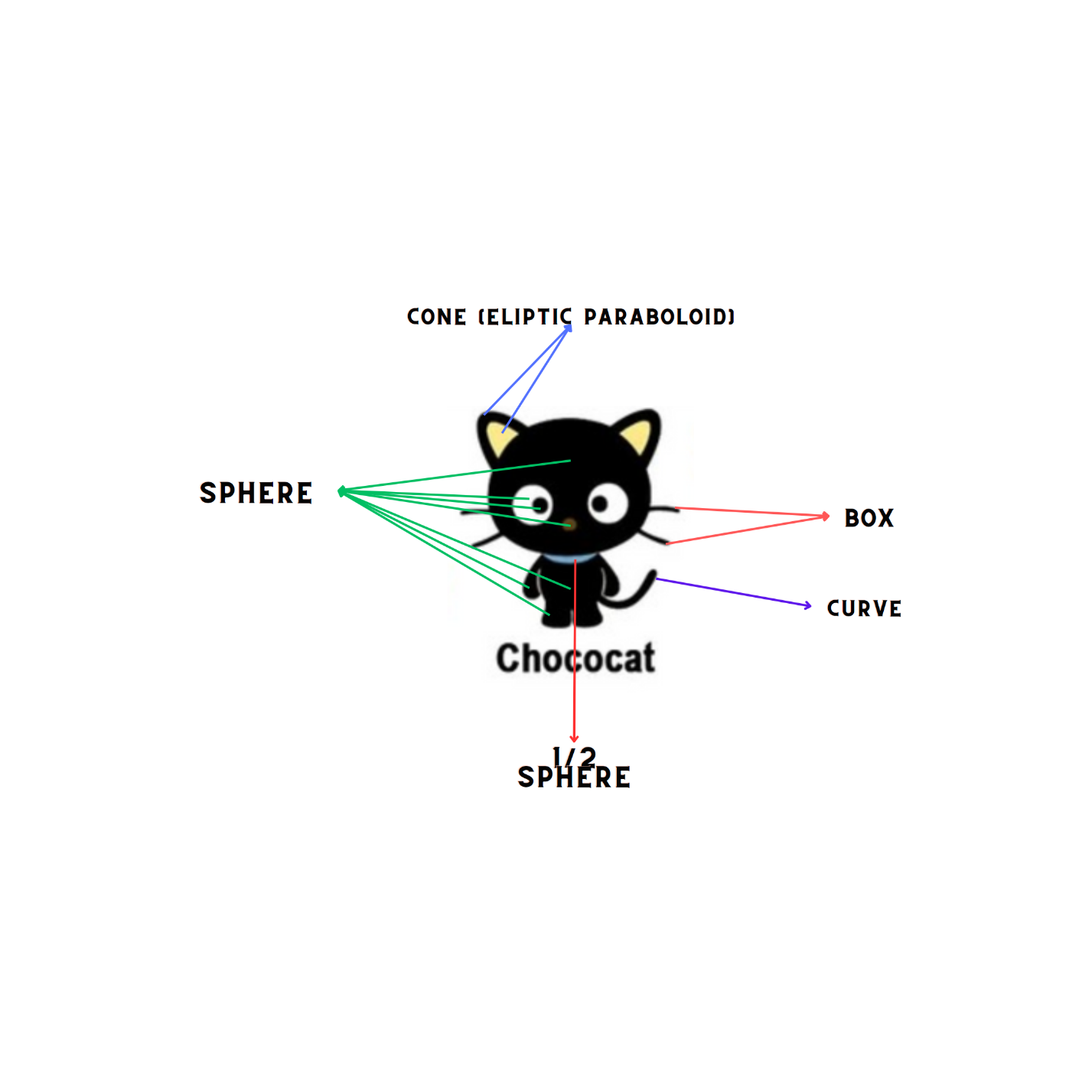
Laporan Proyek I Grafika Komputer

Nama : Vincent Christopher

NRP : C14210205

Tema : Character Hello Kitty

Model : Chococat

1. Gambarlah sebuah sket sebuah 3D obyek/character dengan tema tertentu (obyek kompleks) dengan menggunakan gabungan obyek-obyek 3D sederhana, kurva dan min 3 jenis parametric quadric (atau obyek 3D lainnya yang digenerate dengan rumus). **Buatlah modifikasi pada rumus modeling obyek 3D** sedemikian rupa sehingga mendapatkan bentuk yang diinginkan. Implementasikan sket tersebut dalam pemrograman. Atur penggabungan komponen masing2 jenis obyek 3D dengan menggunakan transformasi seperti mengatur ukuran, kemiringan/rotasi, mengatur posisi dengan translasi. Posisikan masing-masing obyek sehingga membentuk gabungan yang diinginkan. Tuliskan rancangan obyek dan masing-masing komponennya beserta dengan transformasi yang digunakan. 

Character model yang akan saya gunakan adalah Chococat.

* Kepala (Parent 1)

Pada bagian kepala saya menggunakan object sphere yang berwarna hitam. Transformasi yang gunakan adalah scale object agar sphere bisa menyerupai bentuk kepala seperti pada model saya. Kepala merupakan titik default 0,0 saya. Sehingga tidak perlu translasi untuk merubah posisi kepala.

objects.get(0).scaleObject(1f, 0.85f, 0.75f);

* Telinga (Child Kepala)

Pada bagian telinga saya menggunakan object cone atau eliptic paraboloid dengan warna hitam. Transformasi yang saya gunakan adalah scale, translate, dan rotate. Saya men-scale cone terhadap sumbu x,y, dan z agar besarnya bisa proporsional dengan kepala dan agar ukurannya bisa sesuai dengan model telinga saya. Kemudian saya menggunakan rotasi terhadap sumbu x dan z agar posisinya bisa sedikit miring ke kiri atau kanan dan sedikit terbuka ke atas. Lalu setelah kemiringannya sesuai, saya menggunakan translasi terhadap sumbu x dan sumbu y untuk memposisikan telinga kanan dan telinga kiri di bagian kiri dan kanan atas kepala.

//Right

objects.get(0).getChildObject().get(0).scaleObject(0.012f, 0.005f, 0.0025f);  
objects.get(0).getChildObject().get(0).rotateObject((*float*) Math.*toRadians*(90f), 1f, 0f, 0f);  
objects.get(0).getChildObject().get(0).rotateObject((*float*) Math.*toRadians*(-38f), 0f, 0f, 1f);  
objects.get(0).getChildObject().get(0).translateObject(0.12f, 0.12f, 0.0f);

//Left

objects.get(0).getChildObject().get(1).scaleObject(0.012f, 0.005f, 0.0025f);  
objects.get(0).getChildObject().get(1).rotateObject((*float*) Math.*toRadians*(95f), 1f, 0f, 0f);  
objects.get(0).getChildObject().get(1).rotateObject((*float*) Math.*toRadians*(38f), 0f, 0f, 1f);  
objects.get(0).getChildObject().get(1).translateObject(-0.12f, 0.12f, 0.0f);

* Telinga Dalam (Child Kepala)

Pada bagian telinga dalam ini, saya juga menggunakan objek cone seperti pada telinga dengan warna kuning. Tranformasi yang saya gunakan juga sama dengan transformasi telinga yaitu scale, translate, dan rotate. Namun perbedaannya ada pada warna dan metode scale nya. Untuk metode scale nya saya menggunakan value yang lebih kecil agar tampak seperti telinga yang lebih kecil.

//Right

objects.get(0).getChildObject().get(2).scaleObject(0.007f, 0.007f, 0.002f);  
objects.get(0).getChildObject().get(2).rotateObject((*float*) Math.*toRadians*(90f), 1f, 0f, 0f);  
objects.get(0).getChildObject().get(2).rotateObject((*float*) Math.*toRadians*(-38f), 0f, 0f, 1f);  
objects.get(0).getChildObject().get(2).translateObject(0.1f, 0.099f, 0.005f);

//Left

objects.get(0).getChildObject().get(3).scaleObject(0.007f, 0.007f, 0.002f);  
objects.get(0).getChildObject().get(3).rotateObject((*float*) Math.*toRadians*(95f), 1f, 0f, 0f);  
objects.get(0).getChildObject().get(3).rotateObject((*float*) Math.*toRadians*(38f), 0f, 0f, 1f);  
objects.get(0).getChildObject().get(3).translateObject(-0.1f, 0.099f, 0.005f);

* Mata (Child Kepala)

//Right

objects.get(0).getChildObject().get(5).scaleObject(0.2f, 0.2f, 0.075f);  
objects.get(0).getChildObject().get(5).translateObject(0.033f, 0.02f, 0.09f);

//Left

objects.get(0).getChildObject().get(6).scaleObject(0.2f, 0.2f, 0.075f);  
objects.get(0).getChildObject().get(6).translateObject(-0.033f, 0.02f, 0.09f);

Pada bagian mata saya menggunakan objek sphere yang berwarna putih. Transformasi yang saya gunakan adalah scale dan translate. Scale saya gunakan terhadap sumbu x,y,z untuk menyesuaikan ukuran mata terhadap kepala agar proporsional. Kemudian untuk translate pada sumbu x,y,z saya gunakan untuk memposisikan mata di bagian kanan dan kiri kepala di tengah. Untuk mata posisi mata hanya tinggal mengubah plus dan minus dari translate di sumbu x.

* Pupil (Child Kepala)

//Right

objects.get(0).getChildObject().get(7).scaleObject(0.07f, 0.07f, 0.05f);  
objects.get(0).getChildObject().get(7).translateObject(0.025f, 0.015f, 0.1f);

//Left

objects.get(0).getChildObject().get(8).scaleObject(0.07f, 0.07f, 0.05f);  
objects.get(0).getChildObject().get(8).translateObject(-0.025f, 0.015f, 0.1f);

Pada bagian pupil, saya juga menggunakan objek sphere dengan warna hitam. Transformasi yang saya gunakan adalah scale dan translate. Scale pada sumbu x,y,z saya gunakan untuk menyesuaikan ukuran pupil agar sesuai dengan bagian mata. Sedangkan translate pada sumbu x,y,z saya gunakan untuk memposisikan pupil di tengah mata. Pupil kiri dan kanan hanya dibedakan pada plus atau minus bagian translasi sumbu x.

* Hidung (Child Kepala)

Pada bagian hidung saya menggunakan objek sphere dengan warna coklat. Transformasi yang saya gunakan adalah scale dan translate. Scale pada sumbu x,y,z saya gunakan untuk mengubah dan memperkecil ukuran hidung sehingga kecil dan agak lonjong. Sedangkan translasi pada sumbu y,z saya gunakan untuk memposisikan hidung di tengah mata dan sedikit kebawah.

objects.get(0).getChildObject().get(4).scaleObject(0.07f, 0.05f, 0.075f);  
objects.get(0).getChildObject().get(4).translateObject(0.0f, -0.02f, 0.1f);

* Kumis (Child Kepala)

Pada bagian kumis saya menggunakan objek balok berwarna hitam dengan efek transparan. Transformasi yang saya gunakan adalah adalah rotate, scale, dan translate. Scale terhadap sumbu x,y,z saya gunakan untuk menipiskan dan memendekkan balok sehingga terlihat seperti kumis. Rotate terhadap sumbu z saya gunakan untuk memiringkan balok sehingga tidak terlihat lurus dan jadi sedikit miring ke bawah dan ke atas. Kemudian translate pada sumbu x,y,z saya gunakan untuk memposisikan dua kumis di sebelah kiri dan dua kumis di sebelah kanan.

//Left

objects.get(0).getChildObject().get(9).scaleObject(0.7f, 0.05f, 0.05f);  
objects.get(0).getChildObject().get(9).translateObject(-0.13f, 0.015f, 0.06f);  
objects.get(0).getChildObject().get(9).rotateObject((*float*) Math.*toRadians*(30f), 0f, 0f, 1f);

objects.get(0).getChildObject().get(10).scaleObject(0.7f, 0.05f, 0.05f);  
objects.get(0).getChildObject().get(10).translateObject(-0.13f, 0.01f, 0.06f);  
objects.get(0).getChildObject().get(10).rotateObject((*float*) Math.*toRadians*(0f), 0f, 0f, 1f);

//Right

objects.get(0).getChildObject().get(11).scaleObject(0.7f, 0.05f, 0.05f);  
objects.get(0).getChildObject().get(11).translateObject(0.13f, 0.015f, 0.06f);  
objects.get(0).getChildObject().get(11).rotateObject((*float*) Math.*toRadians*(-30f), 0f, 0f, 1f);

objects.get(0).getChildObject().get(12).scaleObject(0.7f, 0.05f, 0.05f);  
objects.get(0).getChildObject().get(12).translateObject(0.13f, 0.01f, 0.06f);  
objects.get(0).getChildObject().get(12).rotateObject((*float*) Math.*toRadians*(0f), 0f, 0f, 1f);

* Badan (Parent 2)

Badan merupakan object parent ke dua. Pada bagian badan saya menggunakan object sphere berwarna hitam. Transformasi yang saya gunakan adalah scale dan translate. Scale terhadap sumbu x,y,z saya gunakan untuk merubah sphere menjadi agak lonjong ke bawah menyerupai badan. Kemudian translasi terhadap sumbu y saya gunakan untuk memposisikan object di bawah kepala.

objects.get(1).scaleObject(0.6f, 0.75f, 0.5f);  
objects.get(1).translateObject(0.0f, -0.14f, 0.0f);

* Kalung (Child Badan)

Pada bagian kalung saya menggunakan object setengah sphere yang berwarna biru. Transformasi yang saya gunakan adalah scale, translate, rotate. Scale saya gunakan pada sumbu x,y,z saya gunakan untuk menyesuaikan ukuran kalung sesuai dengan ukuran tubuh dan lebih pendek. Translasi pada pada sumbu y,z saya gunakan untuk memposisikan kalung di antara kepala dan badan. Rotasi saya gunakan pada sumbu X saya gunakan untuk memposisikan bagian datar ke dalam dan bagian lengkung keluar.

objects.get(1).getChildObject().get(0).scaleObject(0.55f, 0.07f, 0.3f);  
objects.get(1).getChildObject().get(0).translateObject(0.0f, -0.09f, 0.02f);  
objects.get(1).getChildObject().get(0).rotateObject((*float*) Math.*toRadians*(0f), 1f, 0f, 0f);

* Tangan (Child Badan)

//Right

objects.get(1).getChildObject().get(1).scaleObject(0.5f, 0.2f, 0.2f);  
objects.get(1).getChildObject().get(1).translateObject(0.06f, -0.12f, 0.0f);  
objects.get(1).getChildObject().get(1).rotateObject((*float*) Math.*toRadians*(20f), 0f, 0f, 1f);

//Left

objects.get(1).getChildObject().get(2).scaleObject(0.5f, 0.2f, 0.2f);  
objects.get(1).getChildObject().get(2).translateObject(-0.06f, -0.12f, 0.0f);  
objects.get(1).getChildObject().get(2).rotateObject((*float*) Math.*toRadians*(-20f), 0f, 0f, 1f);

Pada bagian tangan saya menggunakan object sphere berwarna hitam. Tranformasi yang saya gunakan adalah rotate, translate, scale. Scale pada sumbu x,y,z saya gunakan untuk mengecilkan sphere menjadi bentuk lonjong. Kemudian rotate pada sumbu z saya gunakan untuk memiringkan tangan kanan dan kiri. Lalu translate saya gunakan untuk memposisikan tangan di bagian kanan dan kiri badan.

* Kaki (Child Badan)

//Right

objects.get(1).getChildObject().get(4).scaleObject(0.2f, 0.4f, 0.2f);  
objects.get(1).getChildObject().get(4).translateObject(0.035f, -0.23f, 0.0f);

//Left

objects.get(1).getChildObject().get(3).scaleObject(0.2f, 0.4f, 0.2f);  
objects.get(1).getChildObject().get(3).translateObject(-0.035f, -0.23f, 0.0f);

Pada bagian kaki saya menggunakan object sphere berwarna hitam. Transformasi yang saya gunakan adalah translate, scale. Scale pada sumbu x,y,z saya gunakan untuk membuat sphere menjadi lonjong ke bawah. Translate pada sumbu x,y saya gunakan untuk memposisikan kaki di bawah badan.

* Ekor (Child Badan)

objects.get(1).getChildObject().get(5).translateObject(0.0f, 0.2f, 0.0f);  
objects.get(1).getChildObject().get(5).rotateObject((*float*) Math.*toRadians*(180f), 0f, 0f, 1f);  
objects.get(1).getChildObject().get(5).rotateObject((*float*) Math.*toRadians*(-90f), 0f, 1f, 0f);

Pada bagian ekor saya menggunakan object kurva berwarna hitam. Transformasi yang saya gunakan translate dan rotate. Rotate pada sumbu y dan z saya gunakan untuk memposisikan ekor menjadi tegak dan mengarah ke belakang. Translate pada saya sumbu y saya gunakan untuk meletakkan ekor di bagian bawah belakang badan.

1. Tuliskan rancangan animasi pada obyek 3D yang dibuat di no 1. Untuk mendapatkan animasi yang diinginkan, misalnya mengatur gerakan pada bagian tertentu. Buatlah daftar transformasi apa yang dibutuhkan untuk masing-masing obyek yang digabungkan. Implementasikan animasi tersebut pada obyek 3D yang dibuat.

* Kepala dan pupil menengok ke kanan dan ke kiri. Menggunakan rotasi terhadap sumbu Y. Kemudian saya menggunakan batas agar rotasi tidak berlebihan.

*if* (window.isKeyPressed(GLFW\_KEY\_Q)) { *//Kepala Kiri* countDegree++;  
 *if* (*HeadDegX*<=-35f)  
 *setHeadTurnX*(*false*);  
 *else  
 setHeadTurnX*(*true*);  
 *if*(*HeadTurnX*){  
 objects.get(0).rotateObject((*float*)Math.*toRadians*(-0.5f),0f,1f,0f);  
 *setHeadDegX*(-0.5f);  
 }  
  
 *if* (*EyeDegX*<=-5f)  
 *setEyeTurnX*(*false*);  
 *else  
 setEyeTurnX*(*true*);  
 *if*(*EyeTurnX*){  
 objects.get(0).getChildObject().get(7).rotateObject((*float*)Math.*toRadians*(-0.1f),0f,1f,0f);  
 objects.get(0).getChildObject().get(8).rotateObject((*float*)Math.*toRadians*(-0.1f),0f,1f,0f);  
 *setEyeDegX*(-0.1f);  
 }  
}  
*if* (window.isKeyPressed(GLFW\_KEY\_E)) { *//Kepala Kanan* countDegree++;  
 *if* (*HeadDegX*>=35f)  
 *setHeadTurnX*(*false*);  
 *else  
 setHeadTurnX*(*true*);  
 *if*(*HeadTurnX*){  
 objects.get(0).rotateObject((*float*)Math.*toRadians*(0.5f),0f,1f,0f);  
 *setHeadDegX*(0.5f);  
 }  
  
 *if* (*EyeDegX*>=5f)  
 *setEyeTurnX*(*false*);  
 *else  
 setEyeTurnX*(*true*);  
 *if*(*EyeTurnX*){  
 objects.get(0).getChildObject().get(7).rotateObject((*float*)Math.*toRadians*(0.1f),0f,1f,0f);  
 objects.get(0).getChildObject().get(8).rotateObject((*float*)Math.*toRadians*(0.1f),0f,1f,0f);  
 *setEyeDegX*(0.1f);  
 }  
}

* Kepala dan pupil menengok ke atas dan bawah. Menggunakan rotasi terhadap sumbu x. Kemudian saya menggunakan batas agar rotasi tidak berlebihan.

*if* (window.isKeyPressed(GLFW\_KEY\_F)) { *//Kepala Atas* countDegree++;  
 *if* (*HeadDegY*>=20f)  
 *setHeadTurnY*(*false*);  
 *else  
 setHeadTurnY*(*true*);  
 *if*(*HeadTurnY*){  
 objects.get(0).rotateObject((*float*)Math.*toRadians*(0.5f),1f,0f,0f);  
 *setHeadDegY*(0.5f);  
 }  
  
 *if* (*EyeDegY*>=5f)  
 *setEyeTurnY*(*false*);  
 *else  
 setEyeTurnY*(*true*);  
 *if*(*EyeTurnY*){  
 objects.get(0).getChildObject().get(7).rotateObject((*float*)Math.*toRadians*(0.2f),1f,0f,0f);  
 objects.get(0).getChildObject().get(8).rotateObject((*float*)Math.*toRadians*(0.2f),1f,0f,0f);  
 *setEyeDegY*(0.2f);  
 }  
}  
*if* (window.isKeyPressed(GLFW\_KEY\_R)) { *//Kepala Bawah* countDegree++;  
 *if* (*HeadDegY*<=-90f)  
 *setHeadTurnY*(*false*);  
 *else  
 setHeadTurnY*(*true*);  
 *if*(*HeadTurnY*){  
 objects.get(0).rotateObject((*float*)Math.*toRadians*(-0.5f),1f,0f,0f);  
 *setHeadDegY*(-0.5f);  
 }  
  
 *if* (*EyeDegY*<=-5f)  
 *setEyeTurnY*(*false*);  
 *else  
 setEyeTurnY*(*true*);  
 *if*(*EyeTurnY*){  
 objects.get(0).getChildObject().get(7).rotateObject((*float*)Math.*toRadians*(-0.2f),1f,0f,0f);  
 objects.get(0).getChildObject().get(8).rotateObject((*float*)Math.*toRadians*(-0.2f),1f,0f,0f);  
 *setEyeDegY*(-0.2f);  
 }  
}

* Badan bergerak ke kanan dan ke kiri. Menggunakan rotasi terhadap sumbu y. Kemudian saya menggunakan batas agar rotasi tidak berlebihan.

*if* (window.isKeyPressed(GLFW\_KEY\_D)) { *//Badan Kanan* countDegree++;  
 *if* (*BodyDeg*>=35f)  
 *setBodyTurn*(*false*);  
 *else  
 setBodyTurn*(*true*);  
 *if*(*BodyTurn*){  
 objects.get(1).rotateObject((*float*)Math.*toRadians*(0.5f),0f,1f,0f);  
 *setBodyDeg*(0.5f);  
 }  
}  
*if* (window.isKeyPressed(GLFW\_KEY\_A)) { *//Badan Kiri* countDegree++;  
 *if* (*BodyDeg*<=-35f)  
 *setBodyTurn*(*false*);  
 *else  
 setBodyTurn*(*true*);  
 *if*(*BodyTurn*){  
 objects.get(1).rotateObject((*float*)Math.*toRadians*(-0.5f),0f,1f,0f);  
 *setBodyDeg*(-0.5f);  
 }  
}

* Tangan bergerak ke atas dan ke bawah. Menggunakan rotasi dan translasi terhadap sumbu z. Kemudian saya menggunakan batas agar rotasi tidak berlebihan. Lalu saya menggunakan rotate untuk menjaga pergerakan tangan jika badan dirotasi.

*if* (window.isKeyPressed(GLFW\_KEY\_W)) { *//Tangan Atas* countDegree++;  
 objects.get(1).getChildObject().get(1).rotateObject((*float*)Math.*toRadians*(-*yBadan*),0f,1f,0f);  
 objects.get(1).getChildObject().get(2).rotateObject((*float*)Math.*toRadians*(-*yBadan*),0f,1f,0f);  
  
 *if* (*HandDegY* == 0f)  
 *setHandTurnY*(*false*);  
 *else  
 setHandTurnY*(*true*);  
 *if*(*HandTurnY*){  
 *List*<Float> tangan = *new* ArrayList<>(objects.get(1).getCenterPoint());  
 objects.get(1).getChildObject().get(1).translateObject(tangan.get(0) \* -1, tangan.get(1) \* -1, tangan.get(2) \* -1);  
 objects.get(1).getChildObject().get(1).rotateObject((*float*) Math.*toRadians*(0.5f), 0.0f, 0.0f, 1.0f);  
 objects.get(1).getChildObject().get(1).translateObject(tangan.get(0) \* 1, tangan.get(1) \* 1, tangan.get(2) \* 1);  
  
 objects.get(1).getChildObject().get(2).translateObject(tangan.get(0) \* -1, tangan.get(1) \* -1, tangan.get(2) \* -1);  
 objects.get(1).getChildObject().get(2).rotateObject((*float*) Math.*toRadians*(-0.5f), 0.0f, 0.0f, 1.0f);  
 objects.get(1).getChildObject().get(2).translateObject(tangan.get(0) \* 1, tangan.get(1) \* 1, tangan.get(2) \* 1);  
 *setHandDegY*(-0.5f);  
 }  
  
 objects.get(1).getChildObject().get(1).rotateObject((*float*)Math.*toRadians*(*yBadan*),0f,1f,0f);  
 objects.get(1).getChildObject().get(2).rotateObject((*float*)Math.*toRadians*(*yBadan*),0f,1f,0f);  
}  
  
*if* (window.isKeyPressed(GLFW\_KEY\_S)) { *//Tangan Bawah* countDegree++;  
 objects.get(1).getChildObject().get(1).rotateObject((*float*)Math.*toRadians*(-*yBadan*),0f,1f,0f);  
 objects.get(1).getChildObject().get(2).rotateObject((*float*)Math.*toRadians*(-*yBadan*),0f,1f,0f);  
  
 *if* (*HandDegY* >= 20f)  
 *setHandTurnY*(*false*);  
 *else  
 setHandTurnY*(*true*);  
 *if*(*HandTurnY*){  
 *List*<Float> tangan = *new* ArrayList<>(objects.get(1).getCenterPoint());  
 objects.get(1).getChildObject().get(1).translateObject(tangan.get(0) \* -1, tangan.get(1) \* -1, tangan.get(2) \* -1);  
 objects.get(1).getChildObject().get(1).rotateObject((*float*) Math.*toRadians*(-0.5f), 0.0f, 0.0f, 1.0f);  
 objects.get(1).getChildObject().get(1).translateObject(tangan.get(0) \* 1, tangan.get(1) \* 1, tangan.get(2) \* 1);  
  
 objects.get(1).getChildObject().get(2).translateObject(tangan.get(0) \* -1, tangan.get(1) \* -1, tangan.get(2) \* -1);  
 objects.get(1).getChildObject().get(2).rotateObject((*float*) Math.*toRadians*(0.5f), 0.0f, 0.0f, 1.0f);  
 objects.get(1).getChildObject().get(2).translateObject(tangan.get(0) \* 1, tangan.get(1) \* 1, tangan.get(2) \* 1);  
 *setHandDegY*(+0.5f);  
 }  
  
  
 objects.get(1).getChildObject().get(1).rotateObject((*float*)Math.*toRadians*(*yBadan*),0f,1f,0f);  
 objects.get(1).getChildObject().get(2).rotateObject((*float*)Math.*toRadians*(*yBadan*),0f,1f,0f);  
}

* Tangan bergerak menutup dan membuka. Menggunakan rotasi dan translasi dengan menggunakan sumbu y. Kemudian saya menggunakan batas agar rotasi tidak berlebihan. Lalu saya menggunakan rotate untuk menjaga pergerakan tangan jika badan dirotasi.

*if* (window.isKeyPressed(GLFW\_KEY\_V)) { *//Tangan Tutup* countDegree++;  
 objects.get(1).getChildObject().get(1).rotateObject((*float*)Math.*toRadians*(-*yBadan*),0f,1f,0f);  
 objects.get(1).getChildObject().get(2).rotateObject((*float*)Math.*toRadians*(-*yBadan*),0f,1f,0f);  
 *if* (*HandDegX* >= 80f)  
 *setHandTurnX*(*false*);  
 *else  
 setHandTurnX*(*true*);  
 *if*(*HandTurnX*){  
 *List*<Float> tangan = *new* ArrayList<>(objects.get(1).getCenterPoint());  
 objects.get(1).getChildObject().get(1).translateObject(tangan.get(0) \* -1, tangan.get(1) \* -1, tangan.get(2) \* -1);  
 objects.get(1).getChildObject().get(1).rotateObject((*float*) Math.*toRadians*(-0.5f), 0.0f, 1.0f, 0.0f);  
 objects.get(1).getChildObject().get(1).translateObject(tangan.get(0) \* 1, tangan.get(1) \* 1, tangan.get(2) \* 1);  
  
 objects.get(1).getChildObject().get(2).translateObject(tangan.get(0) \* -1, tangan.get(1) \* -1, tangan.get(2) \* -1);  
 objects.get(1).getChildObject().get(2).rotateObject((*float*) Math.*toRadians*(0.5f), 0.0f, 1.0f, 0.0f);  
 objects.get(1).getChildObject().get(2).translateObject(tangan.get(0) \* 1, tangan.get(1) \* 1, tangan.get(2) \* 1);  
 *setHandDegX*(+0.5f);  
 }  
  
  
 objects.get(1).getChildObject().get(1).rotateObject((*float*)Math.*toRadians*(*yBadan*),0f,1f,0f);  
 objects.get(1).getChildObject().get(2).rotateObject((*float*)Math.*toRadians*(*yBadan*),0f,1f,0f);  
}  
  
*if* (window.isKeyPressed(GLFW\_KEY\_B)) { *//Tangan Buka* countDegree++;  
 objects.get(1).getChildObject().get(1).rotateObject((*float*)Math.*toRadians*(-*yBadan*),0f,1f,0f);  
 objects.get(1).getChildObject().get(2).rotateObject((*float*)Math.*toRadians*(-*yBadan*),0f,1f,0f);  
  
 *if* (*HandDegX* == 0f)  
 *setHandTurnX*(*false*);  
 *else  
 setHandTurnX*(*true*);  
 *if*(*HandTurnX*){  
 *List*<Float> tangan = *new* ArrayList<>(objects.get(1).getCenterPoint());  
 objects.get(1).getChildObject().get(1).translateObject(tangan.get(0) \* -1, tangan.get(1) \* -1, tangan.get(2) \* -1);  
 objects.get(1).getChildObject().get(1).rotateObject((*float*) Math.*toRadians*(0.5f), 0.0f, 1.0f, 0.0f);  
 objects.get(1).getChildObject().get(1).translateObject(tangan.get(0) \* 1, tangan.get(1) \* 1, tangan.get(2) \* 1);  
  
 objects.get(1).getChildObject().get(2).translateObject(tangan.get(0) \* -1, tangan.get(1) \* -1, tangan.get(2) \* -1);  
 objects.get(1).getChildObject().get(2).rotateObject((*float*) Math.*toRadians*(-0.5f), 0.0f, 1.0f, 0.0f);  
 objects.get(1).getChildObject().get(2).translateObject(tangan.get(0) \* 1, tangan.get(1) \* 1, tangan.get(2) \* 1);  
 *setHandDegX*(-0.5f);  
 }

* Kaki bergerak keatas dan kebawah. Menggunakan rotasi dan translasi dengan menggunakan sumbu x. Kemudian saya menggunakan batas agar rotasi tidak berlebihan. Lalu saya menggunakan rotate untuk menjaga pergerakan tangan jika badan dirotasi.

*if* (window.isKeyPressed(GLFW\_KEY\_G)) { *//Kaki Mundur* countDegree++;  
 objects.get(1).getChildObject().get(3).rotateObject((*float*)Math.*toRadians*(-*yBadan*),0f,1f,0f);  
 objects.get(1).getChildObject().get(4).rotateObject((*float*)Math.*toRadians*(-*yBadan*),0f,1f,0f);  
  
 *if* (*LegDeg* == 0f)  
 *setLegTurn*(*false*);  
 *else  
 setLegTurn*(*true*);  
 *if*(*LegTurn*){  
 *List*<Float> kaki = *new* ArrayList<>(objects.get(1).getCenterPoint());  
 objects.get(1).getChildObject().get(3).translateObject(kaki.get(0) \* -1, kaki.get(1) \* -1, kaki.get(2) \* -1);  
 objects.get(1).getChildObject().get(3).rotateObject((*float*) Math.*toRadians*(0.5f), 1.0f, 0.0f, 0.0f);  
 objects.get(1).getChildObject().get(3).translateObject(kaki.get(0) \* 1, kaki.get(1) \* 1, kaki.get(2) \* 1);  
  
 objects.get(1).getChildObject().get(4).translateObject(kaki.get(0) \* -1, kaki.get(1) \* -1, kaki.get(2) \* -1);  
 objects.get(1).getChildObject().get(4).rotateObject((*float*) Math.*toRadians*(0.5f), 1.0f, 0.0f, 0.0f);  
 objects.get(1).getChildObject().get(4).translateObject(kaki.get(0) \* 1, kaki.get(1) \* 1, kaki.get(2) \* 1);  
 *setLegDeg*(+0.5f);  
 }  
  
  
  
 objects.get(1).getChildObject().get(3).rotateObject((*float*)Math.*toRadians*(*yBadan*),0f,1f,0f);  
 objects.get(1).getChildObject().get(4).rotateObject((*float*)Math.*toRadians*(*yBadan*),0f,1f,0f);  
}  
*if* (window.isKeyPressed(GLFW\_KEY\_T)) { *//Kaki Maju* countDegree++;  
 objects.get(1).getChildObject().get(3).rotateObject((*float*)Math.*toRadians*(-*yBadan*),0f,1f,0f);  
 objects.get(1).getChildObject().get(4).rotateObject((*float*)Math.*toRadians*(-*yBadan*),0f,1f,0f);  
  
 *if* (*LegDeg*<=-60f)  
 *setLegTurn*(*false*);  
 *else  
 setLegTurn*(*true*);  
 *if*(*LegTurn*){  
 *List*<Float> kaki = *new* ArrayList<>(objects.get(1).getCenterPoint());  
 objects.get(1).getChildObject().get(3).translateObject(kaki.get(0) \* -1, kaki.get(1) \* -1, kaki.get(2) \* -1);  
 objects.get(1).getChildObject().get(3).rotateObject((*float*) Math.*toRadians*(-0.5f), 1.0f, 0.0f, 0.0f);  
 objects.get(1).getChildObject().get(3).translateObject(kaki.get(0) \* 1, kaki.get(1) \* 1, kaki.get(2) \* 1);  
  
 objects.get(1).getChildObject().get(4).translateObject(kaki.get(0) \* -1, kaki.get(1) \* -1, kaki.get(2) \* -1);  
 objects.get(1).getChildObject().get(4).rotateObject((*float*) Math.*toRadians*(-0.5f), 1.0f, 0.0f, 0.0f);  
 objects.get(1).getChildObject().get(4).translateObject(kaki.get(0) \* 1, kaki.get(1) \* 1, kaki.get(2) \* 1);  
 *setLegDeg*(-0.5f);  
 }

* Objek berjalan maju. Menggunakan translasi dan rotasi terhdap sumbu y dan x. Kemudian saya menggunakan batas agar rotasi tidak berlebihan.

*if* (window.isKeyPressed(GLFW\_KEY\_J)){ *//Karakter Berjalan* objects.get(1).getChildObject().get(3).rotateObject((*float*)Math.*toRadians*(-*yBadan*),0f,1f,0f);  
 objects.get(1).getChildObject().get(4).rotateObject((*float*)Math.*toRadians*(-*yBadan*),0f,1f,0f);  
 objects.get(0).rotateObject((*float*)Math.*toRadians*(-*yBadan*),0f,1f,0f);  
 objects.get(1).rotateObject((*float*)Math.*toRadians*(-*yBadan*),0f,1f,0f);  
  
 *float* degree = 0.15f;  
 *if* (walkDeg >= 15f){  
 direction \*= -1;  
 walkDeg = 0;  
 }  
 objects.get(0).rotateObject((*float*)Math.*toRadians*(-*yBadan*),0f,1f,0f);  
  
 objects.get(1).getChildObject().get(3).rotateObject((*float*)Math.*toRadians*(direction \*degree),1f,0f,0f);  
 objects.get(1).getChildObject().get(4).rotateObject((*float*)Math.*toRadians*(direction \*-degree),1f,0f,0f);  
 objects.get(0).translateObject(0.0f, 0.0f, 0.001f);  
 objects.get(1).translateObject(0.0f, 0.0f, 0.001f);  
  
 objects.get(0).rotateObject((*float*)Math.*toRadians*(*yBadan*),0f,1f,0f);  
 walkDeg +=degree;  
  
 objects.get(1).getChildObject().get(3).rotateObject((*float*)Math.*toRadians*(*yBadan*),0f,1f,0f);  
 objects.get(1).getChildObject().get(4).rotateObject((*float*)Math.*toRadians*(*yBadan*),0f,1f,0f);  
 objects.get(0).rotateObject((*float*)Math.*toRadians*(*yBadan*),0f,1f,0f);  
 objects.get(1).rotateObject((*float*)Math.*toRadians*(*yBadan*),0f,1f,0f);  
 }  
}

1. Pada obyek 3D yang sudah merupakan gabungan (dengan transformasi individu yg ada), transformasikan keseluruhan obyek dengan translasi, rotasi dan skala.

//Transformasi Skala

Transformasi scale menggunakan method scale dan agar semua bagian bisa menjadi besar secara bersamaan maka scale akan terjadi pada ketiga sumbu x,y,z. Jika ingin memperkecil maka menggunakan nilai <1, jika ingin memperbesar gunakan nilai >1.

*if* (window.isKeyPressed(GLFW\_KEY\_L)) { *//Scale Kecil* objects.get(0).scaleObject(0.99f, 0.99f, 0.99f);  
 objects.get(1).scaleObject(0.99f, 0.99f, 0.99f);  
}  
*if* (window.isKeyPressed(GLFW\_KEY\_P)) { *//Scale Besar* objects.get(0).scaleObject(1.01f, 1.01f, 1.01f);  
 objects.get(1).scaleObject(1.01f, 1.01f, 1.01f);  
}

//Transformasi Rotasi

Transformasi rotasi menggunakan method rotate. Jika ingin merotate ke atas dan kebawah maka menggunakan sumbu x. Jika ingin merotate ke kanan dan ke kiri maka menggunakan sumbu y.

*if* (window.isKeyPressed(GLFW\_KEY\_X)) { *//Rotate Keatas* objects.get(0).rotateObject((*float*)Math.*toRadians*(0.5f),1f,0f,0f);  
 objects.get(1).rotateObject((*float*)Math.*toRadians*(0.5f),1f,0f,0f);  
}  
*if* (window.isKeyPressed(GLFW\_KEY\_Y)) { *//Rotate Kebawah* objects.get(0).rotateObject((*float*)Math.*toRadians*(-0.5f),1f,0f,0f);  
 objects.get(1).rotateObject((*float*)Math.*toRadians*(-0.5f),1f,0f,0f);  
}

*if* (window.isKeyPressed(GLFW\_KEY\_C)) { *//Rotasi Kanan* countDegree++;  
 objects.get(0).rotateObject((*float*) Math.*toRadians*(0.5f),0.0f,1.0f,0.0f);  
 objects.get(1).rotateObject((*float*) Math.*toRadians*(0.5f),0.0f,1.0f,0.0f);  
 *setyBadan*(0.5f);  
}  
*if* (window.isKeyPressed(GLFW\_KEY\_Z)) { *//Rotasi Kiri* countDegree++;  
 objects.get(0).rotateObject((*float*) Math.*toRadians*(-0.5f),0.0f,1.0f,0.0f);  
 objects.get(1).rotateObject((*float*) Math.*toRadians*(-0.5f),0.0f,1.0f,0.0f);  
 *setyBadan*(-0.5f);  
}

//Transformasi Translasi

Transformasi ini menggunakan method translate. Translasi Kanan dan kiri menggunakan sumbu x.

*if* (window.isKeyPressed(GLFW\_KEY\_O)) { *//Translasi Kanan* objects.get(0).translateObject(0.001f,0.0f,0.0f);  
 objects.get(1).translateObject(0.001f,0.0f,0.00f);  
}  
*if* (window.isKeyPressed(GLFW\_KEY\_I)) { *//Translasi Kiri* objects.get(0).translateObject(-0.001f,0.0f,0.0f);  
 objects.get(1).translateObject(-0.001f,0.0f,0.00f);  
}