**LAPORAN GRAFIKA KOMPUTER**

**KELOMPOK 22**

****

**Disusun oleh : Kelompok 22**

Jason Kristanto - c14210188

Christopher Joseph - c14210221

Joshua Briantama - c14210005

**MATA KULIAH GRAFIKA KOMPUTER**

1. **Pembagian Tugas**

Christopher Joseph C14210221

Environment: rumah, landscape, cewek.   
Lighting, senter

Story

Joshua Briantama - c14210005

Environment: pohon, batu, sumur, kayu salib

Collision

Bug fixing

Jason Kristanto c14210188

Environment : main character, kuda, anjing, rusa, awan

Camera

Editing

1. **Story**

Seorang bocah laki-laki tiba-tiba terbangun dan menemukan dirinya dalam suatu peternakan. Dia tidak mengerti bagaimana dia bisa berada di peternakan tersebut. Dia menyadari bahwa dia sendirian dan terjebak. Dengan bermodalkan senter yang

dia temukan, dia mulai menjelajahi peternakan yang angker tersebut.

Saat dia menjelajahi peternakan tersebut, anak lelaki itu menemukan rumah dan sumur tua yang sudah tidak digunakan. Namun anehnya, masih terdapat beberapa hewan disana. Dia mendekati salah satu hewannya dan mengecek apakah hewan tersebut masih hidup. Ternyata, hewan-hewan tersebut semuanya masih hidup. Tiba-tiba, dia mendengar nafas kuda dan bisikan, “bahan makanan tidak boleh kemana-mana”. Seketika itu, bocah tersebut lari ke belakang rumahnya.

Ternyata, disana malah menemukan kuburan yang terdapat mayat manusia di atasnya. Seketika dia menyadari bahwa hewan-hewan di peternakan tersebutlah yang menculik dia dan mau membunuhnya untuk dimakan.

Bocah tersebut lari dan berusaha mencari jalan keluar dari peternakan tersebut namun tanpa solusi. Pada akhirnya, bocah tersebut memutuskan untuk lari ke kolam kemudian ia mendengar suara yang menyuruhnya untuk masuk ke kolam untuk bunuh diri daripada menjadi makanan untuk hewan-hewan ternak di peternakan tersebut.

*Eksekusi singkat story:*

Jadi nanti awalnya ada bocah yang terbangun di peternakannya itu di sebelah cottage. Nah nanti bocahnya jalan-jalan buat melakukan cek object2 yang ada sesuai story, dengan menggunakan senter. Objectnya didatangi satu per satu menggunakan spotlight dari senter. Lingkungan di farm dibuat gelap karena harinya sudah malam. Lalu bocah mendekati satu kuda dan mendekat ke badannya. Disana kuda lalu ngomong, dan bocahnya langsung lari ke belakang rumah. Waktu lari, dia keliatan ada mayat yang digantung di kuburannya. Setelah itu bocahnya lari lagi, dan akhirnya bunuh diri ke kolam.

1. **Environment**

Proses pembuatan program ini dimulai dengan pengaturan *environtment*. Kami menempatkan berbagai macam objek dalam program ini sehingga bisa menjadi sebuah kesatuan yang membentuk *environtment* yang menampilkan kondisi lingkungan peternakan. Tema yang kami ambil adalah *Haunted Farm.*

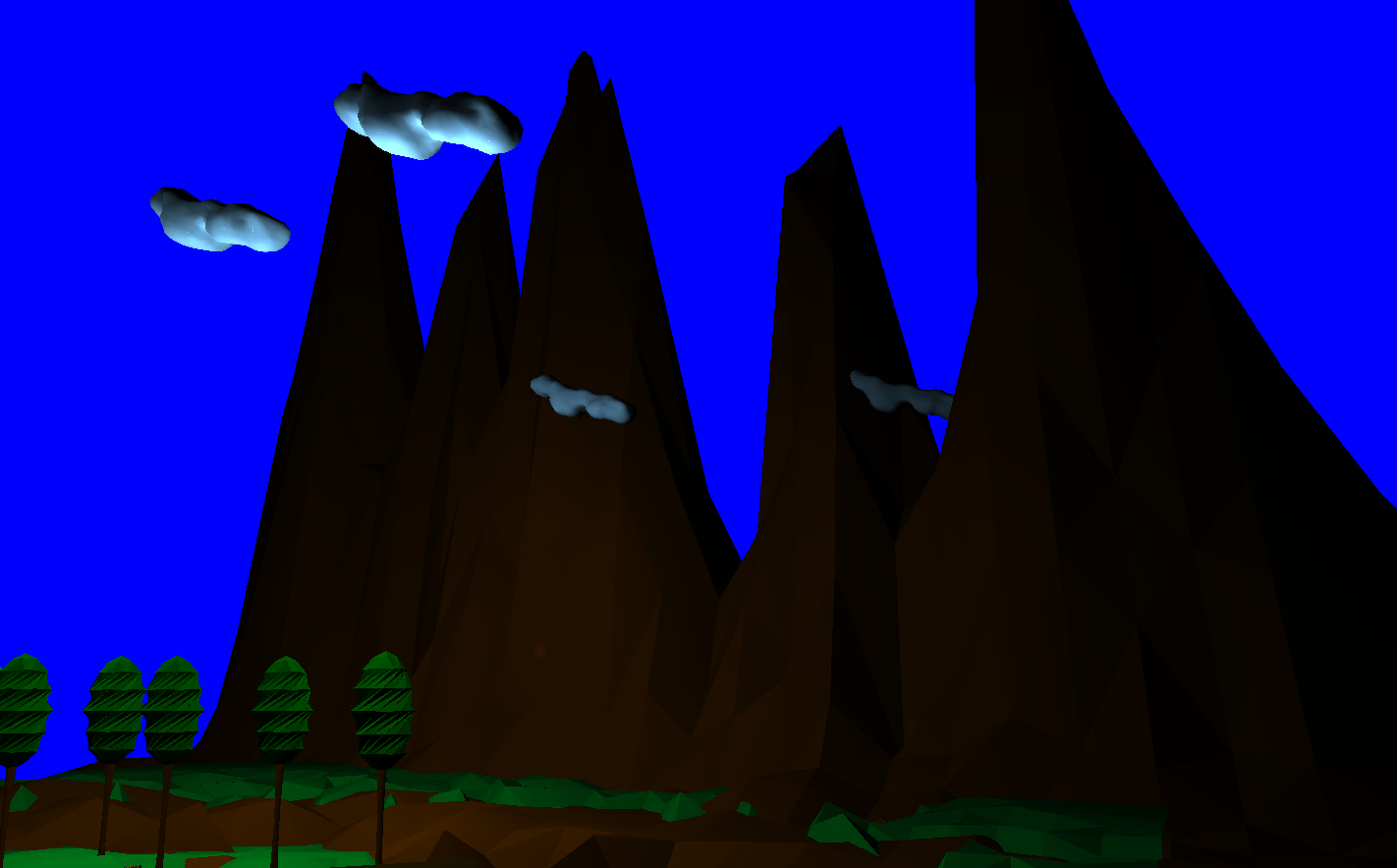


*Ini merupakan tampilan dari atas*

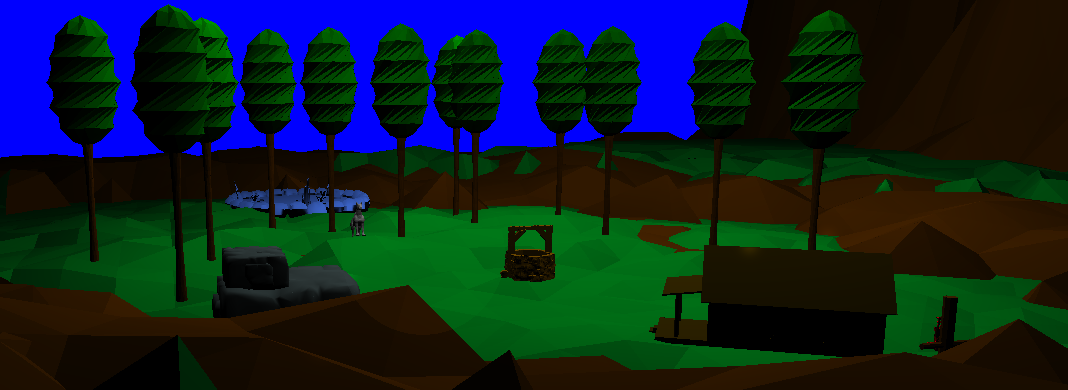
Karena tema kami adalah *Haunted Farm*, maka kami menciptakan lingkungan pertanian yang terdapat pohon-pohonan, rumah, landscape pegunungan, kolam, dan komponen-komponen pendukung lainnya.

Kami membuat objek-objek tersebut menggunakan objectloader dengan memanggil file dengan tipe file .fbx. Kemudian kami menentukan warna yang sesuai untuk objek tersebut. Kami juga mengatur skala translasi dan rotasi objek-objek tersebut sehingga bisa menyatu dengan baik dan bisa lebih dipahami.

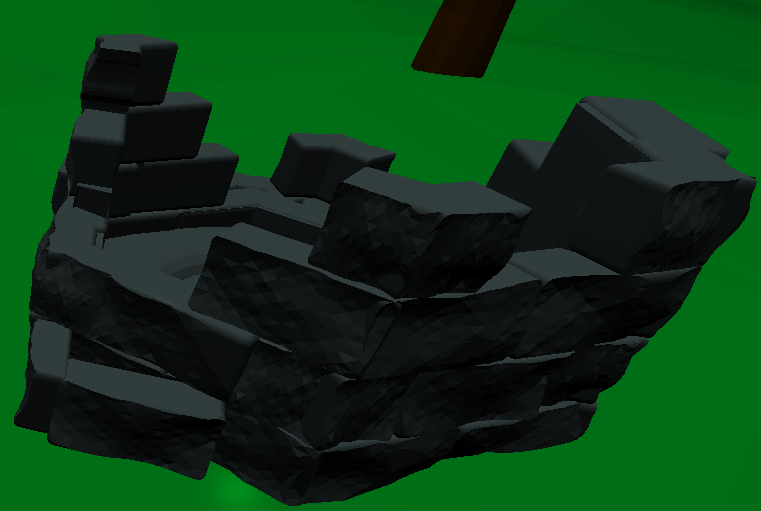
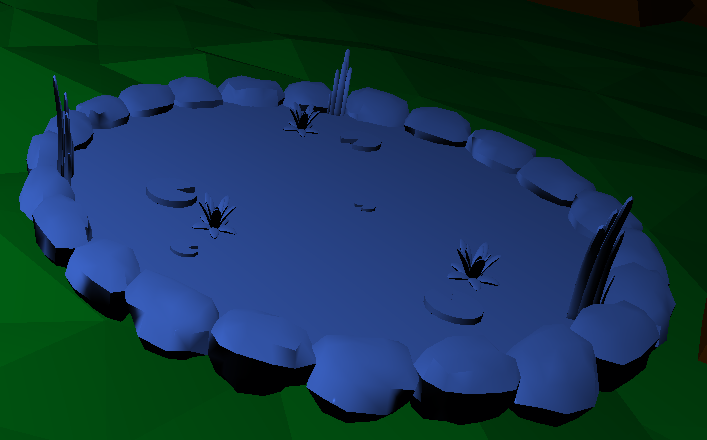
Objectnya yang kami buat dimulai dari object *LPTerrain* yang merupakan tanah berwarna coklat, termasuk pegunungannya.



Lalu terdapat object *rumput* yang menjadi object dasar rerumputan berwarna hijau untuk object utama traverse di environmentnya.

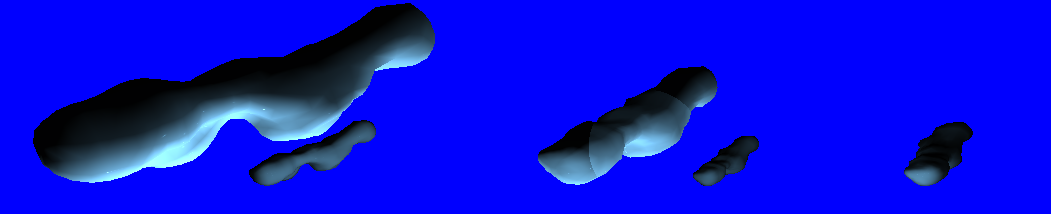


Selanjutnya, kami memasukkan beberapa object-object pendukung seperti rumah, sumur, kolam, batu, awan, gunung, dll. Object-object tersebut hanya diload biasa dan diatur warnanya.



Sedangkan untuk object yang ada banyak, kita menggunakan function untuk melakukan generate objectnya sehingga tidak perlu load satu per satu.

Contohnya adalah pohon dan awan



1. **Lighting dan Camera**

Lighting world nanti dibuat gelap, dan hanya ada light source di mayat, batu, dan kolam, dengan tujuan untuk menggambarkan suasana gelap yang mencekam. Bocah karakter utama membawa senter, yang menghasilkan light source untuk menerangi kemana dia jalan. Cahaya dari senter bisa dilihat di depan dari arah karakter bocah menghadap.

Cara menentukan posisi light source adalah dengan menaruh light source di koordinatnya. Lighting harus diset di 4 titik.

Vector3f[] \_pointLightPositions = {

new Vector3f(0f, 0f, 1f),

new Vector3f(4.0f,-7f,10f),

new Vector3f(-14.0f, -5f, -30f),

new Vector3f(5f,0f,25f)

};

Langit dibuat gelap dengan cara mengubah value diffusenya. Semakin kecil semakin gelap:

for (int i = 0; i < \_pointLightPositions.length; i++) {

uniformsMap.setUniform("pointLights[" + i + "].position", \_pointLightPositions[i]);

uniformsMap.setUniform("pointLights[" + i + "].ambient", new Vector3f(0.05f, 0.05f, 0.05f));

uniformsMap.setUniform("pointLights[" + i + "].diffuse", new Vector3f(0.5f, .5f, .5f));

uniformsMap.setUniform("pointLights[" + i + "].specular", new Vector3f(1.0f, 1.0f, 1.0f));

uniformsMap.setUniform("pointLights[" + i + "].constant", 1.0f);

uniformsMap.setUniform("pointLights[" + i + "].linear", 0.09f);

uniformsMap.setUniform("pointLights[" + i + "].quadratic", 0.032f);

}

Cara membuat senter:

Posisi dan directionnya diatur berdasarkan posisi dan direction camera. Lalu level terang senter berada di diffuse. Valuenya dapat diubah jadi 0 apabila ingin mematikan senter.

// senter

float camX =camera.getPosition().x;

float camY=camera.getPosition().y;

float camZ=camera.getPosition().z;

Vector3f senter = new Vector3f(camX, camY,camZ);

float camX1=camera.getDirection().x;

float camY2=camera.getDirection().y;

float camZ2=camera.getDirection().z;

Vector3f senterArah = new Vector3f(camX1, camY2,camZ2);

uniformsMap.setUniform("spotLight.position", senter);

uniformsMap.setUniform("spotLight.direction", senterArah);

uniformsMap.setUniform("spotLight.ambient", new Vector3f(0.0f, 0.0f, 0.0f));

uniformsMap.setUniform("spotLight.diffuse", new Vector3f(*senterA*, *senterA*, *senterA*));

uniformsMap.setUniform("spotLight.specular", new Vector3f(1.0f, 1.0f, 1.0f));

uniformsMap.setUniform("spotLight.constant", 1.0f);

uniformsMap.setUniform("spotLight.linear", 0.09f);

uniformsMap.setUniform("spotLight.quadratic", 0.032f);

uniformsMap.setUniform("spotLight.cutOff", (float) Math.*cos*(Math.*toRadians*(5f)));

uniformsMap.setUniform("spotLight.outerCutOff", (float) Math.*cos*(Math.*toRadians*(5f)));

uniformsMap.setUniform("viewPos", senter);

Camera kita buat menjadi third person, supaya karakter utama dapat terlihat ketika berjalan dan lingkungan sekitar terlihat lebih jelas. Camera mengikuti karakter ketika di gerakkan, dan apabila karakter melakukan rotasi, camera juga berotasi. Kamera bisa digerakkan dengan rotasi terhadap karakter utama sehingga bisa lebih leluasa dalam mengatur sudut pandang dan mengontrol arah gerak karakter.

Cara membuat gerakan maju third person

if (window.isKeyPressed(*GLFW\_KEY\_W*)) {

check\_collision(camera.getDirection().x, camera.getDirection().z,0);

if (batasX) {

camera.moveForward(move);

objects.get(11).translateObject(camera.getDirection().x, 0f, camera.getDirection().z);

posisi[0] += camera.getDirection().x;

posisi[2] += camera.getDirection().z;

collision\_cek();

}

}

Camera di majukan sebesar move lalu translate object sebesar direction kamera x dan direction kamera z lalu simpan posisi player menggunakan array posisi seperti gambar tersebut.

1. **Movement and Collision**

Pergerakan yang ada di program ini hanya bisa dilakukan pada karakter utama. Kami membuat karakter utama supaya bisa bergerak ke depan, belakang, kanan dan kiri. *Key W* untuk membuat karakter maju, *key A* untuk ke kiri, *key S* untuk mundur, dan *key D* untuk ke kanan. Selain itu *key Q* dan *key E* bisa digunakan untuk rotasi karakter beserta dengan rotasi kamera. Karakter bisa bergerak berdasarkan sudut kamera, sehingga saat menekan W setelah kamera dirotasi, maka karakter tetap akan bergerak ke depan dari arah kamera. Selain itu untuk collision, jika karakter bergerak menabrak objek, maka karakter akan berhenti sehingga tidak menembus objek.

Karakter maju

if (window.isKeyPressed(*GLFW\_KEY\_W*)) {

check\_collision(camera.getDirection().x, camera.getDirection().z,0);

if (batasX) {

camera.moveForward(move);

objects.get(11).translateObject(camera.getDirection().x, 0f, camera.getDirection().z);

posisi[0] += camera.getDirection().x;

posisi[2] += camera.getDirection().z;

collision\_cek();

}

}

Kamera dibuat maju kemudian karakter maju sesuai besaran dari jarak kamera. Kemudian posisi dari karakter juga diubah sesuai dengan perubahan jarak kamera sehingga saat nanti ada pergerakan lain, kamera dan karakter tetap bergerak bersama. Kemudian tiap pergerakan mengecek apakah terdapat collision.

Rotasi kamera dan karakter

if(window.isKeyPressed(*GLFW\_KEY\_Q*)){

camera.moveForward(5f);

camera.addRotation(0, (float) Math.*toRadians*(-0.7f));

objects.get(11).translateObject(-posisi[0],-posisi[1],-posisi[2]);

objects.get(11).rotateObject((float)Math.*toRadians*(0.7f),0f,1f,0f);

objects.get(11).translateObject(posisi[0], posisi[1], posisi[2]);

camera.moveBackwards(5f);

}

Selanjutnya untuk membuat kamera berotasi, menggunakan function yang memutar kamera kemudian memutar karakter dengan arah yang sama sehingga kamera dan karakter bisa menghadap arah yang sama saat rotasi

Collision menggunakan perhitungan jarak

public void check\_collision(float x, float y, int temp){

for (int i = 0; i< Collision.size(); i++){

double hypotenuse = Math.*sqrt*(Math.*pow*((Collision.get(i).x - (posisi[0] + x)),2) + Math.*pow*((Collision.get(i).z - (posisi[2] + y)),2));

if (hypotenuse < radius.get(i)){

if (temp == 0 & batasXrev){

batasX = false;

} else if (temp == 1 & batasX) {

batasXrev = false;

} else if (temp == 2 & batasZrev) {

batasZ = false;

} else if (temp == 3 & batasZ) {

batasZrev = false;

}

}

}

}

Menggunakan rumus phytagoras untuk mengecek jarak antar 2 titik nya sehingga jika jarak lebih kecil dari radius maka ada sebuah variable yang di false

Dan variable tersebut digunakan untuk pengecekan di input apabil falsa maka tidak bisa bergerak

Penjelasan movement sama seperti di atas bagian camera dan lighting