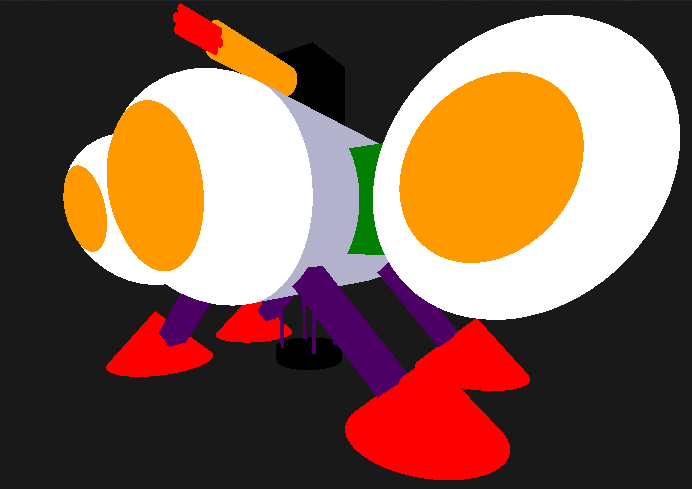
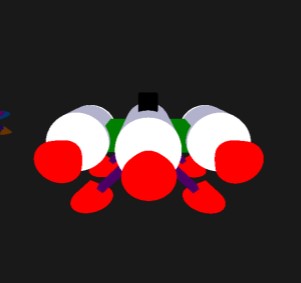


Terdapat empat tabung tipis berwarna ungu yang berfungsi sebagai rel untuk *lift*. Tabung - tabung tersebut dapat dibuat oleh fungsi createCylinder().



1. Bentuk Akhir

Bagian-bagian yang telah dipecah akan digabungkan sedemikian rupa dan ditambahkan transformasi *rotate* dan *translate* menjadi seperti gambar dibawah ini:



Tampak depan Tampak belakang

## *Myssilia Clan Emblem* (oleh: Denzel)



(Konsep Design)

*Emblem* ini terdiri dari objek - objek:

1. Prisma Trapezoid

Prisma Trapezoid dibuat dengan fungsi createTrapezoidPrism(), dimana fungsi tersebut adalah derivasi dari fungsi createBoxVertices(), yang ditambah dua parameter sudut. Kedua sudut tersebut akan menentukan penempatan verteks - verteks yang berada pada posisi negatif Y.

Prisma Trapezoid digunakan di berbagai tempat. Bagian - bagian yang berwarna hijau adalah bentuk - bentuk yang menggunakan Prisma Trapezoid.



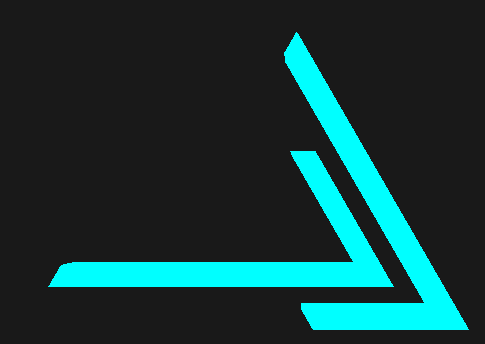
Bentuk pertama yang akan dibuat adalah salah satu sudut segitiga besar yang berada di bagian atas *emblem*. Dengan fungsi createTrapezoidPrism(), terbentuklah objek sebagai berikut.



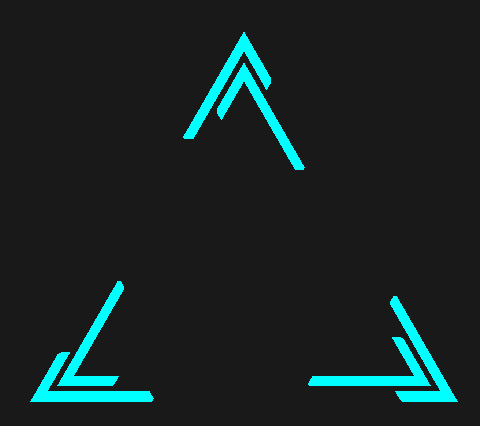
Untuk menggambar bentuk yang kedua, di buat bentuk yang mirip, lalu di *translate* dan di *rotate* sampai posisinya persis di sebelah kanan objek pertama.



Kemudian, kedua objek tersebut dicopy menggunakan *copy constructor (deep copy)*, dan hasilnya akan di *mirror*, di *translate*,dan di *rotate* hingga menjadi seperti berikut.



Akhirnya, semua objek yang telah dibuat akan di *copy* dua kali dan di *rotate* hingga berada di tempat yang benar, sehingga membuat segitiga besar.

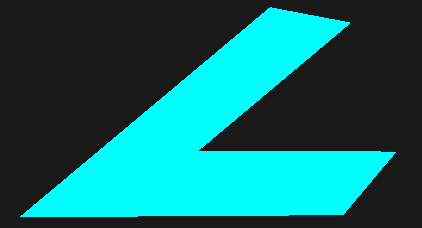


Bentuk *snowflake* di bagian tengah *emblem* terdiri dari 24 Prisma Trapezoid. Daun dari *snowflake* tersebut adalah bentuk layang - layang yang memiliki sudut - sudut 60 - 130 - 40 - 130 derajat.

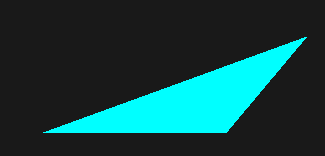
Untuk membuat bagian luar dari daun, di buat prisma trapezoid seperti berikut.



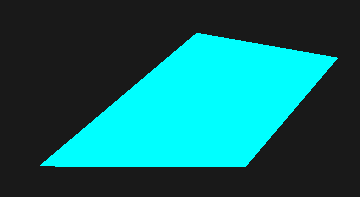
Prisma trapezoid ini akan di *copy*, di *mirror*, dan di *rotate* hingga menjadi seperti berikut.



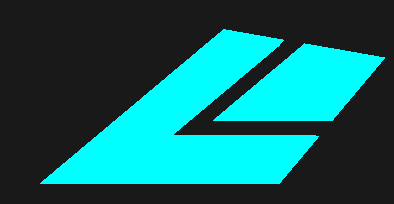
Untuk bagian dalam daun, di buat prisma trapezoid dengan panjang sumbu X = 0.



Seperti sebelumnya, prisma trapezoid ini akan di *copy*, di *mirror*, dan di *rotate*.



Saat digabung, keempat objek tersebut akan menjadi seperti berikut.



Sebelum objek - objek tersebut di *copy* lima kali, objek - objek tersebut akan di *translate* sehingga bagian dalam daun memiliki jarak tertentu dari titik (0, 0, 0).

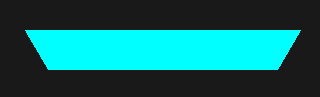
Setelah di *translate*, objek - objek tersebut akan di *copy* lima kali untuk menghasilkan bentuk sebagai berikut.



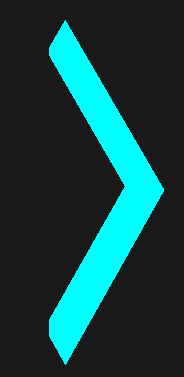
*Emblem* bagian bawah terdiri dari 20 Prisma Trapezoid.



Prisma Trapezoid no. 1 adalah sebagai berikut.



Objek tersebut akan di *translate*, di *rotate*, dan di *copy*. *Copy* dari objek tersebut akan di *mirror* pada sumbu Y untuk menghasilkan bentuk sebagai berikut.



Kemudian, kedua - dua objek tersebut akan di *copy* dan di *mirror* pada sumbu X untuk menghasilkan bentuk sebagai berikut.



Demikian untuk objek no. 1 - 4.

Dengan proses yang mirip seperti objek no. 1 - 4, objek no. 5 adalah sebagai berikut.



Objek no. 6 adalah *copy* dari no. 5. Sebelum di *copy*, objek no. 5 di *translate*, di *rotate*, dan di *translate* lagi. Setelah di *copy,* objek no. 6 di *mirror* pada sumbu X.

Objekno. 7 dan 8 adalah *copy* dari no. 5 dan 6. Setelah di *copy*, objek - objek tersebut akan di *mirror*.

Berikut adalah hasil dari objek no. 5 - 8.



Untuk mengurangi redundansi, berikut hasil dari objek no. 1 - 16. Objek no. 13 - 16 adalah *copy* dari objek no. 9 - 12 yang di *translate* sesuai dengan gambar.



Berikut adalah objek no. 1 - 20. objek no. 18 adalah *mirror* dari objek no. 17, dan objek no. 19 dan 20 adalah *mirror* dari objek no. 17 dan 18.



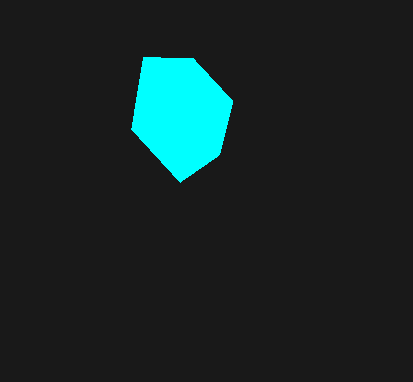
1. Prisma Belah Ketupat

Prisma Belah Ketupat dibuat dengan fungsi createRhombicPrism(). Fungsi tersebut adalah derivasi dari fungsi createBoxVertices(), dimana terdapat perbedaan dalam penempatan verteks.

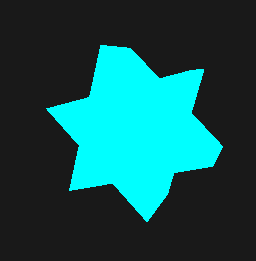
Prisma Belah Ketupat ini digunakan di berbagai tempat. Bagian - bagian yang berwarna hijau adalah bentuk - bentuk yang menggunakan Prisma Belah Ketupat.



Bintang enam yang berada di tengah *emblem* terdiri dari enam prisma belah ketupat. Berikut adalah salah satu prisma yang dibuat oleh fungsi createRhombicPrism().



Untuk membuat bintang enam, prisma ini akan di copy lima kali dan di rotate hingga membentuk bentuk yang sesuai.





Prisma Belah Ketupat no. 1 tidak memiliki transformasi apapun; saat objek tersebut dibuat, penempatan objek sudah dilakukan oleh fungsi createRhombicPrism().

Prisma Belah Ketupat no. 2 adalah *copy* dari no. 1. Setelah di *copy*, objek tersebut di *scale* dan di *translate*.

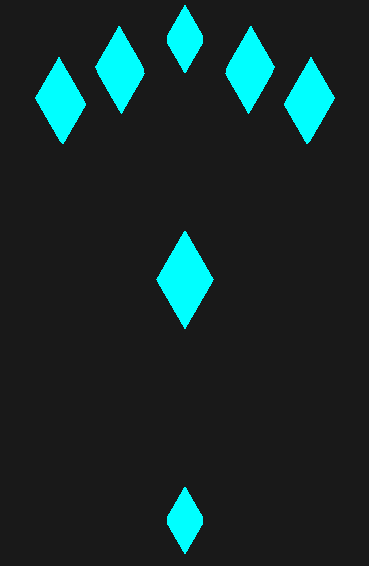
Prisma Belah Ketupat no. 3 adalah *copy* dari no. 2 dan memiliki transformasi *translate.*

Prisma Belah Ketupat no. 4 dan 5 adalah *copy* dari no. 2 dan 3 dan memiliki transformasi *mirror*.

Prisma Belah Ketupat no. 6 adalah *copy* dari no. 2. Setelah di *copy*, objek tersebut di *scale* dan di *translate*.

Prisma Belah Ketupat no. 7 adalah *copy* dari no. 1. Untuk mendapatkan koordinat yang tepat sesuai dengan gambar original, setelah di *copy*, objek tersebut di *mirror* pada sumbu Y yang berpusat di titik tengah objek no. 6.

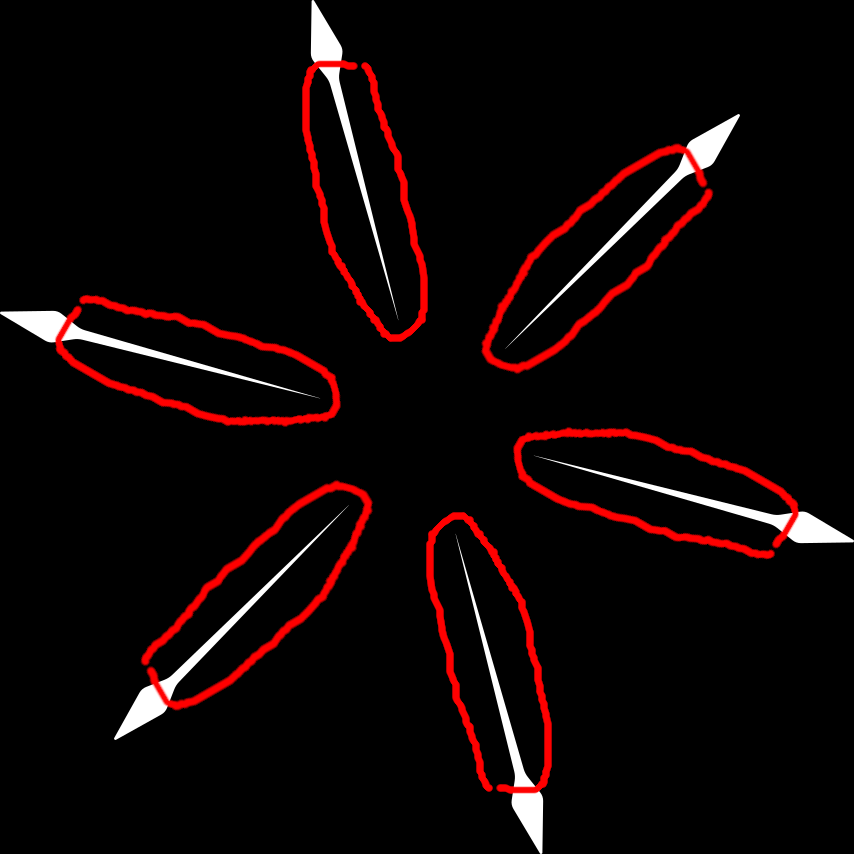
Berikut adalah gambar dari hasil objek - objek di atas.



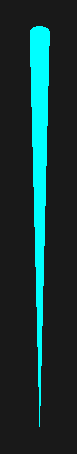
1. Kerucut

Kerucut dibuat dengan fungsi createEllipticCone().

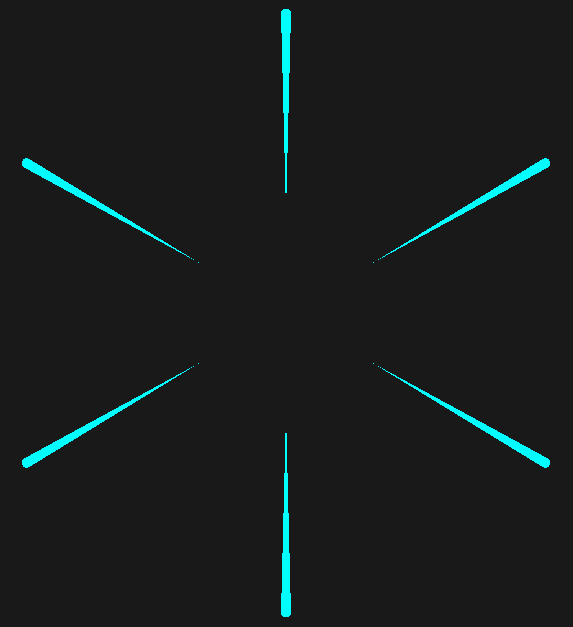
Bentuk Kerucut digunakan untuk menggambar bagian tangkai dari snowflake di bagian atas *emblem*.



Dengan fungsi createEllipticCone(), terbentuklah bentuk kerucut sebagai berikut.



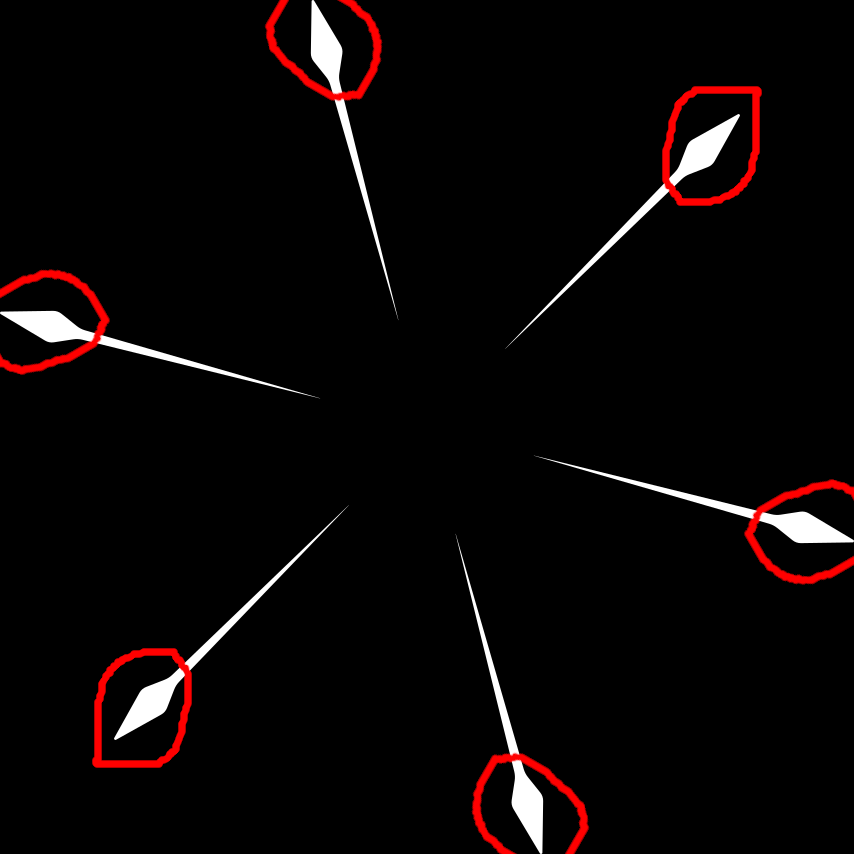
Bentuk kerucut tersebut akan di *copy* dan di *rotate* lima kali untuk menghasilkan bentuk sebagai berikut.



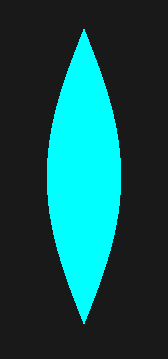
1. Kapsul Sinusoidal

Kapsul Sinusoidal adalah bentuk yang terbuat dari fungsi createSinusoidCapsule(). Fungsi tersebut adalah derivasi dari fungsi createEllipsoid(), dimana penempatan verteks yang melalui sumbu kutub berubah dari Sin menjadi linear, sehingga saat verteks ditempatkan oleh fungsi, terlihat jelas bentuk kurva Sin pada hasil akhir.

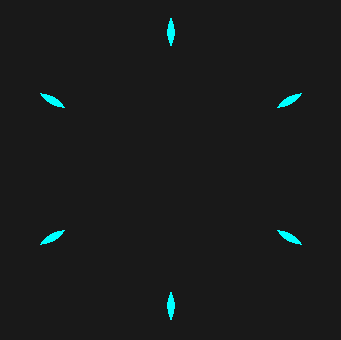
Bentuk Kapsul Sinusoidal digunakan untuk menggambar bagian kepala dari snowflake di bagian atas *emblem*.



Dengan fungsi createSinusoidCapsule(), di buat bentuk sebagai berikut.



Bentuk tersebut akan di *copy* dan di *rotate* lima kali untuk menghasilkan bentuk sebagai berikut.



1. Bentuk Akhir



## UFO (oleh: Andreas)

Ufo ini terdiri dari bagian objek:

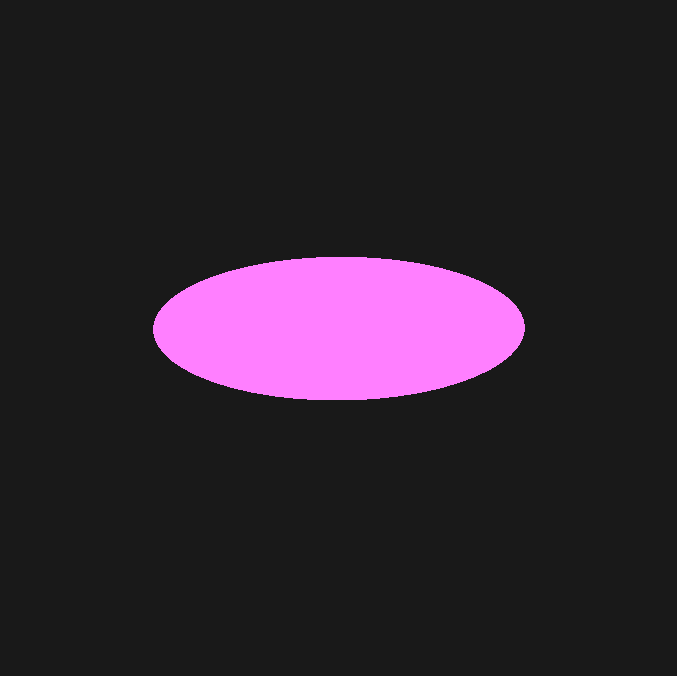
1. Setengah Bola

Objek ini digunakan sebagai ruangan kemudi UFO yang terletak di atas lingkaran warna pink. Setengah bola ini dapat dibuat oleh fungsi createFractionedEllipsoid(), dimana nilai parameter *fraction* adalah 0.5 sehingga bentuk bola akan digambar sampai setengah dari ukuran penuh. *Optional argument* startPoint berisi nilai 0, sehingga bola akan mulai digambar dari titik paling atas.



1. Lingkaran

Lingkaran ini digunakan sebagai penghubung antara setengah bola dan tabung yang warna biru di no. 1. Lingkaran ini dibuat oleh fungsi createCircle().

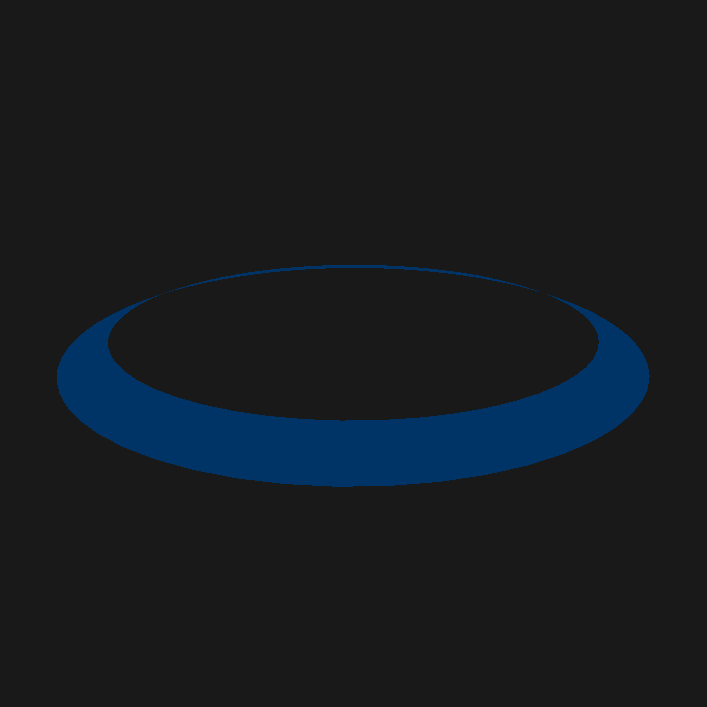


1. Tabung

Beberapa bentuk tabung yang disusun dengan cara ditumpuk ini merupakan badan ufo.

1. Tabung no. 1

Tabung ini digunakan sebagai penghubung antara lingkaran dan tabung no. 2. Tabung ini dapat dibuat oleh fungsi createVariedCylinder(), dimana terdapat dua parameter untuk menentukan radius dari sisi atas dan sisi bawah dari tabung. Terdapat juga dua parameter untuk menentukan bila sisi atas dan sisi bawah akan digambar atau tidak. Untuk tabung no. 1, sisi atas dan sisi bawah tidak digambar.



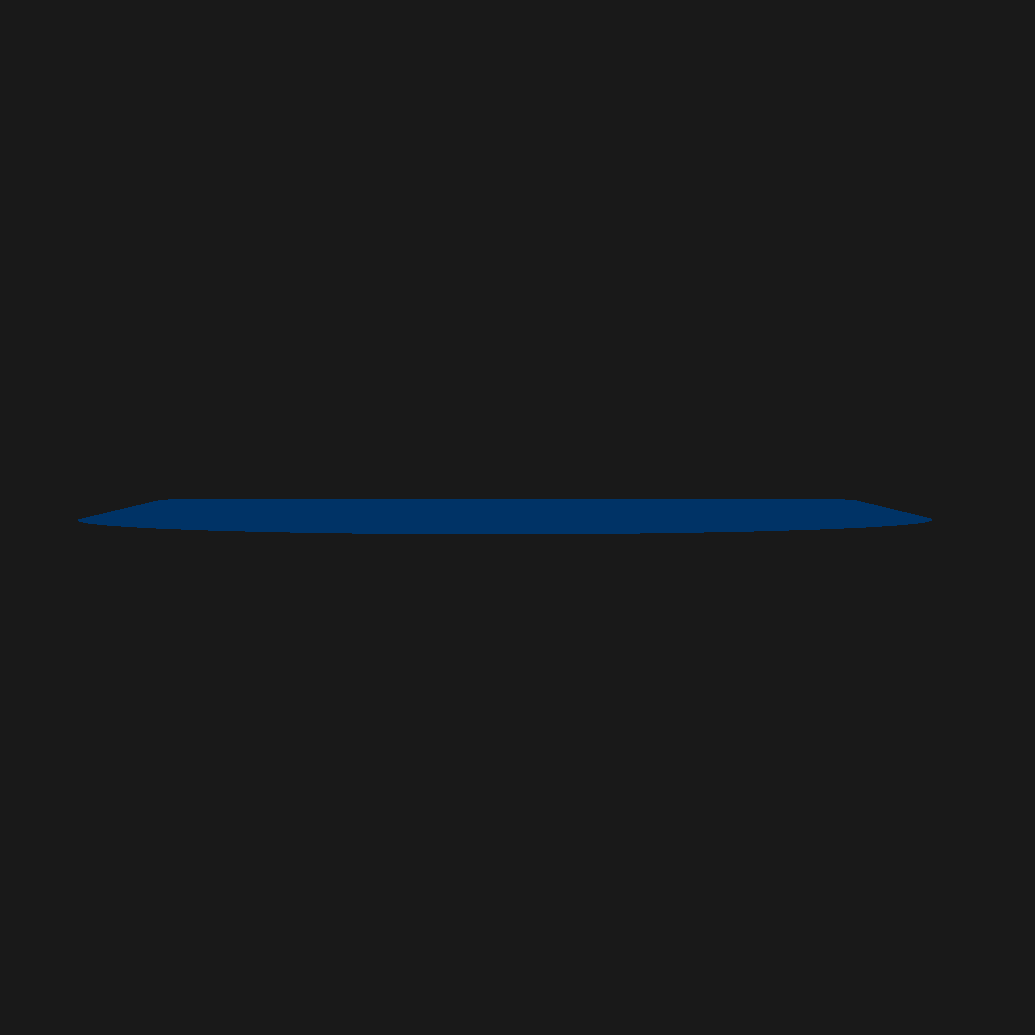
1. Tabung no. 2

Tabung ini digunakan sebagai penghubung antara tabung no. 1 dan tabung no. 3. Tabung ini juga dapat dibuat oleh fungsi createVariedCylinder(). Tabung no. 2 juga tidak memiliki sisi atas dan bawah.



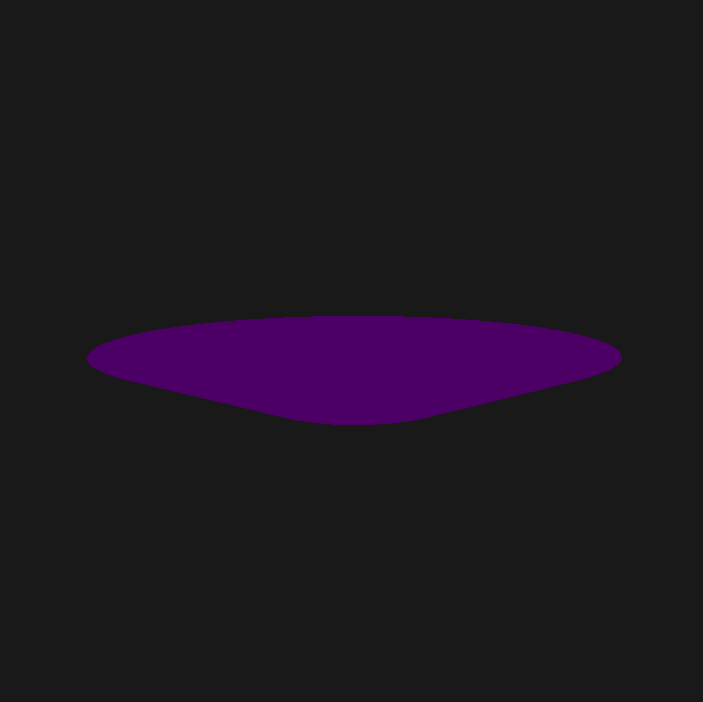
1. Tabung no. 3

Tabung ini digunakan sebagai penghubung antara tabung no. 2 dan tabung no. 4. Tabung ini juga dapat dibuat oleh fungsi createVariedCylinder(). Tabung no. 3 tidak memiliki sisi atas, tetapi memiliki sisi bawah.



1. Tabung no. 4

Tabung ini digunakan sebagai penghubung antara tabung no. 3 dan ketiga balok. Tabung ini juga dapat dibuat oleh fungsi createVariedCylinder(). Tabung no. 4 tidak memiliki sisi atas, tetapi memiliki sisi bawah, dan juga memiliki radius sisi atas yang lebih besar daripada radius sisi bawah.

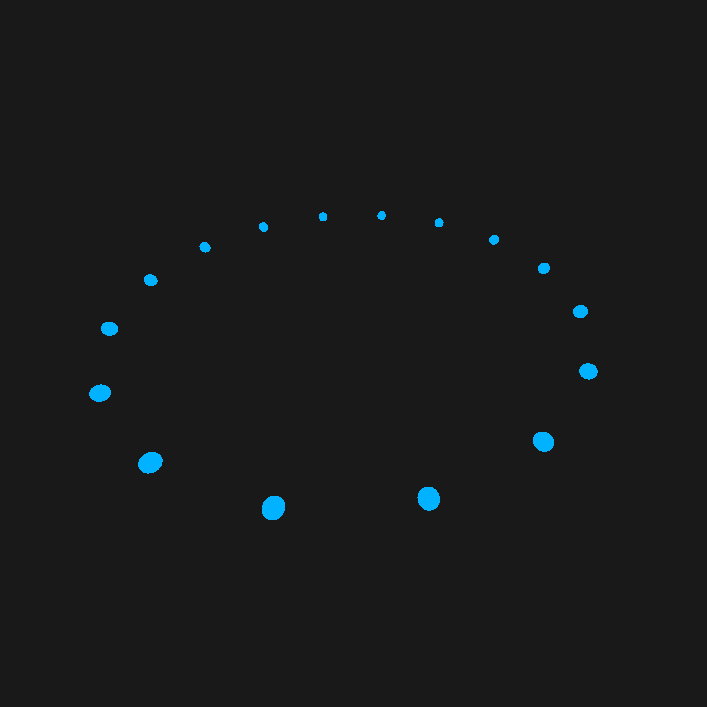


1. Bola

Terdapat 12 bola yang dibuat oleh fungsi createEllipsoid(). Fungsi tersebut dipanggil 12 kali dengan for loop dan di *rotate* sebelum dimasukkan ke dalam kumpulan objek. Bola - bola tersebut terletak pada samping tabung no. 2.

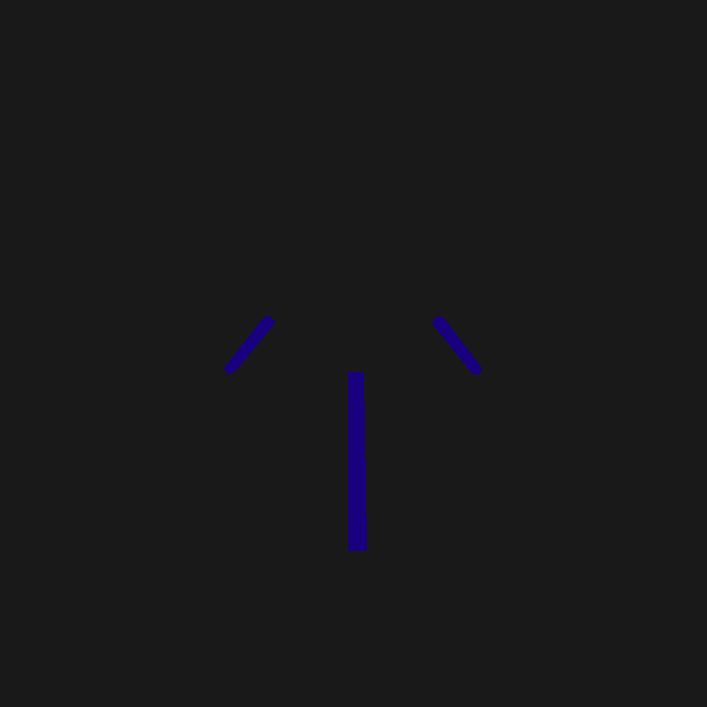


Terdapat 16 bola yang dibuat oleh fungsi createEllipsoid(). Fungsi tersebut dipanggil 16 kali dengan for loop dan di *rotate* sebelum dimasukkan ke dalam kumpulan objek. Bola - bola tersebut terletak pada sisi bawah tabung no. 3 yang digunakan sebagai lampu sensor UFO saat ingin terbang.



1. Balok

Terdapat 3 buah balok yang berfungsi sebagai penyangga UFO saat sedang mendarat. Balok - balok tersebut dibuat dengan fungsi createCuboid(). createCuboid() adalah derivasi dari fungsi createBoxVertices(), dimana terdapat tiga parameter untuk menentukan ukuran balok. Fungsi createCuboid() dipanggil tiga kali dalam sebuah for loop, dan balok - balok tersebut di *translate* dan di *rotate* hingga menghasilkan posisi dan putaran yang diinginkan.

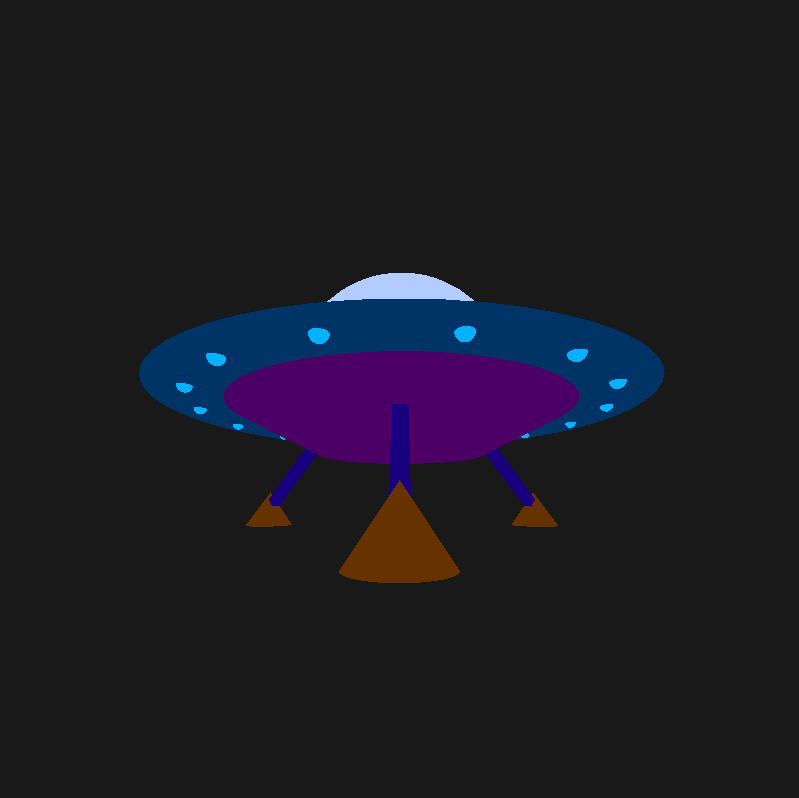
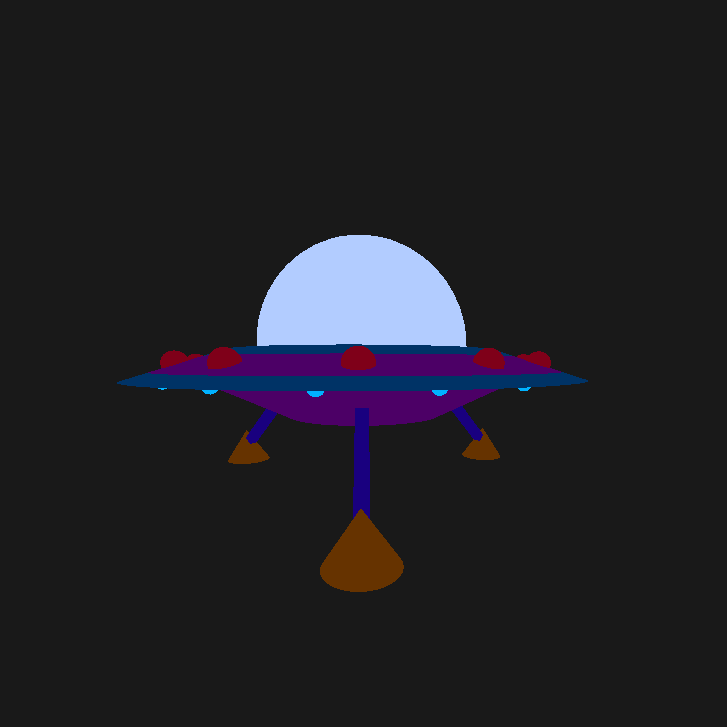


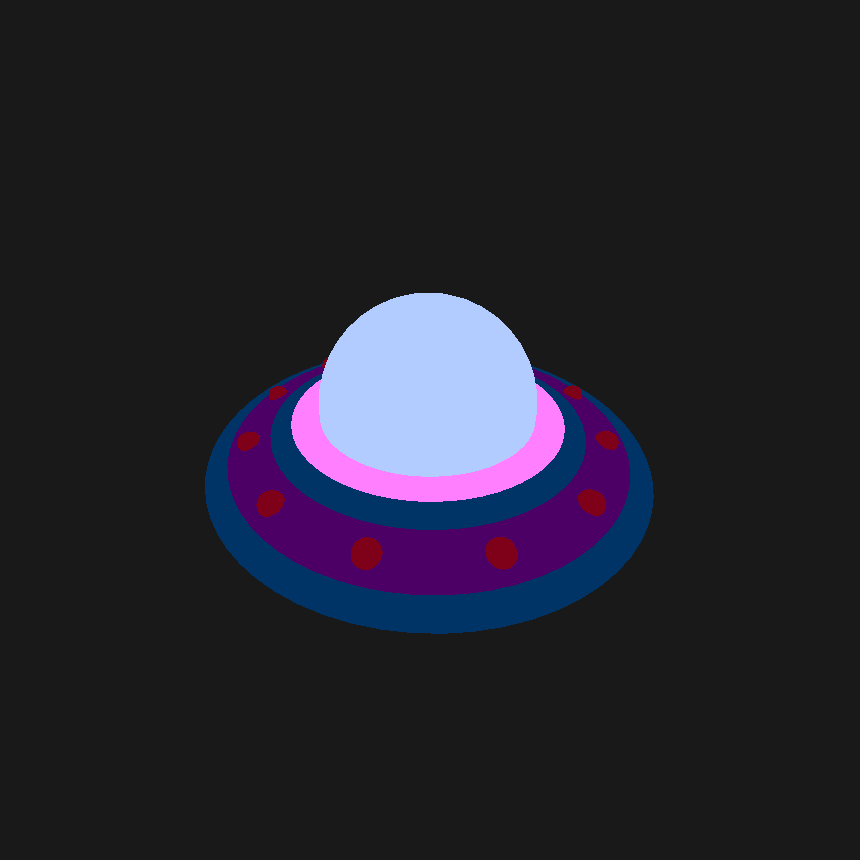
1. Kerucut

Terdapat 3 buah kerucut yang berfungsi sebagai pijakan UFO saat sedang mendarat. Kerucut - kerucut tersebut dibuat dengan fungsi createEllipticCone(). Fungsi createEllipticCone() dipanggil tiga kali dalam sebuah for loop, dan kerucut - kerucut tersebut di *translate* dan di *rotate* hingga menghasilkan posisi dan putaran yang diinginkan.



1. Bentuk Akhir

Bagian-bagian yang telah dipecah akan digabungkan sedemikian rupa dan ditambahkan transformasi *rotate* dan *translate* menjadi seperti gambar dibawah ini:



# Animasi Objek

Kelompok kami memutuskan untuk memetakan animasi objek - objek kami menurut sebuah musik. Musik yang kami pilih adalah *Au5 - Crystal Mathematics (Instrumental)*. Musik ini memiliki BPM yaitu 145,5.

Animasi objek - objek kami dipetakan sebagai berikut:



Setiap kotak menandakan delapan ketukan.

Battle Spaceship milik Wilson menggunakan animasi berupa penerbangan spaceship ke arah yang telah ditentukan. Tingkat intensitas dari kecepatan animasi tergantung dari suasana musik pada saat detik tersebut (Saat terjadi pergerakan musik yang intensif, spaceship akan bergerak lebih cepat). Semua transformasi (*rotation, scaling,* dan *translation*) digunakan untuk mencapai hasil yang diinginkan.

Terdapat empat kelompok objek, dimana tiga kelompok yang tertulis di bawah memiliki animasi khusus:



Terdapat tiga animasi yang lepas dari pergerakan badan utama dari Battle Spaceship, yaitu gerakan naik turun lift pada tabung berwarna hitam dan empat tabung berwarna ungu, gerakan pergeseran naik pada keempat kaki (balok dan kerucut) serta gerakan senjata yang berada di atas Battle Spaceship. Pertama, proses gerakan naik lift ini terjadi pada saat sebelum Battle Spaceship mulai bergerak dan sebelum animasi berhenti. Objek - objek tersebut akan bergerak naik turun sesuai dengan iringan musik. Kedua, keempat kaki (balok dan kerucut) akan bergerak sedikit kebelakang saat Battle Spaceship bergerak maju. Proses geraknya akan mengubah posisi kaki balok dan kerucut ke geser naik atas sedikit mungkin. Ketiga, saat Battle Spaceship bergerak pada beberapa saat tertentu, senjata di bagian atas Battle Spaceship akan berputar.

*Emblem* milik Denzel menggunakan animasi berupa perputaran dan pergeseran bagian - bagian dari *emblem* secara individu. Saat musik memasuki bagian *chorus*, animasi akan menjadi sangat agresif, dimana berbagai bagian dari *emblem* akan terbang ke segala arah. Semua transformasi (*rotation, scaling,* dan *translation*) digunakan untuk mencapai hasil yang diinginkan.

Terdapat 14 kelompok objek, yang diurutkan sebagai berikut:



Semua objek yang akan diputar memiliki *rotationId* yang unik, sehingga tingkat kontrol terhadap animasi yang akan dilakukan cukup tinggi. Sekelompok objek yang bergeser menjauh dari titik pusat asli mereka tetap dapat memutar secara benar dengan adanya tambahan objek kosong yang menandakan titik putar mereka. Titik putar ini memiliki *rotationId* yang sama, sehingga tidak dimungkinkan untuk susunan sekelompok objek menjadi hancur, selama objek - objek tersebut diputar dengan titik putar yang benar.

UFO milik Andreas menggunakan animasi berupa penerbangan badan UFO ke arah yang telah ditentukan. Tingkat intensitas dari kecepatan animasi tergantung dari suasana musik pada saat detik tersebut. Semua transformasi (*rotation, scaling,* dan *translation*) digunakan untuk mencapai hasil yang diinginkan.

Terdapat lima kelompok objek, dimana empat kelompok yang tertulis di bawah memiliki animasi khusus:



Terdapat tiga animasi yang lepas dari pergerakan badan utama dari UFO, yaitu perputaran lampu di sisi atas UFO, perputaran lampu di sisi bawah UFO, dan keluar masuk kaki UFO. Perputaran kedua kelompok lampu tersebut berjalan secara konstan selama animasi berjalan, dan kecepatan lampu berputar juga dapat diubah saat animasi berjalan, sedangkan keluar masuk kaki UFO akan dijalankan di beberapa waktu saat animasi berjalan.

# Penutupan

Demikian laporan yang kami buat, kami berharap bahwa laporan ini telah menjelaskan proyek kami dengan jelas. Kami mohon maaf apabila ada kesalahan kata dan ejaan yang kurang tepat. Atas waktunya kami ucapkan terima kasih sebesar-besarnya.