

LAPORAN TUGAS
Untuk Memenuhi Nilai Ujian Tengah Semester
Mata Kuliah Grafika Komputer Semester Genap Tahun Ajaran 2022



Disusun oleh:

Ivan Chandra C14200119

Ivan Vincent Kwenandar C14200138

Sebastian Jonathan C14200157

Dosen:

ANDHIKA EVANTIA IRAWAN, S.Kom.

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

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS KRISTEN PETRA
SURABAYA**

DAFTAR ISI

DAFTAR ISI	i
PENDAHULUAN	ii
HASIL AKHIR	iii
DESAIN	6
ANIMASI	31
PENUTUP	45

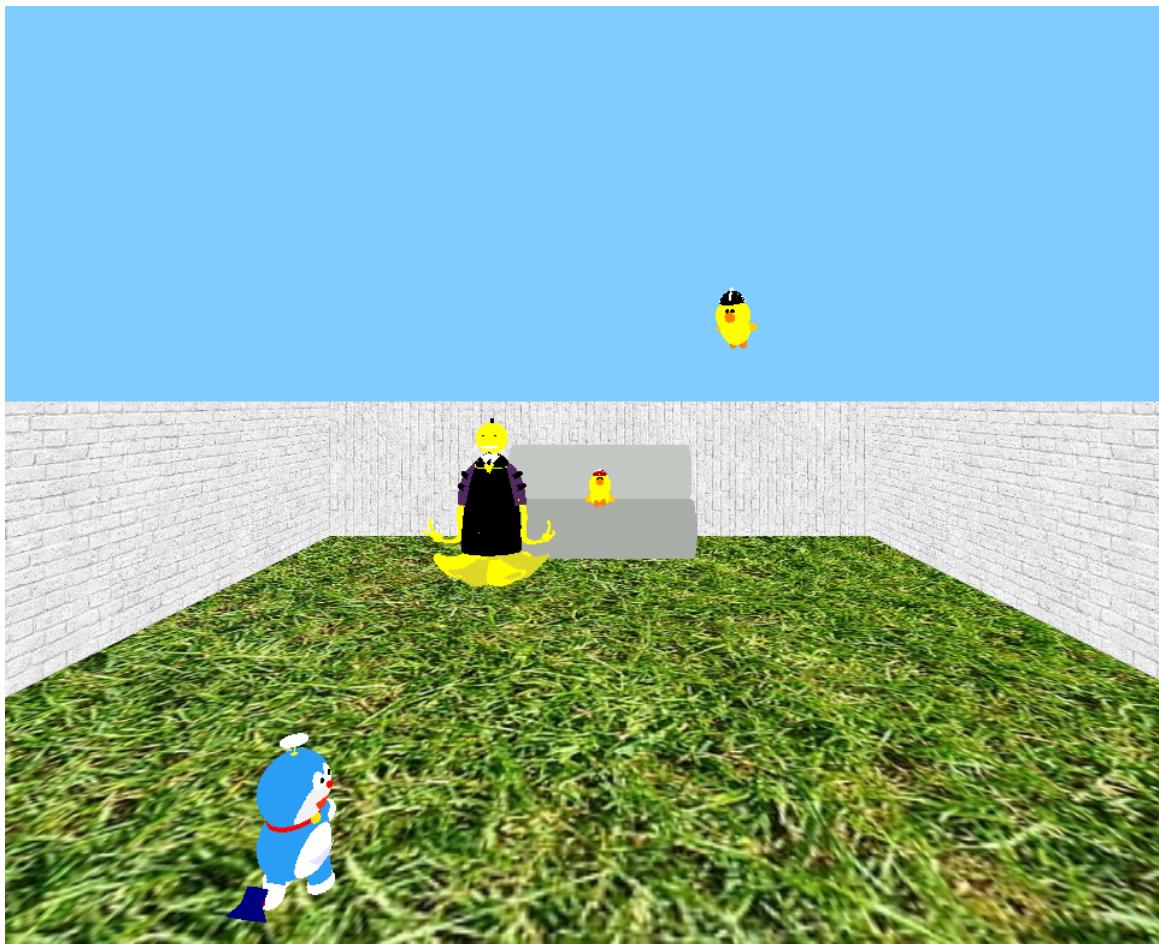
PENDAHULUAN

Pada Ujian Tengah Semester (UTS) ini kami mendapatkan proyek dengan tujuan utama untuk membuat sebuah sket berisi objek 3 (tiga) dimensi (3D) mengenai suatu karakter fantasi (objek kompleks) yang menggunakan gabungan dari objek-objek sederhana, kurva, dan minimal 3 jenis *parametric quadric* (atau objek 3D lainnya yang di *generate* menggunakan rumus) dengan melakukan modifikasi pada rumus *modeling* objek 3D sedemikian rupa sehingga didapatkan bentuk yang diinginkan. Kemudian lakukan implementasi dari sket tersebut menggunakan bahasa pemrograman ($C\# \rightarrow C\text{-Sharp}$) dan penggabungan tiap komponen dari masing-masing objek 3D dengan menggunakan transformasi untuk mengatur ukuran, kemiringan, rotasi, dan sebagainya.

Proyek ini kami buat menggunakan *Visual Studio* 2022 dan memanfaatkan penggunaan *library* OpenTK yang merupakan kumpulan *binding* C# tingkat rendah yang cepat, portabel, untuk OpenGL, OpenGL ES, OpenAL, dan OpenCL. OpenTK dapat digunakan di semua *platform* utama dan mendukung ratusan aplikasi, game, dan program penelitian ilmiah. Tema dari proyek yang kami buat adalah **“Doraemon and Koro Crossover sponsored by LINE”** dimana sesuai dengan tema tersebut kami membuat 3 macam karakter yang mewakili tiap komponen dari tema itu sendiri yaitu **Koro, Doraemon, dan Sally (LINE)**.

HASIL AKHIR

Kami berhasil membuat desain objek 3D dan melakukan implementasi serta penggabungan objek secara individu dimana dilakukan penyesuaian posisi objek dan gerakannya. Kemudian, semua objek milik kami bertiga digabungkan ke dalam 1 program untuk menunjukan pergerakan dari masing-masing karakter yang sudah kami buat.



Berikut daftar karakter yang kami buat:

1. Sally (Ivan Chandra):

Karakter ini merupakan referensi karakter *Sally* dari aplikasi Line. karakter ini merupakan bebek yang didesain dalam kartun dan dijadikan sebagai maskot aplikasi LINE dimana ada pengambilan referensi dari internet dan penggambaran sket dalam pembuatan karakter 3D ini.



2. Koro-Sensei (Ivan Vincent Kwenandar):

Karakter terinspirasi dari sebuah animasi Jepang berjudul ‘*Assassination Classroom*’ (*Ansatsu Kyōshitsu* / 暗殺教室). Nama karakternya adalah ‘*Koro-Sensei*’ yang dimana merupakan manusia yang bermutasi menjadi sebuah makhluk berwarna kuning yang menyerupai Gurita akibat dari eksperimen militer.

Untuk rancangan desainnya saya membuat gambaran awal karakter pada secerik kertas terlebih dahulu untuk mempermudah proses pembuatan. Karena bentuk karakter yang sedikit rumit, ada beberapa bagian yang saya minimalkan dan kurangi demi mempermudah proses pengerjaan tanpa mengurangi esensi dari karakter itu sendiri.



3. Doraemon (Sebastian Jonathan):

Karakter ini merupakan terinspirasi dari serial TV *Doraemon*. Salah satu hal populer dalam Doraemon adalah terbang dengan baling-baling bambu. Maka dari itu, saya memberi Doraemon baling-baling bambu serta memberi animasi terbang (*fly*). Dalam proses desain, saya ingin Doraemon memegang sesuatu, maka dari itu saya



memberi Doraemon sebuah sirine/megaphone. Karakter ini akan berjalan dan terbang.

Untuk desain *environment* dari karakter kami memanfaatkan penggunaan *Texture* yang dimana kami gunakan untuk melakukan penambahan efek warna yang didasarkan pada file foto (.jpeg, .png, .jpg) yang di panggil. *Texture* diaplikasikan pada bidang *cuboid* yang masing-masing diposisikan agar dapat membentuk lantai dan tembok.

DESAIN

Ivan Chandra

Objek Dasar 3D	
Ellipsoid CreateEllipsoid()	<p>Input: (x, y, z) = Titik pusat objek dan pivot awal radiusX = jari-jari sumbu X radiusY = jari-jari sumbu Y radiusZ = jari-jari sumbu Z</p> <p>Persamaan: $X = \text{radiusX} * \cos(v) * \cos(u)$ $Y = \text{radiusY} * \cos(v) * \sin(u)$ $Z = \text{radiusZ} * \sin(v)$</p> <p>Range: - $\pi < u < \pi$ - $\pi/2 < v < \pi/2$</p>
Half Ellipsoid (Modifikasi)	<p>Input: (x, y, z) = Titik pusat objek dan pivot awal radiusX = jari-jari sumbu X radiusY = jari-jari sumbu Y radiusZ = jari-jari sumbu Z</p> <p>Persamaan: $X = \text{radiusX} * \cos(v) * \cos(u)$ $Y = \text{radiusY} * \cos(v) * \sin(u)$ $Z = \text{radiusZ} * \sin(v)$</p> <p>Range: - $\pi < u < \pi/314$ - $\pi/2 < v < \pi/2$</p>
Cylinder	<p>Input: (x,y,z) = Titik pusat objek dan pivot awal radius = Radius Cylinder Height = tinggi cylinder</p> <p>Persamaan: $X = \text{radius} * \cos(u)$ $Y = v$ $Z = \text{radius} * \sin(u)$</p>

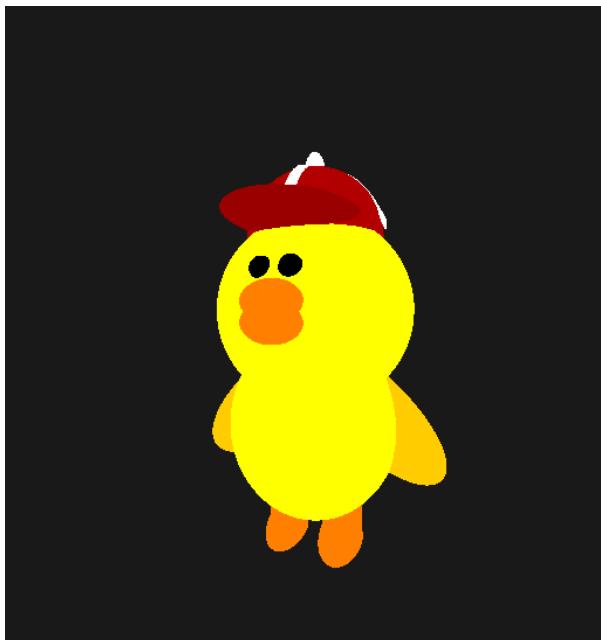
	<p>Range: -height < v < height - π < u < π</p>
$\frac{1}{3}$ Ellipsoid(Modifikasi)	<p>Input: (x, y, z) = Titik pusat objek dan pivot awal radiusX = jari-jari sumbu X radiusY = jari-jari sumbu Y radiusZ = jari-jari sumbu Z</p> <p>Persamaan: $X = \text{radiusX} * \cos(v) * \cos(u)$ $Y = \text{radiusY} * \cos(v) * \sin(u)$ $Z = \text{radiusZ} * \sin(v)$</p> <p>Range: - π < u < $\pi/2$ - $\pi/2$ < v < $\pi/2$</p>
Paraboloid	<p>Input: (x, y, z) = Titik pusat objek dan pivot awal radiusX = jari-jari sumbu X radiusY = jari-jari sumbu Y radiusZ = jari-jari sumbu Z</p> <p>Persamaan: $X = \text{radiusX} * \cos(v) * \cos(u)$ $Y = \text{radiusY} * \cos(v) * \sin(u)$ $Z = \text{radiusZ} * 2 * \sin(v)$</p> <p>Range: - π < u < π 0 < v < $\pi/2$</p>
BezierCylinder	<p>Input: (x, y, z) = Titik pusat objek dan pivot awal Radius = radius Cylinder Panjang = panjang bezier curve ext = posisi titik ke-2 pada X di kurva</p> <p>Persamaan: $X = \text{kurva.X} + \cos(u)$ $Y = \text{kurva.Y} + \sin(u)$ $Z = v$</p> <p>Range:</p>

	$-\pi < u < \pi$ -panjang / 2 < v < panjang / 2
--	--

Desain Model Karakter



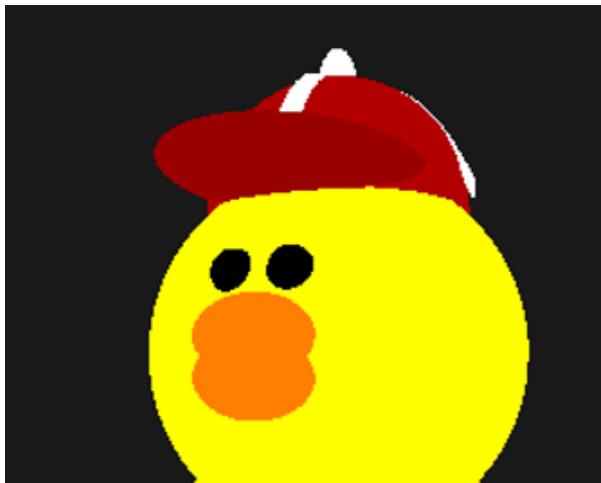
Rancangan Desain



Hasil Akhir

Dengan hasil rancangan berhasil dibuatnya 3d objek Sally ini yang menggunakan objek quadric dan kurva.

Penempatan tiap bagian 3D objek ini menggunakan metode translasi, dan rotasi



Kepala

Untuk Kepala, mata, dan mulut menggunakan objek solid *Ellipsoid*

Adapun diaturnya radius Y pada mulut sehingga sesuai dengan bentuk mulut.

	<p>Sayap Sayap menggunakan objek solid <i>Half Ellipsoid</i> yang merupakan hasil modifikasi <i>Ellipsoid</i> dan kemudian di scale dan rotate sesuai posisi sayapnya</p>
	<p>Kaki Pembuatan objek kaki merupakan gabungan dari objek solid <i>Cylinder</i> dan <i>1/3Ellipsoid</i></p>
	<p>Topi Pembuatan objek Topi merupakan gabungan dari objek solid <i>HalfCylinder</i> dan <i>Paraboloid</i> dan <i>BezierCurveCylinder</i> Kurva bezier ini menggunakan 3 titik dimana terletak pada ujung atas/bawah, beserta pada tengah kurva</p>

Objek Dasar 3D	
Ellipsoid createEllipsoid()	<p>Input: (x, y, z) = Titik pusat objek radiusX = jari-jari sumbu X radiusY = jari-jari sumbu Y radiusZ = jari-jari sumbu Z sectorCount = jumlah garis bujur stackCount = jumlah garis lintang</p> <p>Persamaan: $X = \text{radiusX} * \cos(v) * \cos(u)$ $Y = \text{radiusY} * \cos(v) * \sin(u)$ $Z = \text{radiusZ} * \sin(v)$</p>
Ellipsoid (versi <i>Collar</i>) createCollarTop()	<p>Input: (x, y, z) = Titik pusat objek radiusX = jari-jari sumbu X radiusY = jari-jari sumbu Y radiusZ = jari-jari sumbu Z sectorCount = jumlah garis bujur stackCount = jumlah garis lintang</p> <p>Persamaan: $X = \text{radiusX} * \cos(v) * \cos(u)$ $Y = \text{radiusY} * \cos(v) * \sin(u)$ $Z = \text{radiusZ} * \sin(v)$</p> <p>Untuk rumus sama dengan fungsi createEllipsoid(), namun yang berubah adalah nilai sectorCount dan stackCount masing-masing dikalikan dengan (-0.5f).</p>
Cylinder createCap()	<p>Input: (x, y, z) = Titik pusat objek radius = jari-jari</p> <p>Persamaan: Untuk nilai i mulai dari 0 sampai 359, dimana degInRad nya adalah PI dikali i lalu dibagi 180.</p> $X = \text{radius} * \cos(\text{degInRad}) + _X;$ $Y = Y;$ $Z = \text{radius} * \sin(\text{degInRad}) + _Z;$

Cuboid createCuboid() createCapTop()	<p>Input: (x, y, z) = Titik pusat objek length = Panjang ruas kubus</p> <p>Persamaan:</p> <p>Titik 1 = $(x - \text{length}/2, y + \text{length}/2, z - \text{length}/2)$ Titik 2 = $(x + \text{length}/2, y + \text{length}/2, z - \text{length}/2)$ Titik 3 = $(x - \text{length}/2, y - \text{length}/2, z - \text{length}/2)$ Titik 4 = $(x + \text{length}/2, y - \text{length}/2, z - \text{length}/2)$ Titik 5 = $(x - \text{length}/2, y + \text{length}/2, z + \text{length}/2)$ Titik 6 = $(x + \text{length}/2, y + \text{length}/2, z + \text{length}/2)$ Titik 7 = $(x - \text{length}/2, y - \text{length}/2, z + \text{length}/2)$ Titik 8 = $(x + \text{length}/2, y - \text{length}/2, z + \text{length}/2)$</p>
Cuboid (versi Kerah luar) createTieR() createTieL()	<p>Input: (x, y, z) = Titik pusat objek length = Panjang ruas kubus</p> <p>Persamaan (createTieR()):</p> <p>Titik 1 = $(x - \text{length}/3, y + \text{length}/3, z - \text{length}/3)$ Titik 2 = $(x + \text{length}/2, y + \text{length}/2, z - \text{length}/2)$ Titik 3 = $(x - \text{length}/2, y - \text{length}/2, z - \text{length}/2)$ Titik 4 = $(x + \text{length}/2, y - \text{length}/2, z - \text{length}/2)$ Titik 5 = $(x - \text{length}/2, y + \text{length}/2, z + \text{length}/2)$ Titik 6 = $(x + \text{length}/2, y + \text{length}/2, z + \text{length}/2)$ Titik 7 = $(x - \text{length}/2, y - \text{length}/2, z + \text{length}/2)$ Titik 8 = $(x + \text{length}/2, y - \text{length}/2, z + \text{length}/2)$</p> <p>Persamaan (createTieL()):</p> <p>Titik 1 = $(x - \text{length}/2, y + \text{length}/2, z - \text{length}/2)$ Titik 2 = $(x + \text{length}/3, y + \text{length}/3, z - \text{length}/3)$ Titik 3 = $(x - \text{length}/2, y - \text{length}/2, z - \text{length}/2)$ Titik 4 = $(x + \text{length}/2, y - \text{length}/2, z - \text{length}/2)$ Titik 5 = $(x - \text{length}/2, y + \text{length}/2, z + \text{length}/2)$ Titik 6 = $(x + \text{length}/2, y + \text{length}/2, z + \text{length}/2)$ Titik 7 = $(x - \text{length}/2, y - \text{length}/2, z + \text{length}/2)$ Titik 8 = $(x + \text{length}/2, y - \text{length}/2, z + \text{length}/2)$</p>
Cuboid (versi Das) createTieD() createTieD2()	<p>Input: (x, y, z) = Titik pusat objek length = Panjang ruas kubus</p> <p>Persamaan (createTieD()):</p> <p>Titik 1 = $(x - \text{length}/2, y + \text{length}/2, z - \text{length}/2)$</p>

	<p>Titik 2 = (x + length/2, y + length/2, z - length/2) Titik 3 = (x - length/2, y - length/2, z - length/2) Titik 4 = (x + length/2, y - length/2, z - length/2) Titik 5 = (x - length/2, y + length/2, z + length/2) Titik 6 = (x + length/2, y + length/2, z + length/2) Titik 7 = (x - length/10, y - length/10, z + length/10) Titik 8 = (x + length/10, y - length/10, z + length/10)</p> <p>Persamaan (createTieD2()):</p> <p>Titik 1 = (x - length/10, y + length/10, z - length/10) Titik 2 = (x + length/10, y + length/10, z - length/10) Titik 3 = (x - length/2, y - length/2, z - length/2) Titik 4 = (x + length/2, y - length/2, z - length/2) Titik 5 = (x - length/2, y + length/2, z + length/2) Titik 6 = (x + length/2, y + length/2, z + length/2) Titik 7 = (x - length/2, y - length/2, z + length/2) Titik 8 = (x + length/2, y - length/2, z + length/2)</p>
Sphere createSphere()	<p>Input: (x, y, z) = Titik pusat objek radius = jari-jari sectorCount = jumlah garis bujur stackCount = jumlah garis lintang</p> <p>Persamaan:</p> $X = \text{radius} * \text{Cos}(v) * \text{Cos}(u)$ $Y = \text{radius} * \text{Cos}(v) * \text{Sin}(u)$ $Z = \text{radius} * \text{Sin}(v)$
Elliptic Paraboloid createBody()	<p>Input: (x, y, z) = Titik pusat objek radiusX = jari-jari sumbu X radiusY = jari-jari sumbu Y radiusZ = jari-jari sumbu Z sectorCount = jumlah garis bujur stackCount = jumlah garis lintang</p> <p>Persamaan:</p> <p>Untuk nilai $v \geq 0$ dan $v \leq (\text{PI} / 2)$. Untuk nilai $u \geq -\text{PI}$ dan $u \leq \text{PI}$.</p> $X = \text{radiusX} * \text{Cos}(v) * \text{Cos}(u)$ $Y = \text{radiusY} * 2.0f * \text{Sin}(v)$ $Z = \text{radiusZ} * \text{Cos}(v) * \text{Sin}(u)$
Curve Cylinder	<p>Input:</p>

createHandDown()
createHandStrip()
createFoot()

(x, y, z) = Titik pusat objek
radius = jari-jari
height = panjang/tinggi
ext = posisi titik ke-x di kurva

Persamaan:

Dimana $v \geq (-height/2)$ dan $v \leq (height/2)$.

Dimana $u \geq -\pi$ dan $u \leq \pi$.

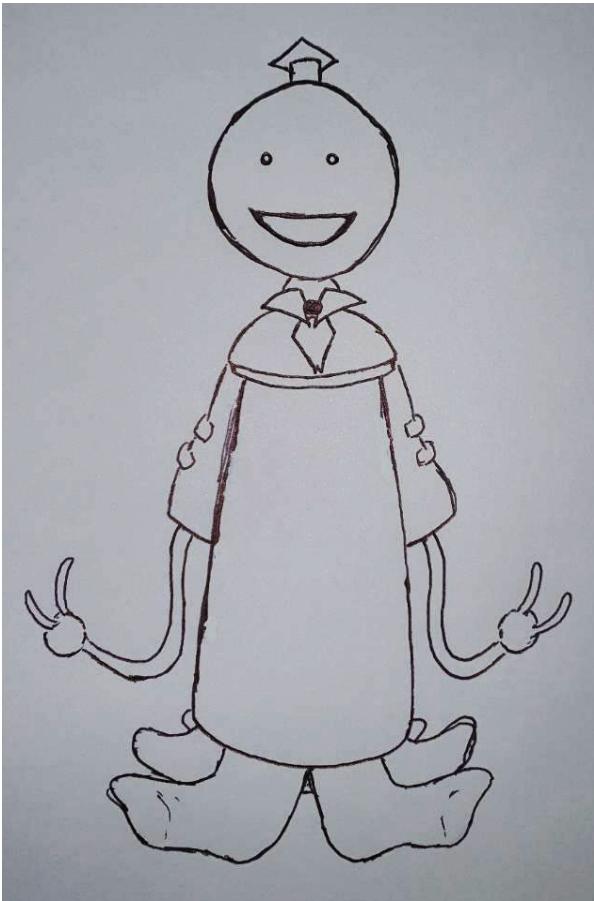
$X = Curve.X + radius * \cos(u)$

$Y = Curve.Y + radius * \sin(u)$

$Z = v$

Curve (Bezier) diatur menggunakan rumus *Pascal*.

Desain Model Karakter

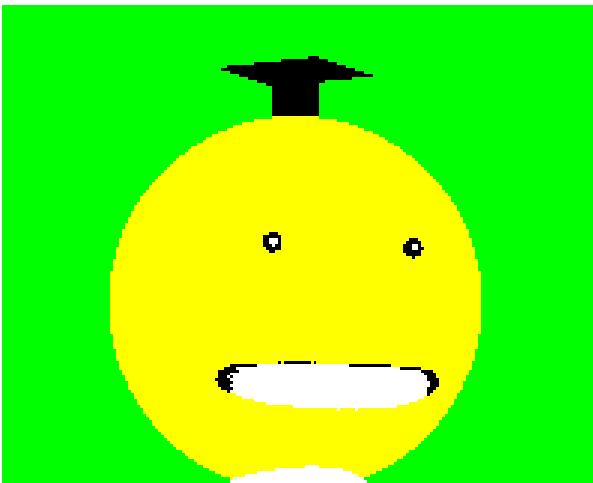


Rancangan Desain



Hasil Akhir:

Semua objek dibuat dan diatur sedemikian menggunakan fungsi *transformation*, *translation*, dan *scale* demi mendapatkan bentuk objek yang proporsional dan gerakan yang sesuai.



Kepala:

Menggunakan *ellipsoid*.

Kemudian diatur posisinya dengan menambah nilai *float Y* agar posisi kepala tepat berada diatas badan.

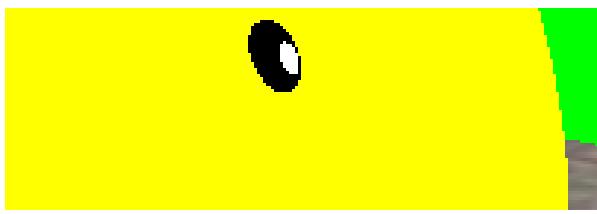
Topi

Menggunakan *cylinder* untuk batang nya dan modifikasi *cuboid* untuk membuat tutup atasnya.

Lalu posisinya diatur dengan menambah nilai *float Y* agar posisi topi tepat berada diatas kepala.

Mulut:

Menggunakan 2 buah *ellipsoid* untuk bagian hitam dan putih mulut dengan mengubah *float z* sehingga dapat menonjolkan *ellipsoid* putih. Posisinya diatur dengan mengubah *float X*, *Y*, dan *Z* sehingga mulut berada tepat pada bagian kepala di posisi yang diinginkan.

**Mata:**

Menggunakan 2 buah *sphere* untuk bagian hitam dan putih mata dengan mengubah *float z* sehingga dapat menonjolkan *sphere* putih. Posisinya diatur dengan mengubah *float X*, *Y*, dan *Z* sehingga mata berada tepat pada bagian kepala di posisi yang diinginkan.

**Kerah Jubah****Sisi dalam (Garis kuning):**

Menggunakan *elliptic paraboloid* dengan mengubah ukuran objek lewat *float y* sehingga menjadi pendek.

Sisi luar (Hitam):

Menggunakan *elliptic paraboloid* dengan mengubah ukuran objek lewat *float y* sehingga menjadi pendek.

Kerah Baju:**Sisi atas (Putih, tepat di bawah kepala):**

Menggunakan seperempat *ellipsoid*.

Kerah (Putih):

Menggunakan *cuboid*.

Buat *vertices* nya lalu sesuaikan *indices* dan ubah beberapa titik pada *vertices* nya sampai membentuk segitiga.

Kemudian atur posisinya menggunakan rotasi terhadap sumbu *X* agar didapatkan kemiringan sedemikian.

Button Dasi (Hitam):

Menggunakan *sphere* yang diatur posisi nya dengan mengubah nilai *float Z*.

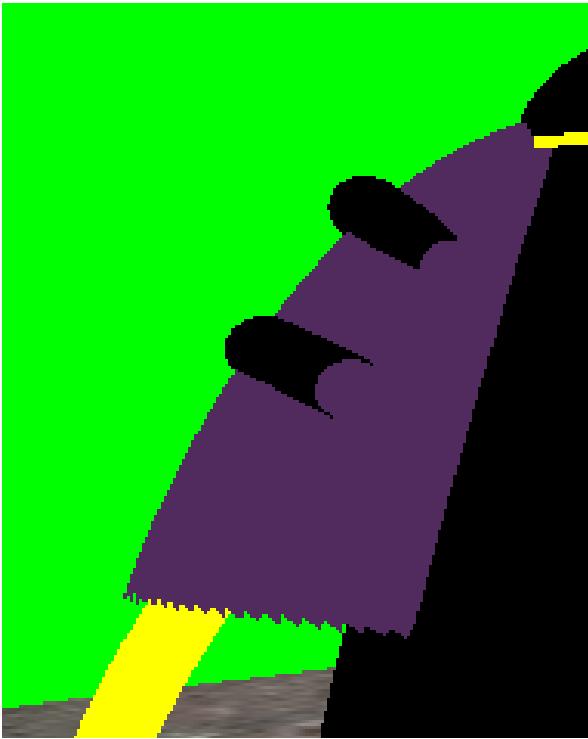
Dasi (Kuning):

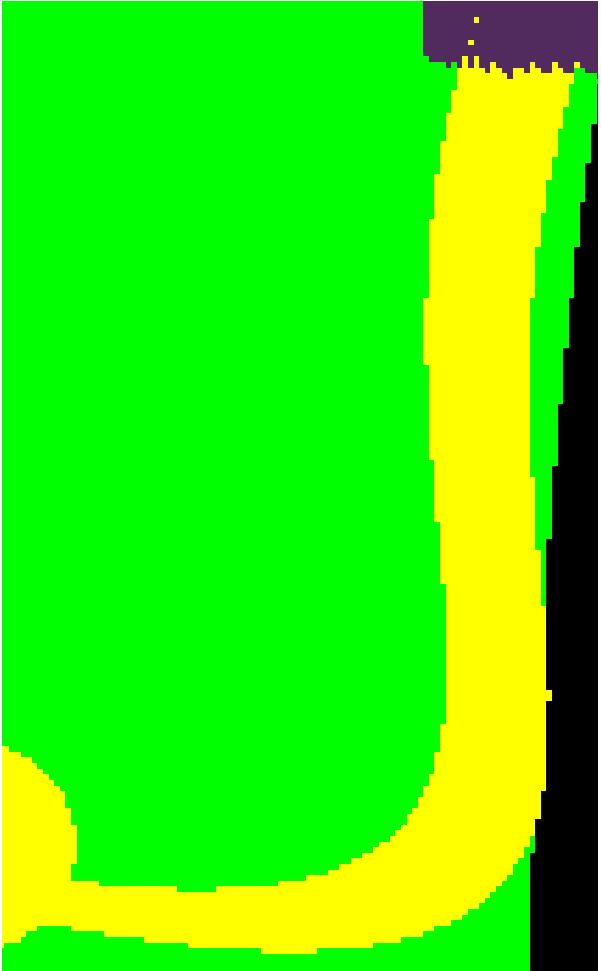
Menggunakan *cuboid*.

Buat *vertices* nya lalu sesuaikan *indices* dan ubah beberapa titik pada *vertices* nya sampai membentuk segitiga.

Kemudian atur posisinya menggunakan rotasi terhadap sumbu *X* agar didapatkan kemiringan sedemikian.

Lalu atur *float X*, *Y*, dan *Z* nya agar posisinya

	bisa tepat di depan badan sebelah atas dan tepat pada bawah kerah.
	<p>Badan (Hitam): Menggunakan <i>elliptic paraboloid</i>. Badan ini dijadikan sebagai objek Parent utama sehingga saat dilakukan <i>render</i>, fungsi hanya akan memanggil Badan.</p>
	<p>Lengan Baju (Ungu): Menggunakan <i>elliptic paraboloid</i> dengan mengatur <i>float X</i> agar posisinya pas.</p> <p>Strip (Hitam): Menggunakan <i>curve cylinder</i>. <i>Curve</i> nya dibuat menggunakan rumus <i>bezier</i> (3 titik). Kemudian dilakukan rotasi terhadap sumbu Z agar kemiringannya pas. Lalu atut <i>float X</i>, <i>Y</i>, dan <i>Z</i> nya agar posisinya bisa tepat pada lengan baju.</p> <p>Persamaan nya adalah: Dimana $v \geq (-\text{height}/2)$ dan $v \leq (\text{height}/2)$. Dimana $u \geq -\text{PI}$ dan $u \leq \text{PI}$. $X = \text{Curve.X} + \text{Cos}(u)$ $Y = \text{Curve.Y} + \text{Sin}(u)$ $Z = v$</p> <p>Untuk <i>bezier</i> 3 titik dapat digunakan perhitungan menggunakan <i>pascal</i> dimana:</p>

	<p>3 titik → 1 2 1 Titik 1 $[0] = (1-t)^2 * (t^0) * 1$</p> <p>Titik 2 $[1] = (1-t)^1 * (t^1) * 2$</p> <p>Titik 3 $[2] = (1-t)^0 * (t^2) * 1$</p>
	<p>Lengan-Tangan: Menggunakan <i>curve cylinder</i>. <i>Curve</i> nya dibuat menggunakan rumus <i>bezier</i> (7 titik). Kemudian dilakukan rotasi terhadap: Sumbu X dan Y untuk tangan kiri. Sumbu X, Y, dan Z untuk tangan kanan. Tujuannya agar kemiringannya pas. Lalu atur <i>float X</i>, <i>Y</i>, dan <i>Z</i> nya agar posisinya bisa tepat di bawah lengan baju.</p> <p>Untuk <i>bezier</i> 7 titik dapat digunakan perhitungan menggunakan <i>pascal</i> dimana: 7 titik → 1 6 15 20 15 6 1</p> <p>Titik 1 $[0] = (1-t)^6 * (t^0) * 1$</p> <p>Titik 2 $[1] = (1-t)^5 * (t^1) * 6$</p> <p>Titik 3 $[2] = (1-t)^4 * (t^2) * 15$</p> <p>Titik 4 $[3] = (1-t)^3 * (t^3) * 20$</p> <p>Titik 5 $[4] = (1-t)^2 * (t^4) * 15$</p> <p>Titik 6 $[5] = (1-t)^1 * (t^5) * 6$</p> <p>Titik 7 $[6] = (1-t)^0 * (t^6) * 1$</p>



Telapak Tangan:

Menggunakan *sphere* yang diatur posisinya lewat *float X, Y, dan Z* sehingga dapat berada tepat pada ujung tangan.

Jari Tangan:

Menggunakan *curve cylinder*. *Curve* nya dibuat menggunakan rumus *bezier* (3 titik). Kemudian dilakukan rotasi terhadap sumbu X dan Y agar kemiringannya pas.

Lalu atur *float X, Y, dan Z* nya agar posisinya bisa tepat pada telapak tangan.

Untuk *bezier* 3 titik dapat digunakan perhitungan menggunakan *pascal* dimana:

3 titik → 1 2 1

Titik 1

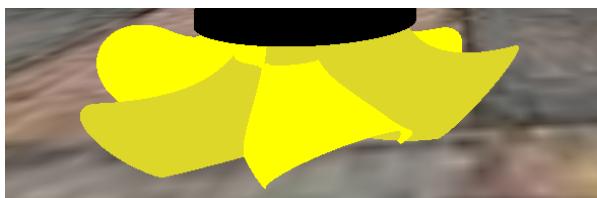
$$[0] = (1-t)^2 * (t^0) * 1$$

Titik 2

$$[1] = (1-t)^1 * (t^1) * 2$$

Titik 3

$$[2] = (1-t)^0 * (t^2) * 1$$



Kaki:

Menggunakan *curve cylinder*. *Curve* nya dibuat menggunakan rumus *bezier* (4 titik).

Kemudian dilakukan rotasi terhadap: Sumbu X dan Y untuk kaki di sisi kiri dan kanan.

Sumbu X, Y, dan Z untuk kaki di sisi depan dan belakang.

Tujuannya agar kemiringannya pas.

Lalu atur *float X, Y, dan Z* nya agar posisinya bisa tepat berada di sebelah bawah badan.

Untuk *bezier* 4 titik dapat digunakan perhitungan menggunakan *pascal* dimana:

4 titik → 1 3 3 1

Titik 1

$$[0] = (1-t)^3 * (t^0) * 1$$

Titik 2
 $[1] = (1-t)^2 * (t^1) * 3$

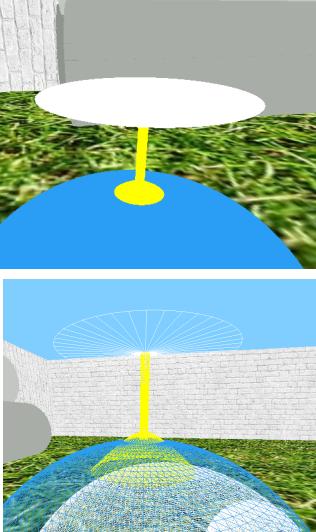
Titik 3
 $[2] = (1-t)^1 * (t^2) * 3$

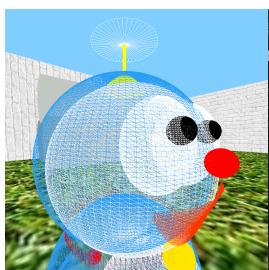
Titik 4
 $[3] = (1-t)^0 * (t^3) * 1$

Objek Dasar	
Kubus createBoxVertices()	<p>Input: (x, y, z) = Titik pusat objek length = Ruas kubus</p> <p>Persamaan: Titik 1 = $(x - \text{length}/2, y + \text{length}/2, z - \text{length}/2)$ Titik 2 = $(x + \text{length}/2, y + \text{length}/2, z - \text{length}/2)$ Titik 3 = $(x - \text{length}/2, y - \text{length}/2, z - \text{length}/2)$ Titik 4 = $(x + \text{length}/2, y - \text{length}/2, z - \text{length}/2)$ Titik 5 = $(x - \text{length}/2, y + \text{length}/2, z + \text{length}/2)$ Titik 6 = $(x + \text{length}/2, y + \text{length}/2, z + \text{length}/2)$ Titik 7 = $(x - \text{length}/2, y - \text{length}/2, z + \text{length}/2)$ Titik 8 = $(x + \text{length}/2, y - \text{length}/2, z + \text{length}/2)$</p>
Silinder Tegak createCylinderUp()	<p>Input: (x, y, z) = Titik pusat objek radius = jari-jari height = tinggi</p> <p>Persamaan: Untuk $v: 0 \leq v \leq 2\pi$: $\mathbf{X} = x + \text{radius} * \text{Cos}(v)$ $\mathbf{Y} = y \pm \text{height}/2$ $\mathbf{Z} = z + \text{radius} * \text{Sin}(v)$</p> <p>[Untuk setiap nilai v akan dihasilkan 2 titik yaitu dengan $\mathbf{Y} = y + \text{height}/2$ dan $\mathbf{Y} = y - \text{height}/2$]</p>
Silinder Tidur createCylinderSide()	<p>Input: (x, y, z) = Titik pusat objek radius = jari-jari length = panjang silinder</p> <p>Persamaan: Untuk $v: 0 \leq v \leq 2\pi$: $\mathbf{X} = x \pm \text{length}/2$ $\mathbf{Y} = y + \text{radius} * \text{Cos}(v)$ $\mathbf{Z} = z + \text{radius} * \text{Sin}(v)$</p> <p>[Untuk setiap nilai v akan dihasilkan 2 titik yaitu dengan $\mathbf{X} = x + \text{length}/2$ dan $\mathbf{X} = x - \text{length}/2$]</p>

Hyperboloid of One Sheet Tidur createHalfHyperOneSheet()	<p>Input: (x, y, z) = Titik pusat objek r = jari-jari length = panjang Hyperboloid</p> <p>Persamaan: Untuk v: $-\text{length} \leq v < 10$:</p> $k = \sqrt{r^2 + v^2}$ <p>Untuk u: $0 \leq u \leq 2\pi$:</p> $\begin{aligned} X &= x + v \\ Y &= y + k * \sin(u) \\ Z &= z + k * \cos(u) \end{aligned}$ <p>[Persamaan hanya akan membuat sebagian dari hyperboloid. Bagian yang akan digambar dikendalikan dengan range v]</p>
Hyperboloid of Two Sheet Bawah createHalfHyperTwoSheet()	<p>Input: (x, y, z) = Titik pusat objek r = jari-jari height = tinggi Hyperboloid</p> <p>Persamaan: Untuk v: $-r \geq v \geq (-r - \text{height})$:</p> $k = \sqrt{v^2 - r^2}$ <p>Untuk u: $0 \leq u \leq 2\pi$:</p> $\begin{aligned} X &= x + k * \cos(u) \\ Y &= y + v \\ Z &= z + k * \sin(u) \end{aligned}$ <p>[Persamaan hanya akan membuat bagian bawah dari hyperboloid. Bagian yang akan digambar dikendalikan dengan range v]</p>
Lingkaran createCircleUp()	<p>Input: (x, y, z) = Titik pusat objek r = jari-jari</p> <p>Persamaan: Untuk v: $0 \leq v \leq 2\pi$:</p> $\begin{aligned} X &= x + r * \cos(v) \\ Y &= y \\ Z &= z + r * \sin(v) \end{aligned}$

Ellipsoid createEllipsoid()	<p>Input:</p> <p>(x, y, z) = Titik pusat objek radX = jari-jari x radY = jari-jari y radZ = jari-jari z</p> <p>Persamaan:</p> <p>Untuk v: $-\pi/2 \leq v \leq \pi/2$: Untuk u: $0 \leq u \leq 2\pi$:</p> $X = x + radX * \cos(v) * \cos(u)$ $Y = y + radY * \sin(v)$ $Z = z + radZ * \cos(v) * \sin(u)$
Bezier Curve createBezierCurve()	<p>Input:</p> <p>guide[] = Array atau List berisi titik-titik yang memandu kurva. Lambang titik guide adalah P.</p> <p>Persamaan:</p> <p>row = jumlah isi guide pascal[] = segitiga pascal dari barisan (row-1)</p> <p>Untuk t: $0 \leq t \leq 1$:</p> $k_i = nCr_i * t^i * (1-t)^{row-1-i}$ $X = \sum_{i=0}^{row} (nCr_i * t^i * (1-t)^{row-1-i} * P_i.X)$ $Y = \sum_{i=0}^{row} (nCr_i * t^i * (1-t)^{row-1-i} * P_i.Y)$ $Z = \sum_{i=0}^{row} (nCr_i * t^i * (1-t)^{row-1-i} * P_i.Z)$
Desain Model Karakter	
	<p>Rancangan Desain</p>

	<p>Hasil Akhir</p> <p>Pada saat penggabungan projek, karakter secara keseluruhan diberi transformasi scale 2 kali dan transformasi translasi ke menuju titik awal karakter agar sesuai dengan proporsi model karakter lain.</p>
 <p>(wireframe)</p>	<p>Baling-baling Bambu</p> <p>Bagian ini terdiri beberapa sub-bagian yaitu:</p> <ul style="list-style-type: none"> - Dasar (<i>balingBase</i>) - Batang (<i>balingStick</i>) - Baling-baling (<i>balingBlade</i>) <p>Sub-bagian dasar dibuat dari Hyperboloid of Two Sheet. Pada saat pembentukan karakter, sub-bagian dasar ditranslasi ke arah y positif agar dapat muncul di atas kepala Doraemon.</p> <p>Sub-bagian batang dibuat dari Silinder Tegak.</p> <p>Sub-bagian baling-baling dibuat dengan Lingkaran.</p> <p>Bagian Baling-baling Bambu merupakan child dari bagian Kepala.</p>



(wireframe)

Kepala

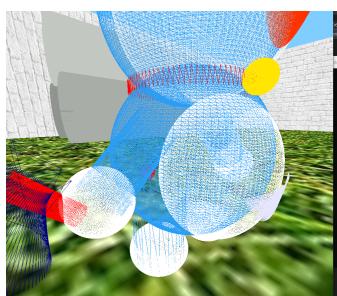
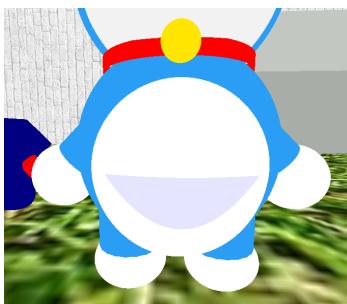
Bagian ini terdiri beberapa sub-bagian yaitu:

- Kepala (*head*)
- Muka (*faceBase*)
- Putih mata kiri (*eyeLeft*)
- Putih mata kanan (*eyeRight*)
- Pupil kiri (*pupilLeft*)
- Pupil kanan (*pupilRight*)
- Hidung (*nose*)
- Mulut (*mouth*)

Sub-bagian kepala, muka, putih mata kiri dan kanan, pupil kiri dan kanan, serta hidung dibuat dengan **Ellipsoid**.

Sub-bagian mulut dibuat dengan 2 **Bezier Curve** (*lipUp* dan *lipBottom*). 1 kurva untuk sisi atas dan 1 kurva untuk sisi bawah mulut. Masing-masing kurva menggunakan 3 titik pandu. Dalam membentuk karakter, sub-bagian ini ditranslasi ke arah z negatif agar dapat menempel pada muka Doraemon.

Bagian Kepala merupakan child dari bagian Badan.



(wireframe)

Badan

Bagian ini terdiri beberapa sub-bagian yaitu:

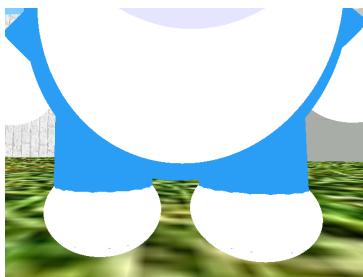
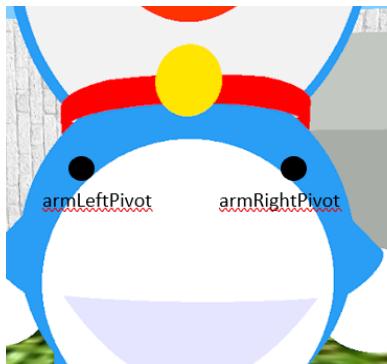
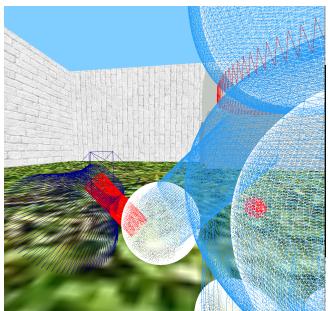
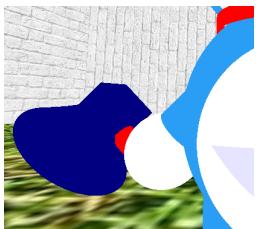
- Badan (*body*)
- Perut (*belly*)
- Kalung (*collar*)
- Bell (*bell*)
- Ekor (*tail*)
- Kantong (*pouch*)

Sub-bagian badan, perut, bell, dan ekor dibuat dengan **Ellipsoid**.

Sub-bagian kalung dibuat dengan **Silinder Tegak**.

Sub-bagian kantong dibuat dengan 2 **Bezier Curve** (*pouchUp* dan *pouchBottom*). 1 kurva untuk sisi atas dan 1 kurva untuk sisi bawah kantong. Masing-masing kurva menggunakan 3 titik pandu. Dalam membentuk karakter, sub-bagian kantong ditranslasi ke arah z negatif agar dapat menempel pada perut Doraemon.

Bagian ini memiliki hirarki teratas.



Lengan

Bagian ini terdiri beberapa sub-bagian yaitu:

- Lengan kiri (*armLeft*)
- Lengan kanan (*armRight*)
- Tangan kiri (*handLeft*)
- Tangan kanan (*handRight*)

Bagian ini memiliki 2 titik pivot untuk titik pusat rotasi, yaitu:

- Pivot lengan kiri (*armLeftPivot*)
- Pivot lengan kanan (*armRightPivot*)

Keduanya hanya berupa titik yang merupakan child dari *body*.

[“kiri” dan “kanan” pada nama sub-bagian terbalik dengan kiri dan kanan dari perspektif Doraemon. Contoh: Tangan kiri Doraemon memegang Sirene]

Sub-bagian lengan kiri dan kanan dibuat dengan **Silinder Tidur**.

Sub-bagian tangan kiri dan kanan dibuat dengan **Ellipsoid**.

Bagian ini merupakan child dari bagian badan.

Pada saat pembuatan karakter, lengan kiri dan tangan kiri dirotasi sumbu Z dengan titik pusat rotasi *armLeftPivot* sebanyak 45° . Sedangkan, lengan dan tangan kanan dirotasi sumbu Z dengan titik pusat rotasi *armRightPivot* sebanyak -45° . Hal ini dilakukan agar lengan berada posisi yang lebih natural.

Kaki

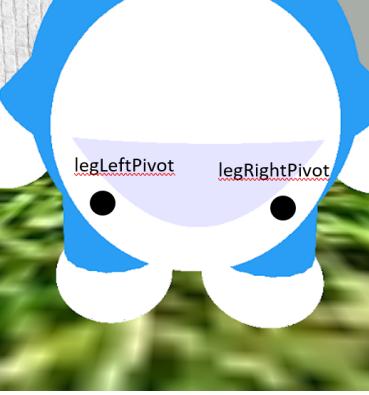
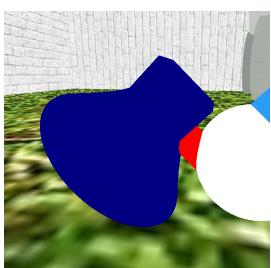
Bagian ini terdiri beberapa sub-bagian yaitu:

- Tungkai kaki kiri (*legLeft*)
- Tungkai kaki kanan (*legRight*)
- Kaki kiri (*footLeft*)
- Kaki kanan (*footRight*)

Bagian ini memiliki 2 titik pivot untuk titik pusat rotasi, yaitu:

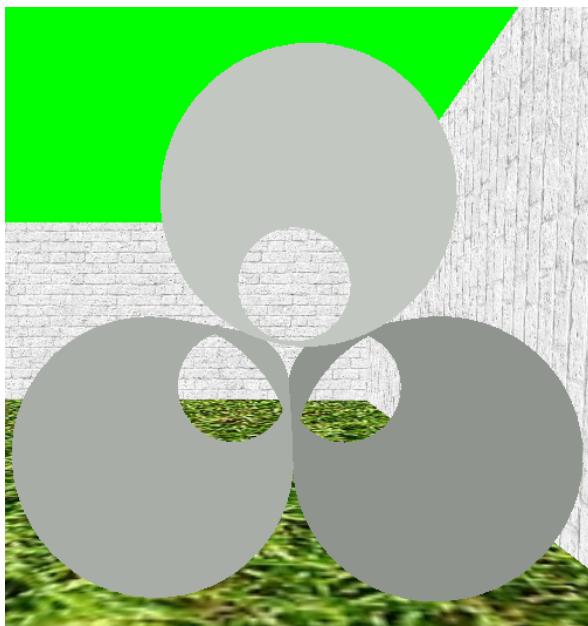
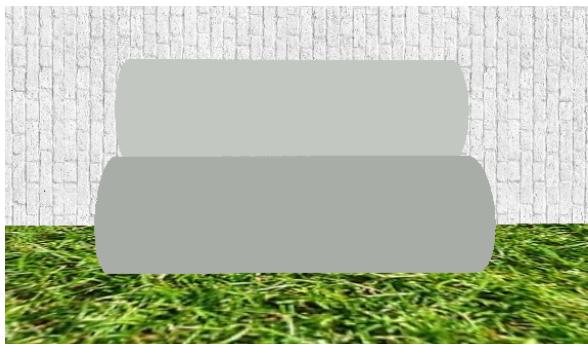
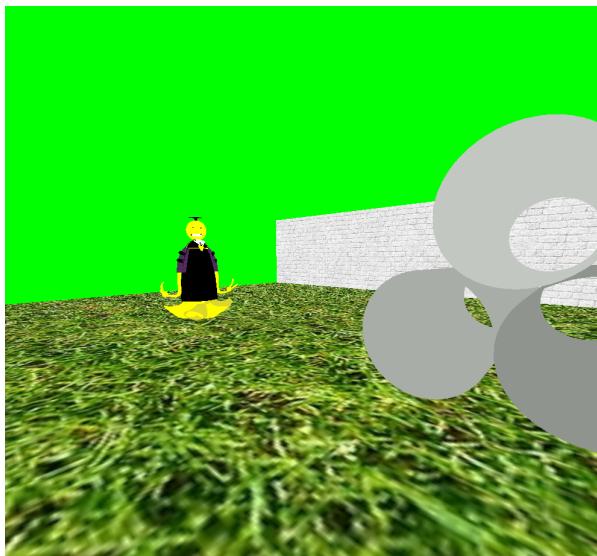
- Pivot kaki kiri (*legLeftPivot*)
- Pivot kaki kanan (*legRightPivot*)

Keduanya hanya berupa titik yang merupakan child dari

	<p><i>body.</i></p> <p>[“kiri” dan “kanan” pada nama sub-bagian terbalik dengan kiri dan kanan dari perspektif Doraemon.]</p> <p>Sub-bagian tungkai kaki kiri dan kanan dibuat dengan Silinder Tegak.</p> <p>Sub-bagian kaki kiri dan kanan dibuat dengan Ellipsoid.</p> <p>Bagian ini merupakan child dari bagian badan.</p>
	<p>Sirene/Megaphone</p> <p>Bagian ini terdiri beberapa sub-bagian yaitu:</p> <ul style="list-style-type: none"> - Corong (<i>sirene</i>) - Belakang sirene (<i>sireneHandle</i>) - Pegangan (<i>sireneBack</i>) <p>Sub-bagian corong dibuat dengan Hyperboloid of One Sheet Tidur. Pada saat pembuatan karakter, sub-bagian ini di-scale sebanyak 0.5 kali agar proporsional dengan tangan Doraemon.</p> <p>Sub-bagian belakang sirene dibuat dengan Kubus.</p> <p>Sub-bagian pegangan dibuat dari Silinder Tegak.</p> <p>Bagian ini merupakan child dari <i>handLeft</i> di bagian lengan. Sehingga saat pembuatan karakter, bagian ini juga ikut ter-rotasi Sumbu Z.</p>

Desain Environment

	<p>Lantai: Dibuat dengan memanfaatkan <i>texture</i>.</p> <p>Pertama buat titik <i>vertices</i> dan atur <i>indices</i> nya sehingga membentuk objek yaitu <i>cuboid</i> sisi bawah yang akan berfungsi sebagai lantai. Kemudian <i>load</i> file image yang akan digunakan dan aplikasikan <i>texture</i> pada objek tersebut.</p>
	<p>Tembok: Dibuat dengan memanfaatkan <i>texture</i>.</p> <p>Pertama buat titik <i>vertices</i> dan atur <i>indices</i> nya sehingga membentuk objek yaitu <i>cuboid</i> sisi tegak dan sisi yang menghadap depan dimana objek yang dibuat akan berfungsi sebagai tembok.</p> <p>Setelah selesai memposisikan objek, <i>load</i> file image yang akan digunakan dan aplikasikan <i>texture</i> pada objek tersebut.</p>



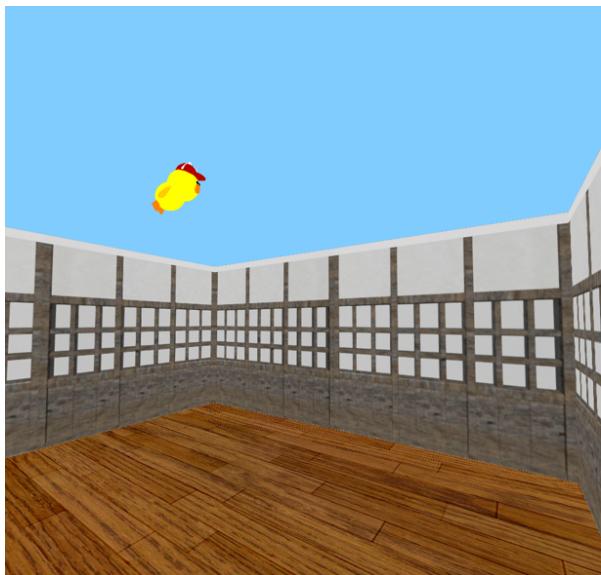
Pipe:

Objek dibuat menggunakan rumus *Cylinder*.

ANIMASI

Ivan Chandra

	<p>Animasi karakter sally dimulai dengan memiringkan badan dan kepalanya pada sumbu X.</p> <p>Untuk animasi sayap digunakan <code>armAngle</code> yang berisi <code>armAngle += args.Time</code>. <code>armAngle</code> ini akan berfungsi sebagai derajat pembatas yang akan diputar pada sayap.</p> <p>Adapun range angle untuk putaran sayapnya: $0^\circ < \text{angle} < 45^\circ$</p>
	<p>Animasi duduk pada sally dimulai dengan posisi memiringkan badan dan kakinya berdasarkan sumbu X, kemudian dilakukan animasi pada kaki.</p> <p>Untuk animasi kaki ini digunakan <code>shinAngle</code> yang berisi <code>shinAngle += args.Time</code>. <code>shinAngle</code> ini akan berfungsi sebagai derajat pembatas yang akan digunakan untuk merotasi kaki pada angle tertentu</p> <p>Adapun range angle untuk putaran kakinya: $0^\circ < \text{angle} < 35^\circ$</p>



Untuk menggerakkan karakter sally, digunakan fungsi translasi pada sumbu Y dimana persamaan ini menggunakan increment Time yang berisi incrementTime += args.Time

Persamaan translasi pada sumbu Y:
 $\text{Sin}(\text{incrementTime} * 2) * 0.6f$

Untuk rotasi pada seluruh karakter dilakukan rotasi pada pivot vektor (8, 0, 8). Dengan kecepatan $90 * \text{deltaTime}$.

Adapun Input untuk Sally apabila ditekan input key F, akan berhenti terbang dan sebaliknya



Head Movement:

Kepala akan diputar pada sumbu Y dan pada titik tertentu akan melakukan gerakan berlawanan arah sehingga akan kelihatan seperti gerakan menggelengkan kepala.

Kepala akan diberi transformasi rotasi sumbu Y dengan titik pusat objeknya sendiri sebagai *pivot*. Setiap kali *render*, kepala akan dirotasi sebesar $30 * Time$. Jika kepala telah dirotasi sebesar 45° ke arah kanan, maka arah rotasi akan berbalik. Jika kepala telah melakukan rotasi sebesar -45° , maka arah rotasi kembali seperti semula.

Karena dibuat dengan mengimplementasikan prinsip hierarki maka saat kepala melakukan rotasi, seluruh objek child 3D milik kepala (Topi, Mata, dan Mulut) juga akan ikut melakukan rotasi tanpa berubah posisinya dari kepala.

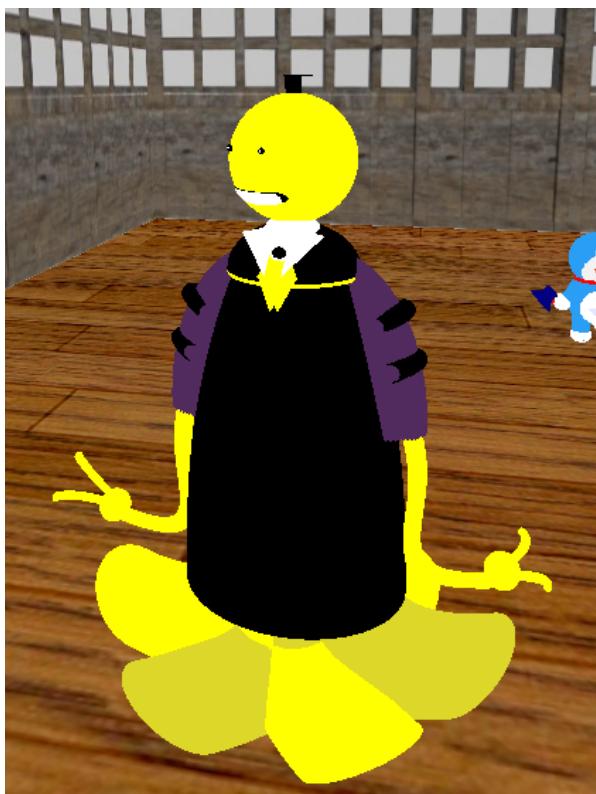


Hand Movement:

Lengan baju dan Tangan akan melakukan gerakan naik-turun sedemikian.

Lengan baju akan diberi transformasi rotasi sumbu Z dengan titik pusat objeknya sendiri sebagai *pivot*. Setiap kali *render*, lengan akan dirotasi sebesar $15f * Time$. Jika lengan baju telah dirotasi sebesar 30° ke arah bawah, maka arah rotasi akan berbalik. Jika lengan baju telah melakukan rotasi ke titik 0° , maka arah rotasi kembali seperti semula.

Karena saya mengimplementasikan prinsip hierarki maka saat lengan baju melakukan rotasi, seluruh objek child 3D milik lengan baju (Tangan, Telapak tangan, dan jari tangan) juga akan ikut melakukan rotasi tanpa berubah posisinya dari lengan dan badan.

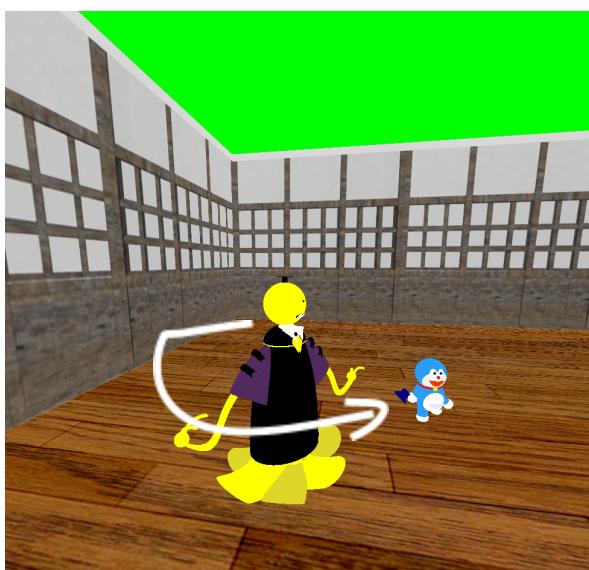
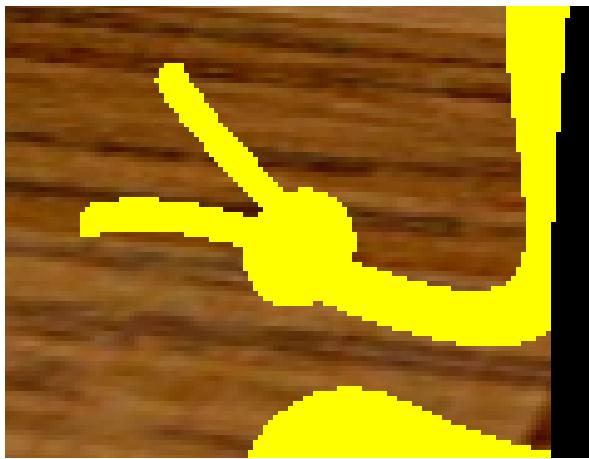


Palm Movement:

Telapak tangan akan melakukan gerakan rotasi sehingga terlihat seperti sedang mengibaskan/memutar telapak.

Telapak tangan akan diberi transformasi rotasi sumbu Z dengan titik pusat objeknya sendiri sebagai *pivot*. Setiap kali *render*, telapak tangan akan dirotasi sebesar $15f * Time$. Jika telapak tangan telah dirotasi sebesar 30° ke arah bawah, maka arah rotasi akan berbalik. Jika lengan telah melakukan rotasi ke titik 0° , maka arah rotasi kembali seperti semula.

Karena saya mengimplementasikan prinsip hierarki maka saat telapak tangan melakukan rotasi, seluruh objek child 3D milik telapak tangan (Jari tangan) juga akan ikut melakukan rotasi tanpa berubah posisinya dari lengan dan telapak tangan.



Land Movement:

Karakter akan bergerak mengikuti rute yang sudah diatur menggunakan penambahan *counter*, *inner-counter*, dan *switch-case*.

Counter berfungsi sebagai faktor untuk melakukan *switch-case*.

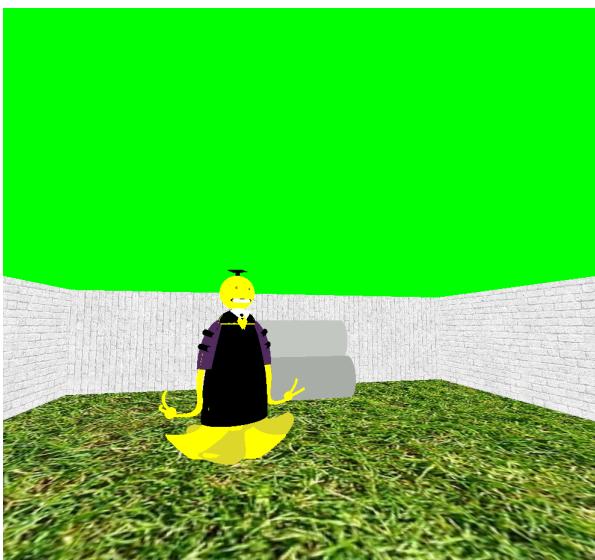
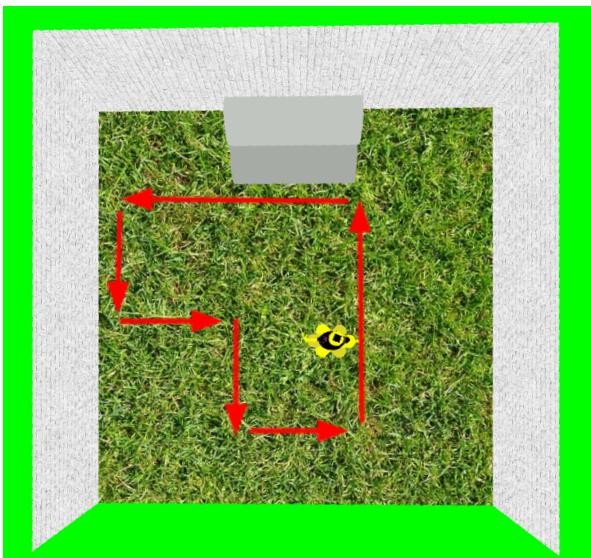
Inner-Counter berfungsi untuk menghitung waktu yang dilakukan selama 1 animasi pada 1 *case*.

Setiap kali satu *case* selesai maka *counter* akan ditambahkan untuk masuk ke *case* berikutnya dan *inner-counter* akan di set menjadi 0.

Apabila telah mencapai *case* terakhir, *counter* dan *inner-counter* akan di set menjadi 0 sehingga proses animasi akan diulang lagi dari *case* awal.

Karakter melakukan gerakan maju ke arah depan menggunakan fungsi *translation* dan disesuaikan dengan posisi karakter dan kemana arah hadapnya.

Jika karakter telah sampai pada suatu titik, maka karakter akan dirotasi terhadap sumbu Y lalu akan melanjutkan translasi sesuai arahnya untuk bergerak maju kedepan.



Aerial Movement:

Karakter akan melakukan translasi yang diikuti rotasi terhadap sumbu Y secara keseluruhan yang meliputi gerak seluruh objek pada karakter tersebut dan melakukan gerakan yang terkesan seperti terbang.

Diberikan batas dimana apabila posisi titik pusat Y karakter mendekati lantai maka konstanta pengali nya akan jadi -1 dan sebaliknya apabila titik pusat Y karakter naik mendekati 5.5f maka akan jadi +1 dimana hal ini dilakukan agar karakter dapat terus melakukan gerakan naik-turun terhadap sumbu Y objek.



Urutan State Animasi

Case 0:

Karakter akan melakukan *aerial movement*.

Case 1:

Karakter akan ditranslasi terhadap sumbu Z untuk melakukan gerakan berjalan kedepan. Karakter akan melakukan *head movement*, *hand movement* dan *palm movement*.

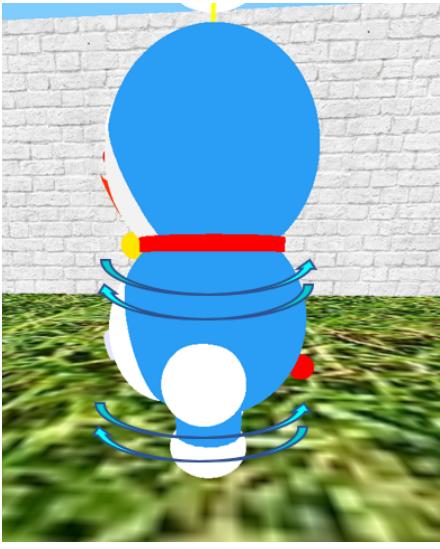
Case 2:

Karakter akan dirotasi ke kiri pada sumbu Y.

Case 3:

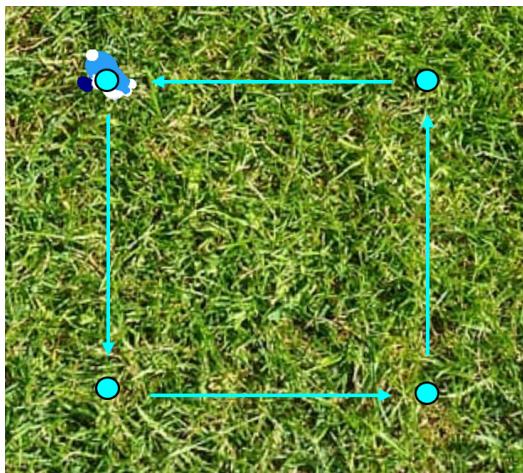
Karakter akan ditranslasi terhadap sumbu X

	<p>untuk melakukan gerakan berjalan kedepan. Karakter akan melakukan <i>head movement</i>, <i>hand movement</i> dan <i>palm movement</i>.</p> <p>Case 4: Karakter akan dirotasi ke kiri pada sumbu Y.</p> <p>Case 5: Karakter akan ditranslasi terhadap sumbu Z untuk melakukan gerakan berjalan kedepan. Karakter akan melakukan <i>head movement</i>, <i>hand movement</i> dan <i>palm movement</i>.</p> <p>Case 6: Karakter akan dirotasi ke kiri pada sumbu Y.</p> <p>Case 7: Karakter akan ditranslasi terhadap sumbu X untuk melakukan gerakan berjalan kedepan. Karakter akan melakukan <i>head movement</i>, <i>hand movement</i> dan <i>palm movement</i>.</p> <p>Case 8: Karakter akan dirotasi ke kiri pada sumbu Y.</p> <p>Case 9: Karakter akan ditranslasi terhadap sumbu Z untuk melakukan gerakan berjalan kedepan. Karakter akan melakukan <i>head movement</i>, <i>hand movement</i> dan <i>palm movement</i>.</p> <p>Case 10: Karakter akan dirotasi ke kiri pada sumbu Y.</p> <p>Case 11: Karakter akan ditranslasi terhadap sumbu X untuk melakukan gerakan berjalan kedepan. Karakter akan melakukan <i>head movement</i>, <i>hand movement</i> dan <i>palm movement</i>.</p> <p>Case 12: Karakter akan dirotasi ke kanan pada sumbu Y dimana akan dilakukan reset <i>counter</i> sehingga gerakan akan diulang ke Case 0.</p>
--	---

 The first screenshot shows Doraemon from the front, smiling with his head tilted back and his arms slightly raised, indicating a shaking motion. The second screenshot shows him from the front again, but with his head upright and a more neutral or slightly surprised expression.	<p>Shaking Animasi ini akan membuat Doraemon menggeleng-gelengkan kepalanya. Animasi ini akan berjalan setiap saat.</p> <p>Bagian Kepala dan semua childnya akan diberi transformasi sumbu Y dengan titik pusat Kepala sebagai titik pusat rotasi.</p> <p>Animasi ini memiliki batas rotasi yaitu dari -20° dan 20°. Saat rotasi Kepala sudah melewati batas, arah rotasi akan dibalik. Misalkan jika Kepala telah diputar ke kanan melewati 20°, maka arah rotasi diubah menjadi ke arah kiri.</p>
 A screenshot showing Doraemon from behind, walking towards the left. His arms are swinging to the left, and his legs are moving forward in a walking cycle. Blue lines and arrows indicate the pivot points for the arms and legs, showing the skeletal structure during movement.	<p>animateStep Animasi ini akan membuat Doraemon mengayunkan tangan dan kakinya dalam gerakan berjalan.</p> <p>Animasi ini bergantung pada titik-titik pivot bagian Lengan dan Kaki (total ada 4 titik). Batas rotasi animasi ini adalah -30° dan 30°.</p> <p>Lengan Kiri akan diberi transformasi rotasi sumbu X dengan <i>armLeftPivot</i> sebagai titik pusat rotasinya, sedangkan Lengan Kanan akan diberi transformasi rotasi sumbu X mengikuti <i>armRightPivot</i>. Arah rotasi lengan kiri dan kanan berlawanan arah.</p> <p>Kaki Kiri akan diberi transformasi rotasi sumbu X memutari <i>legLeftPivot</i>. Kaki Kanan akan diberi transformasi rotasi sumbu X mengikuti <i>legRightPivot</i>. Arah rotasi kaki kanan dan kiri berlawanan arah. Arah rotasi kaki kiri sama dengan lengan kanan dan</p>



sebaliknya.



Move

Animasi ini akan menggerakkan Doraemon membentuk suatu kotak.

Animasi akan memberi Doraemon secara utuh transformasi translasi. Transformasi diatur dengan variabel *direction* sebagai berikut:

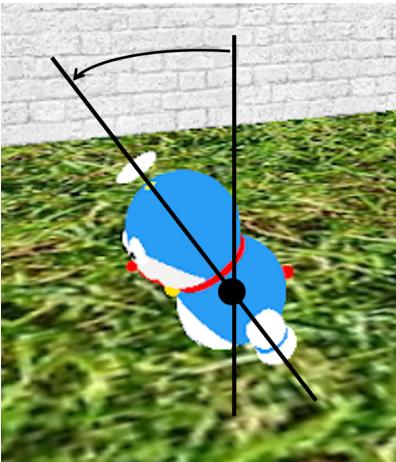
- *direction* %4 = 0
Doraemon ditranslasi arah Z positif
- *direction* %4 = 1
Doraemon ditranslasi arah X positif
- *direction* %4 = 2
Doraemon ditranslasi arah Z negatif
- *direction* %4 = 3
Doraemon ditranslasi arah X negatif

Saat animasi ini dijalankan, animasi **animasiStep** juga dijalankan.



Turning

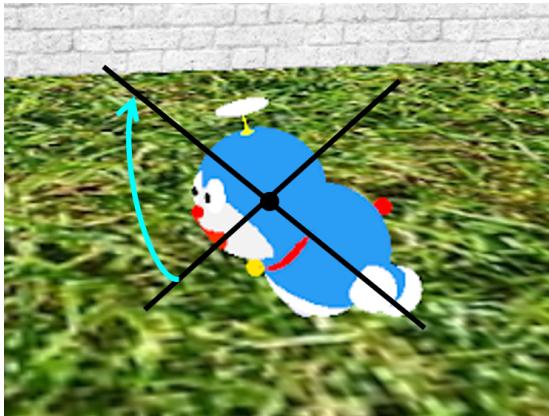
Animasi ini akan memberi Doraemon secara utuh transformasi rotasi sumbu Y dengan titik pusat *body* sebagai titik rotasinya. Total besar rotasi dari animasi ini adalah 90°.



boardingBody

Animasi ini adalah salah satu animasi untuk mempersiapkan Doraemon untuk terbang.

Animasi ini akan memberi Doraemon secara keseluruhan transformasi rotasi sumbu X titik pusat *body* sebagai titik pusat rotasinya. Total besar rotasi dari animasi ini adalah 60° .

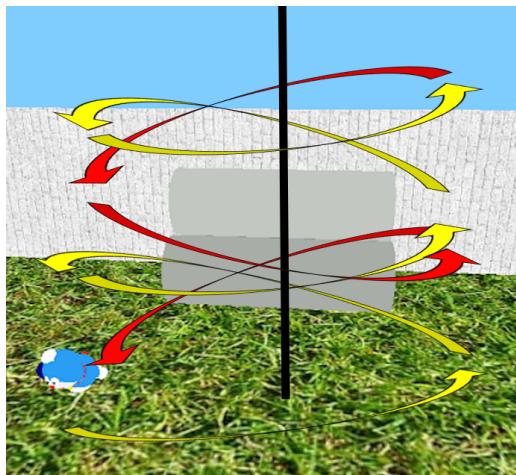


boardingHead

Animasi ini adalah salah satu animasi untuk mempersiapkan Doraemon untuk terbang dengan memutar kepala Doraemon memandang ke depan.

Animasi ini akan memberi Kepala transformasi rotasi sumbu X titik pusat *head* sebagai titik pusat rotasinya. Total besar rotasi dari animasi ini adalah -60° .

Animasi ini juga men-translasi Kepala dengan arah y negatif dan z positif untuk menjaga Kepala tetap melekat pada Badan.

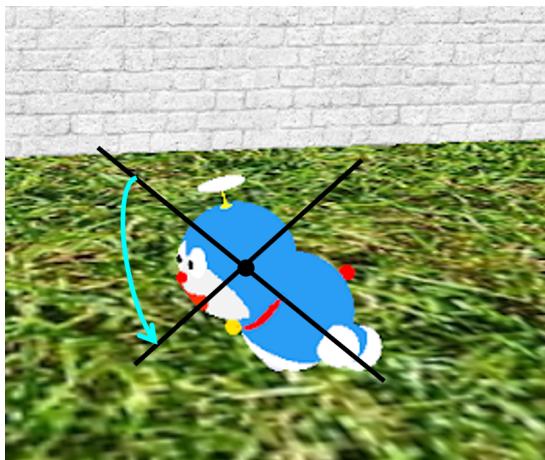
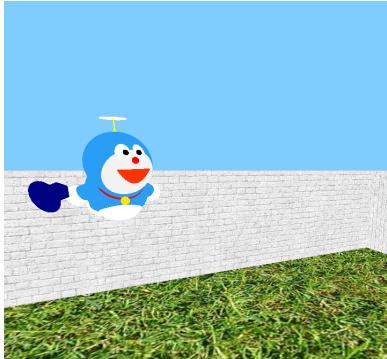


Fly

Animasi ini akan membuat Doraemon terbang dalam bentuk spiral atau per.

Animasi ini akan memberi transformasi rotasi Sumbu Y pada Doraemon secara utuh dengan titik (0,0,0) (diluar Doraemon) sebagai pusat rotasinya.

Selain itu, Animasi ini juga akan memberi Doraemon secara utuh transformasi translasi atas/turun. Maka Doraemon juga akan naik/turun. Ketinggian Doraemon diberi batas atas yaitu $y=20$, sedangkan batas bawah adalah $y=2$. Jika ketinggian Doraemon melebihi batas, arah translasi akan dibalik.

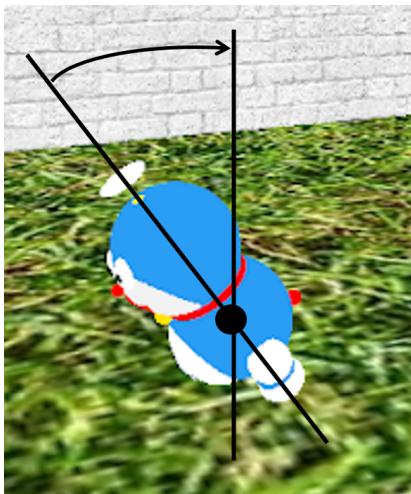


landingHead

Animasi ini adalah salah satu animasi untuk mempersiapkan Doraemon untuk mendarat. Animasi ini seperti **boardingHead** dengan arah terbalik

Animasi ini akan memberi Kepala transformasi rotasi sumbu X titik pusat *head* sebagai titik pusat rotasinya. Total besar rotasi dari animasi ini adalah 60° .

Animasi ini juga men-translasi Kepala dengan arah y positif dan z negatif untuk menjaga Kepala tetap melekat pada Badan.



landingBody

Animasi ini adalah salah satu animasi untuk mempersiapkan Doraemon untuk terbang. Animasi ini seperti **boardingBody** dengan arah terbalik

Animasi ini akan memberi Doraemon secara keseluruhan transformasi rotasi sumbu X titik pusat *body* sebagai titik pusat rotasinya. Total besar rotasi dari animasi ini adalah -60° .

Urutan State Animasi

State = 0

Animasi **move**, **animateStep** akan berjalan. Sehingga Doraemon terlihat seperti berjalan lurus. Arah jalan membentuk kotak.
 $direction = direction + 1$

	<p>Lalu State diubah menjadi State = 1</p> <p>State = 1 Animasi turning akan berjalan. Sehingga Doraemon berputar 90°. Jika <i>direction</i> = 8, State dijadikan 2 dan <i>direction</i> = 0. Jika tidak, kembali State = 0.</p> <p>State = 2 Animasi boardingBody akan berjalan Lalu Mengubah State menjadi State = 3</p> <p>State = 3 Animasi boardingHead akan berjalan Lalu Mengubah State menjadi State = 4</p> <p>State = 4 Animasi fly akan berjalan. Lalu state diubah menjadi State = 5</p> <p>State = 5 Animasi boardingHead akan berjalan. Lalu state diubah menjadi State = 5</p> <p>State = 6 Animasi boardingBody akan berjalan. Lalu state diubah menjadi State = 0</p>
--	--

PENUTUP

Demikian laporan yang telah kami buat, kami berharap bahwa laporan ini dapat menjelaskan proyek kami dengan jelas. Kami mohon maaf apabila ada kesalahan pada ejaan dan penulisan kata. Atas waktunya kami ucapkan terima kasih.