

LAPORAN PEMBUATAN OBJEK

Untuk Memenuhi Nilai Ujian Akhir Semester

Mata Kuliah Komputer Grafika Semester Ganjil Tahun Ajaran 2021/2022



Disusun oleh:

Jeffry Haryanto Gunawan C14180012

Terrius Jeremiah C14190105

Kevin Montoya C14190133

Dosen:

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

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNIK INDUSTRI

UNIVERSITAS KRISTEN PETRA

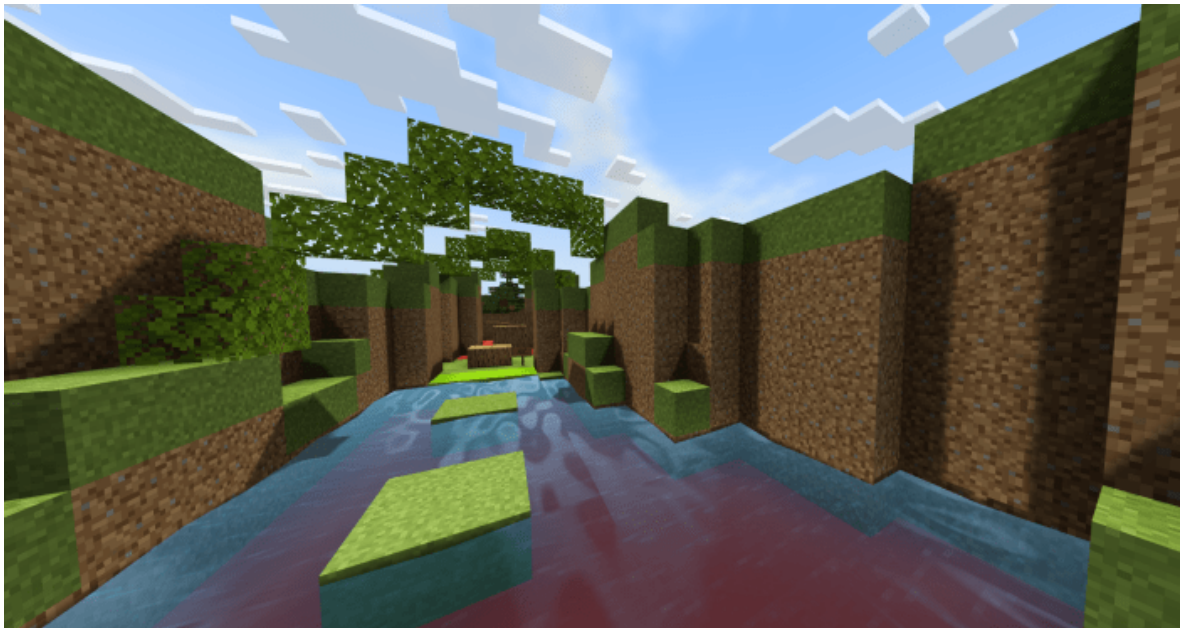
SURABAYA

I. Pendahuluan

Untuk memenuhi nilai Ujian Akhir Semester mata kuliah Komputer Grafika, kelompok kami mengambil ide dari game populer, yaitu Minecraft, dengan modifikasi tertentu dan menggunakan bahasa pemrograman C#. Tema utama dari proyek kami adalah “Minecraft Patrol Police” yang menceritakan patroli di sebuah lapangan luas dan kolam renang dari atas hingga kebawah setiap harinya.

II. Pembuatan Objek

Kami Bertiga memilih character minecraft karena populer. Kami sendiri terinspirasi dari minecraft parkour seperti gambar berikut :



Maka kami berusaha membuatnya dengan tema berbeda yaitu “Minecraft Patrol Police” dan bergerak secara otomatis mengelilingi map.

Realita yang kami buat :

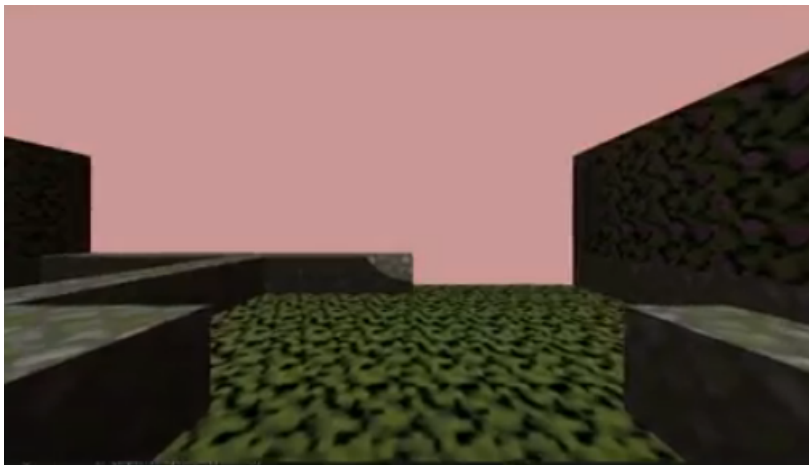
Mode Awal



Mode malam hari



Mode Pagi hari



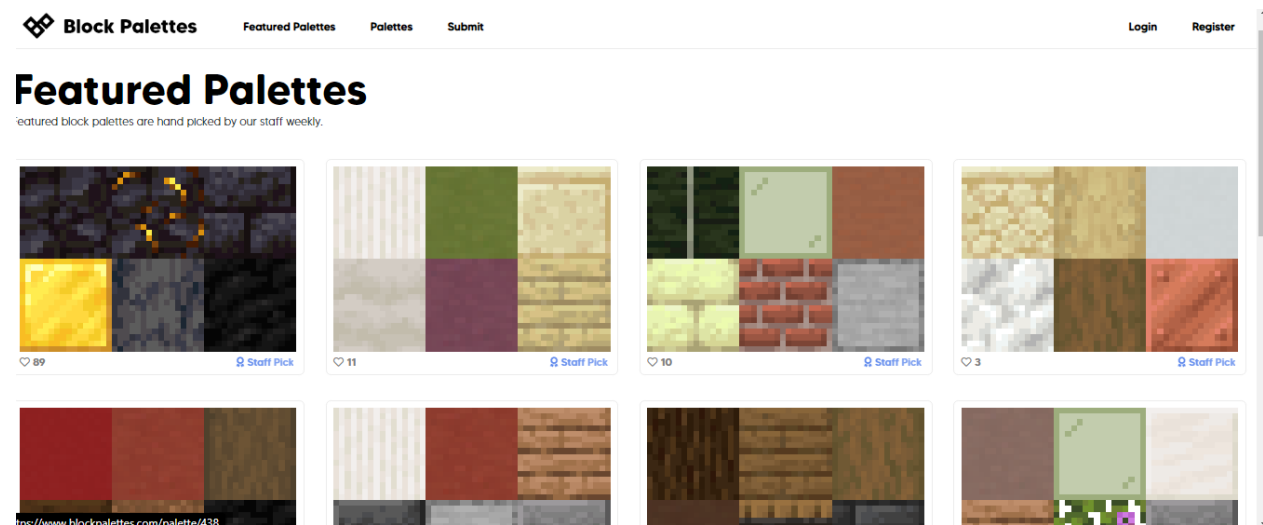
Mode melewati jembatan



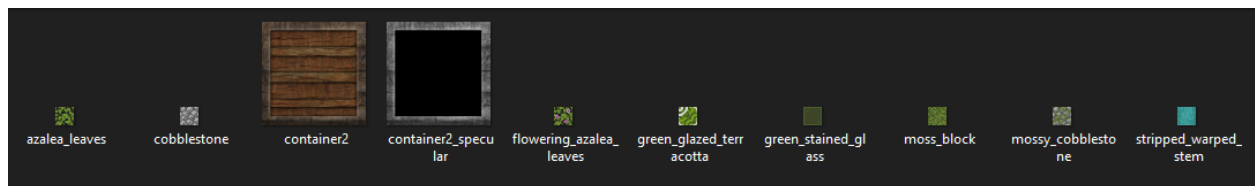
Mode Sore sambil melewati maze



Kami menggunakan block palletes untuk membangun warna kubus atau tampilan kubus yang berbeda, sumber referensi yang kami ambil dari <https://www.blockpalettes.com>.

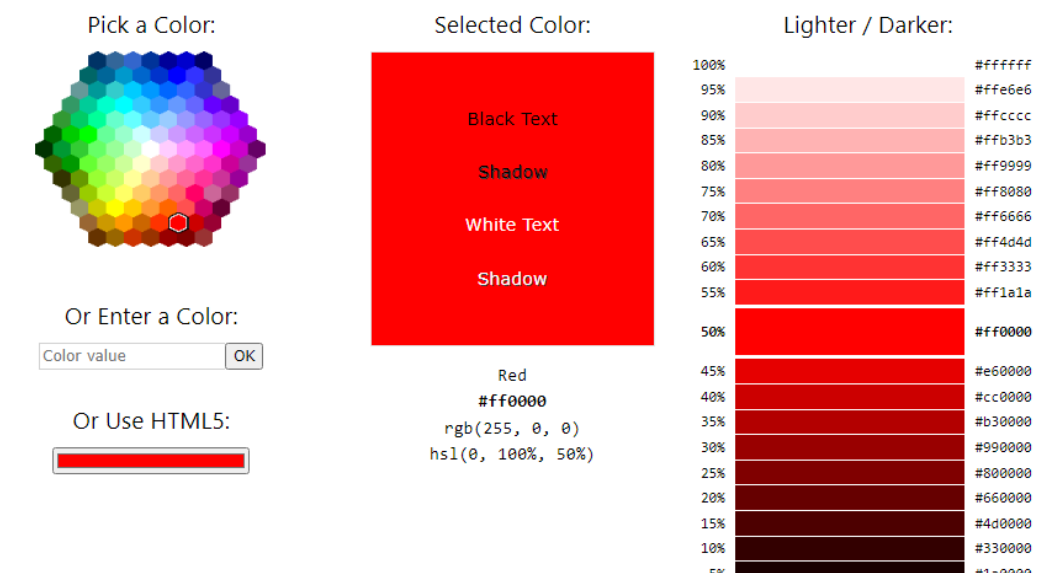


Dan implementasi yang akan kami gunakan adalah :



Kami menggunakan Color Picker dari w3school untuk mengatur warna background/langit.

https://www.w3schools.com/colors/colors_picker.asp?colorhex=ff0000



Pada fungsi onload, kami menentukan warna awal langit dan mengatur skin dari box yang sudah didownload dari internet

```
0 references
protected override void OnLoad()
{
    base.OnLoad();

    //Ganti warna background
    GL.ClearColor(0.5f, 0.5f, 1.0f, 1);
```

 -> GL.ClearColor untuk mengatur warna langit awal.

Code dibawah ini kami gunakan untuk mengatur skin dari setiap kelompok bagian kubus (cth: tangga, jembatan,dll).

```
_object3d[0].load2(Size.X, Size.Y, "cobblestone.png", "container2_specular.png");
_object3d[1].load2(Size.X, Size.Y, "container2.png", "container2_specular.png");
_object3d[2].load2(Size.X, Size.Y, "azalea_leaves.png", "container2_specular.png");
_object3d[3].load2(Size.X, Size.Y, "mossy_cobblestone.png", "container2_specular.png");
_object3d[4].load2(Size.X, Size.Y, "stripped_warped_stem.png", "container2_specular.png");
_object3d[5].load2(Size.X, Size.Y, "flowering_azalea_leaves.png", "container2_specular.png");
_camera = new Camera(new Vector3(Cx,Cy,Cz), Size.X / Size.Y);
```

Inisialisasi light object menggunakan code dibawah ini

```
// inisialisai light object
_vertexBufferLightObject = GL.GenBuffer();
GL.BindBuffer(BufferTarget.ArrayBuffer, _vertexBufferLightObject);
GL.BufferData(BufferTarget.ArrayBuffer, _vertices.Length * sizeof(float), _vertices, BufferUsageHint.StaticDraw);
_lampShader = new Shader(Source + "/Shaders/shader.vert", Source + "/Shaders/shader.frag");
_vaoLamp = GL.GenVertexArray();
```

Di fungsi onRenderFrame, kami memanggil semua cube dengan looping seperti dibawah ini, memanggil dengan masing masing bagian.

```
for (int i = 0; i < _cubePositions.Length; i++)
{
    _object3d[0].render(4, _camera.GetViewMatrix(), _camera.GetProjectionMatrix(),
    _lightPos, _camera.Position, _camera.Front, _pointLightPositions, _cubePositions[i], i);
}

for (int i = 0; i < _cubePos2.Length; i++)
{
    _object3d[1].render(4, _camera.GetViewMatrix(), _camera.GetProjectionMatrix(),
    _lightPos, _camera.Position, _camera.Front, _pointLightPositions, _cubePos2[i], i);
}

for (int i = 0; i < _cuberumput.Length; i++)
{
    _object3d[2].render(4, _camera.GetViewMatrix(), _camera.GetProjectionMatrix(),
    _lightPos, _camera.Position, _camera.Front, _pointLightPositions, _cuberumput[i], i);
}
for (int i = 0; i < _rumputgelap.Length; i++)
{
    _object3d[3].render(4, _camera.GetViewMatrix(), _camera.GetProjectionMatrix(),
    _lightPos, _camera.Position, _camera.Front, _pointLightPositions, _rumputgelap[i], i);
}

for (int i = 0; i < danau.Length; i++)
{
    _object3d[4].render(4, _camera.GetViewMatrix(), _camera.GetProjectionMatrix(),
    _lightPos, _camera.Position, _camera.Front, _pointLightPositions, danau[i], i);
}

for (int i = 0; i < _perhutani.Length; i++)
{
    _object3d[5].render(4, _camera.GetViewMatrix(), _camera.GetProjectionMatrix(),
    _lightPos, _camera.Position, _camera.Front, _pointLightPositions, _perhutani[i], i);
}
```

Kami mengelompokkan menjadi object, kemudian dipanggil di Window.cs seperti berikut

```
1 reference
public Window(GameWindowSettings gameWindowSettings, NativeWindowSettings nativeWindowSettings) : base(gameWindowSettings, nativeWindowSettings)
{
    _object3d[0] = new Asset3d();
    _object3d[1] = new Asset3d();
    _object3d[2] = new Asset3d();
    _object3d[3] = new Asset3d();
    _object3d[4] = new Asset3d(); //danau
    _object3d[5] = new Asset3d(); //hutan
}
```

Kemudian mengaktifkan lighting di fungsi onrenderframe

```
//lighting
_lampShader.Use();

_lampShader.SetMatrix4("view", _camera.GetViewMatrix());
_lampShader.SetMatrix4("projection", _camera.GetProjectionMatrix());
```

Untuk mengatur rgb otomatis, kami menggunakan variabel RGB dan kami ganti value saat tiap frame nya. Variabelnya seperti gambar berikut.

```
//KEYBOARD FUNCTION

float R_ = 0.5f;
float G_ = 0.5f;
float B_ = 1.0f;
```

Untuk mengatur pergerakan kamera, kami menggunakan seperti contoh openTK di gambar berikut

```
if (input.IsKeyDown(Keys.Up))
{
    _camera.Position += _camera.Front * cameraspeed * 2 * (float)args.Time;
}
if (input.IsKeyDown(Keys.Down))
{
    _camera.Position -= _camera.Front * cameraspeed * 2 * (float)args.Time;
}
if (input.IsKeyDown(Keys.Left))
{
    _camera.Position -= _camera.Right * cameraspeed * 2 * (float)args.Time;
}
if (input.IsKeyDown(Keys.Right))
{
    _camera.Position += _camera.Right * cameraspeed * 2 * (float)args.Time;
}
if (input.IsKeyDown(Keys.Space))
{
    _camera.Position += _camera.Up * (cameraspeed*5.0f) * (float)args.Time;
}
if (input.IsKeyDown(Keys.LeftShift))
{
    _camera.Position -= _camera.Up * cameraspeed * (float)args.Time;
}
```

Problem yang kami hadapi dan tidak bisa kami atasi adalah mengatur frame per second yang berhubungan dengan codingan counter kami. Setiap CPU anggota kami berbeda spesifikasi, sehingga saat memproses animasi akan menghasilkan hasil yang sangat berbeda di setiap laptop. Kami sangat kesulitan maka salah satu cara yang dapat kita lakukan adalah menggunakan 1 laptop saja sebagai pusat demo animasi.

Program untuk animasi akan saya jelaskan dan akan

```
if (counter < 250)
{
    _camera.Position += _camera.Front * cameraspeed * 5 * (float)args.Time;
    _camera.Position += _camera.Up * cameraspeed * 5 * (float)args.Time;

    R = R - 0.0005f;
    G = G - 0.0005f;
    B = B - 0.00085f;

    GL.ClearColor(R, G, B, 1);
    Console.WriteLine("naik");
}
else if (counter < 300)
{
    _camera.Position += _camera.Front * cameraspeed * 5 * (float)args.Time;

    R = R - 0.0005f;
    G = G - 0.0005f;
    B = B - 0.00085f;
    GL.ClearColor(R, G, B, 1);
    Console.WriteLine("jalan");
}
```

Program yang pertama digunakan untuk menggerakkan kamera ke atas, dan program yang kedua digunakan untuk menggerakkan kamera maju ke depan arah dari setiap pergerakan ditentukan dari arah kamera jika ingin maju maka camera position ditambah camera front, jika ke atas maka kamera up jika kamera kebawah atau ke belakang maka menggunakan program yang sama tapi dikurangi, pada kedua program tersebut terdapat variabel R , G, B berguna untuk mengubah background world.

```

else if (counter < 1201)
{
    Console.WriteLine(counter);
    //var deltaX = -1f - _lastPos.X;
    _camera.Yaw += 90f;
    Console.WriteLine("muter");
    R = R - 0.0005f;
    G = G - 0.0005f;
    B = B - 0.00085f;
    GL.ClearColor(R, G, B, 1);
}

```

```

else if (counter < 2852)
{
    Console.WriteLine(counter);
    //var deltaX = -1f - _lastPos.X;
    _camera.Pitch -= 15f;
    Console.WriteLine("lihat atas");
    R = R + 0.00117f;
    G = G + 0.00076f;
    B = B + 0.000352f;
    GL.ClearColor(R, G, B, 1);
}

```

Kedua program tersebut sebenarnya sama hanya saja perbedaannya adalah kamera yaw berguna untuk menghadap kanan atau kiri sedangkan pitch berguna untuk menghadap atas atau bawah, jika plus maka ke kanan atau ke atas jika minus sebaliknya

III. Penutupan

Ide dibuat oleh kami bertiga. Jeffry mengerjakan Skin dan Object, Kevin membuat animasi, dan Terrius mengatur warna. Demikian laporan yang kami buat, saran dan kritik sangat kami harapkan demi perkembangan diri dalam proyek-proyek berikutnya. Kami memohon maaf apabila ada kesalahan kata dan ejaan. Atas waktunya kami ucapkan terima kasih sebesar-besarnya.