

## MODUL 15 – Posisi, Gerakan, dan Navigasi Untuk GameObject Character

---

### A. TUJUAN

- Mahasiswa dapat mengontrol GameObject 2D (dan membatasi gerakan dalam rectangle)
- Mahasiswa dapat mengontrol GameObject 3D (dan membatasi gerakan dalam rectangle)

### B. PETUNJUK

1. Awali setiap kegiatan praktikum dengan berdoa
2. Baca dan pahami tujuan, dasar teori, dan latihan-latihan praktikum dengan baik
3. Kerjakan tugas-tugas praktikum dengan baik, sabar dan jujur
4. Tanyakan kepada dosen apabila ada hal-hal yang kurang jelas

### C. ALOKASI WAKTU: 6 jam pelajaran

### D. DASAR TEORI

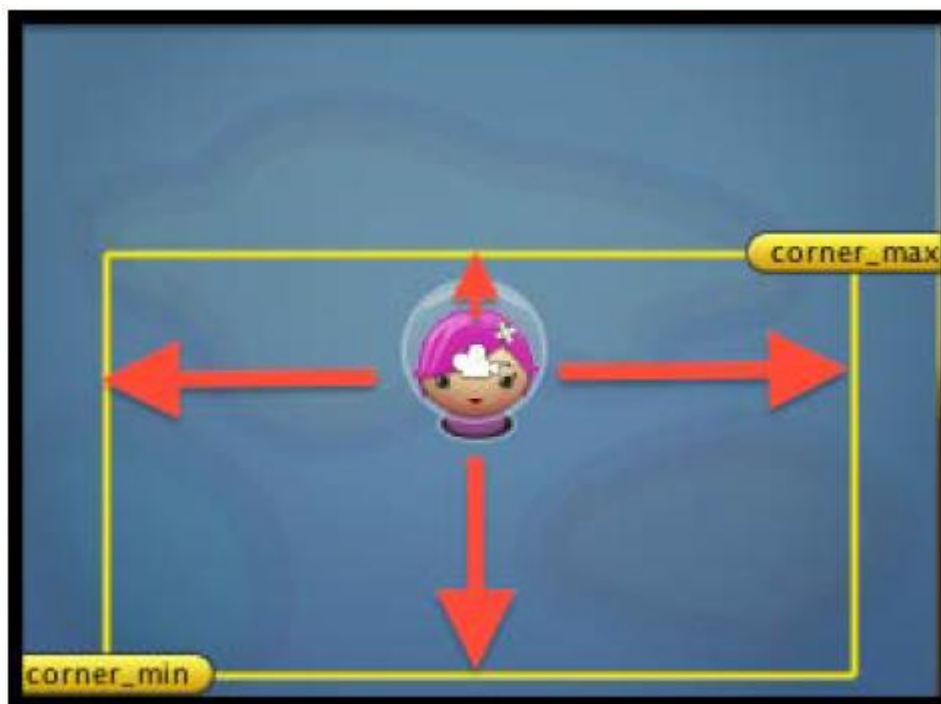
- Banyak GameObject dalam Movement games yang bias dikendalikan oleh pemain, oleh simulasi fisik lingkungan, atau oleh logic NPC (Non-Player Character). Sebagai contoh, objek yang mengikuti jalur tertentu, atau bergerak maju, atau terbang menjauh dari posisi tertentu dari karakter Unity menyediakan beberapa controller, untuk karakter first dan third-person, juga untuk kendaraan seperti mobil dan pesawat. Movement dari GameObject dapat dikendalikan juga oleh state machines dari Unity Mecanim animation system. Unity menyediakan banyak Class dan Component termasuk Class Vector3 dan rigid physic body untuk memodelkan pergerakan realistic, gaya, dan tumbukan pada games. Modul ini akan menggunakan fitur game engine tersebut untuk mengimplementasikan NPC dan pergerakan karakter musuh.
- Untuk game 3D (pada beberapa bagian dari game 2D), class dasar dari object adalah class Vector3 – object yang menyimpan dan memanipulasi nilai (x, y, z) yang merepresentasikan posisi pada ruang 3D. Jika digambarkan garis imajiner dari titik awal (0, 0, 0) ke titik tertentu pada ruang, maka arah dan panjang dari panah ini (vector) dapat merepresentasikan kecepatan atau gaya (atau disebut dengan nilai magnitude pada arah tertentu).
- Jika diabaikan semua komponen controller karakter, collider, dan system physics dalam Unity, kita dapat menulis kode yang melakukan teleport object secara langsung pada lokasi tertentu (x, y, z) pada scene. Dan kadangkala hal ini juga dilakukan, contoh jika ingin memunculkan object secara tiba-tiba di titik tertentu. Akan tetapi pada banyak kasus, object dipindahkan secara realistic, maka harus diterapkan gaya (force) pada object, atau mengubah kecepatan pergerakannya. Atau jika misalkan object memiliki komponen Controller Character, maka dapat menambahkan pesan Move( ). Dengan adanya Unity NavMeshAgents (dan class lain

dari Navigation Meshes), kita sekarang dapat men-set tujuan dari object dengan NavMeshAgent, kemudian logic pathfinding yang tersedia akan melakukan pergerakan object NPC menuju lokasi tujuan (x, y, z) tertentu.

- Selain menentukan teknik apa yang digunakan untuk menggerakkan object, game yang dibuat juga harus menentukan bagaimana menentukan lokasi tujuan, atau arah dan magnitude dari perubahan pergerakan.
- Konsep inti lain pada pergerakan dan pembuatan NPC antara lain:
  - a. Titik Spawn : Lokasi tertentu dalam scene dimana object dibuat atau dipindahkan.
  - b. Waypoints : lokasi sekuens untuk mendefinisikan path untuk NPC atau character Player untuk diikuti.
  - c. Checkpoints : lokasi yang sekali dilalui, akan mengubah atau mengaktifkan suatu kejadian dalam game (contoh, menambah extra time, atau jika pemain mati, akan hidup kembali di titik cekpoint terakhir)

## E. LATIHAN PRAKTIKUM

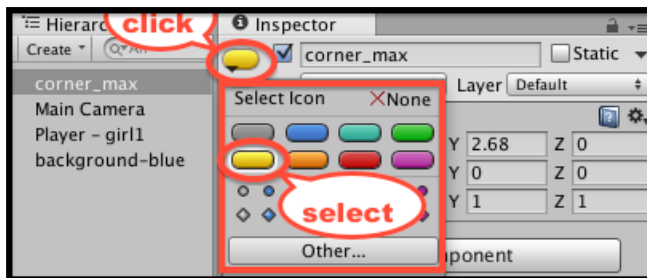
1. Control Player dari GameObject 2D (dan membatasi gerakan dalam rectangle)  
Praktikum ini akan melanjutkan Inventory GUI dari Modul 3, dengan membuat batasan pergerakan player pada rectangle.



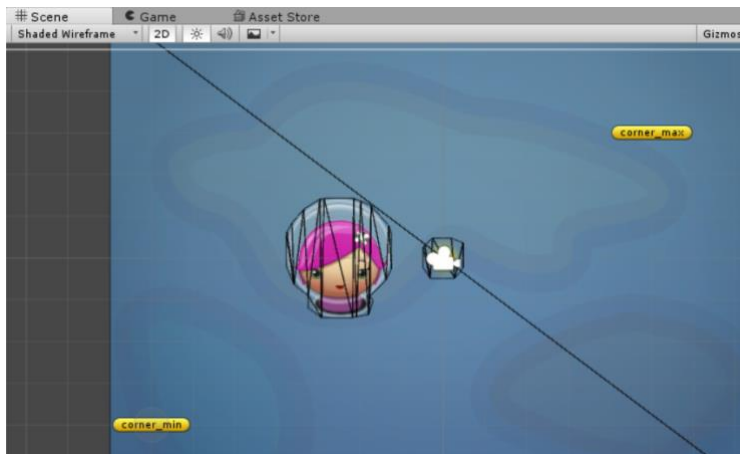
Praktikum ini akan menggunakan minigame Space Girl dari Modul 3, Inventory GUI. Mulai dengan membuka package game ini.

Untuk membuat sprite 2D dikendalikan oleh user dengan pergerakan yang dibatasi didalam rectangle, ikuti langkah-lagkah berikut:

1. Buat GameObject baru yang kosong dan beri nama corner\_max, tempatkan diatas dan kanan dari GameObject Player\_girl1. Pada jendela Hierarchy pilih GameObject ini, pilih icon warna kuning yang terdapat pada panel Inspector.



2. Duplikasi GameObject `corner_max` dengan memberi nama clone sebagai `corner_min`, posisikan dibawah dan kiri dari GameObject `player-spaceGirl1`.



3. Modifikasi script C# `PlayerMove` untuk ditambahkan beberapa variable baru pada bagian awal class:

```
public Transform corner_max;
public Transform corner_min;
private float x_min;
private float y_min;
private float x_max;
private float y_max;
```

4. Modifikasi method `Awake()` dari `PlayerMove` sehingga memiliki reference ke `SpriteRenderer`, dan menggunakan object ini untuk men-set batas maksimum pergerakan di X dan Y:

```
void Awake(){
    rigidBody2D = GetComponent<Rigidbody2D>();
    x_max = corner_max.position.x;
    x_min = corner_min.position.x;
    y_max = corner_max.position.y;
    y_min = corner_min.position.y;
}
```

5. Tambahkan method baru `KeepWithinMinMaxRectangle()`:

```
private void KeepWithinMinMaxRectangle()
{
    float x = transform.position.x;
    float y = transform.position.y;
    float z = transform.position.z;
    float clampedX = Mathf.Clamp(x, x_min, x_max);
    float clampedY = Mathf.Clamp(y, y_min, y_max);
    transform.position = new Vector3(clampedX, clampedY, z);
}
```

6. Modifikasi Method FixedUpdate() dengan menambahkan KeepWithinMaxRectangle() agar dipanggil untuk membatasi pergerakan:

```
void FixedUpdate()
{
    float xMove = Input.GetAxis("Horizontal");
    float yMove = Input.GetAxis("Vertical");
    float xSpeed = xMove * speed;
    float ySpeed = yMove * speed;
    Vector2 newVelocity = new Vector2(xSpeed, ySpeed);
    rigidBody2D.velocity = newVelocity;
    // restrict player movement
    KeepWithinMinMaxRectangle();
}
```

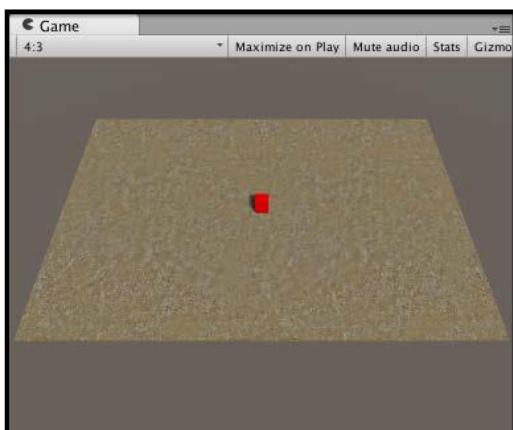
7. Pilih GameObject player-SpaceGirl1 pada view Hierarchy, drag GameObject corner\_max dan corner\_min ke variable public corner\_max dan corner\_min didalam inspector.
8. Sebelum menjalankan scene ini, coba untuk menset posisi corner\_max, corner\_min, dan player-SpaceGirl1 sedemikian rupa sehingga player berada ditengah antara corner\_max (kanan atas dari player) dan corner\_min (kiri bawah dari player).
9. Saat sudah bekerja dengan baik, kita dapat menampilkan batas rectangle dengan menggambarannya pada scene. Tambahkan method berikut pada script PlayerMove:

```
void OnDrawGizmos()
{
    Vector3 top_right = Vector3.zero;
    Vector3 bottom_right = Vector3.zero;
    Vector3 bottom_left = Vector3.zero;
    Vector3 top_left = Vector3.zero;
    if (corner_max && corner_min)
    {
        top_right = corner_max.position;
        bottom_left = corner_min.position;
        bottom_right = top_right;
        bottom_right.y = bottom_left.y;
        top_left = top_right;
        top_left.x = bottom_left.x;
    }
    //Set the following gizmo colors to YELLOW
    Gizmos.color = Color.yellow;
    //Draw 4 lines making a rectangle
    Gizmos.DrawLine(top_right, bottom_right);
    Gizmos.DrawLine(bottom_right, bottom_left);
    Gizmos.DrawLine(bottom_left, top_left);
    Gizmos.DrawLine(top_left, top_right);
}
```



## 2. Control Player dari GameObject 3D ( dan membatasi gerakan didalam rectangle)

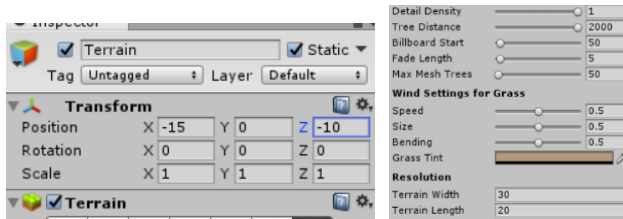
Project ini akan menggunakan scene 3D sederhana yang terdiri dari terrain bertekstur, Main Camera, dan kubus kecil berwarna merah yang dapat dipindahkan user dengan menggunakan 4 tombol panah. Batas pergerakan dari kubus akan ditentukan menggunakan teknik yang sama yang digunakan pada praktikum sebelumnya.



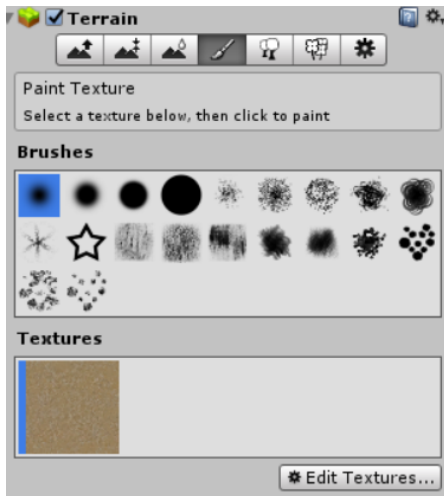
1. Buat project 3D baru yang kosong.
2. Setelah project kosong siap, import satu tekstur terrain yang dinamakan SandAlbedo. Caranya dengan memilih menu Assets | Import Package | Environments, deselect

semua Package dengan mengklik tombol None, kemudian pilih asset SandAlbedo saja  
Assets/Environment/TerrainAssets/SurfaceTextures/SandAlbedo.psd  
(pilih asset yang dipakai saja untuk membuat ukuran folder project Asset kecil)

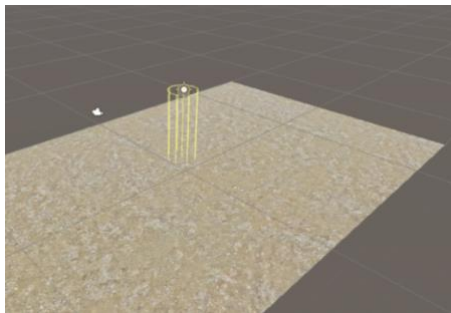
3. Buat terrain dengan posisi (-15,0,-10) dan berukuran 30 x 20.



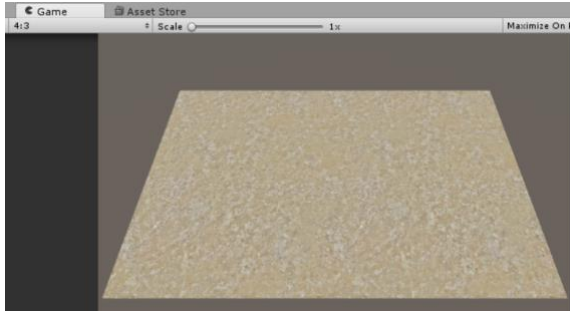
4. Berikan tekstur paint dari terrain ini dengan tekstur SandAlbedo.



5. Set Directional Light agar menghadap tegak lurus terhadap terrain (rotasikan agar posisi dapat tegak lurus).



6. Transformasi Main Camera:
  - Posisi = (0, 20, -15)
  - Rotasi = (60, 0, 0)
7. Ubah Aspect Ration dari Game Panel dari Free Aspect menjadi 4:3. Seluruh Terrain seharusnya dapat dilihat pada Game Panel.



8. Buat GameObject kosong baru dengan nama `corner_max`. Tempatkan pada (14,0,9). Pilih pada Hierarchy, kemudian pilih icon warna kuning pada panel Inspector.
9. Duplikat `corner_max` dan ubah namanya menjadi `corner_min`, posisikan pada (-14,0,-9). Koordinat dari dua GameObject baru ini akan menjadi batas pergerakan dari karakter player.
10. Buat GameObject baru Cube dan beri nama `Cube-player` pada posisi (0,0.5,0), dan ukurannya (1,1,1).
11. Tambahkan RigidBody pada `Cube-player` (Physics | RigidBody), pada panel RigidBody hilangkan centang pada Use Gravity.
12. Buat material merah dengan nama `m_red`. Gunakan material ini pada `Cube-player`.
13. Buat Script baru dan beri nama `PlayerControl` yang akan dipakai untuk mengontrol pergerakan `Cube-player`.

```

using UnityEngine;
using System.Collections;
public class PlayerControl : MonoBehaviour
{
    public Transform corner_max;
    public Transform corner_min;
    public float speed = 40;
    private Rigidbody rigidBody;
    private float x_min;
    private float x_max;
    private float z_min;
    private float z_max;
    void Awake()
    {
        rigidBody = GetComponent<Rigidbody>();
        x_max = corner_max.position.x;
        x_min = corner_min.position.x;
        z_max = corner_max.position.z;
        z_min = corner_min.position.z;
    }
    void FixedUpdate()
    {
        KeyboardMovement();
        KeepWithinMinMaxRectangle();
    }
    private void KeyboardMovement() {
        float xMove = Input.GetAxis("Horizontal") * speed * Time.deltaTime;
        float zMove = Input.GetAxis("Vertical") * speed * Time.deltaTime;
        float xSpeed = xMove * speed;
        float zSpeed = zMove * speed;
        Vector3 newVelocity = new Vector3(xSpeed, 0, zSpeed);
        rigidBody.velocity = newVelocity;
        // restrict player movement
        KeepWithinMinMaxRectangle();
    }
    private void KeepWithinMinMaxRectangle()
    {
        float x = transform.position.x;
        float y = transform.position.y;
        float z = transform.position.z;
        float clampedX = Mathf.Clamp(x, x_min, x_max);
        float clampedZ = Mathf.Clamp(z, z_min, z_max);
        transform.position = new Vector3(clampedX, y, clampedZ);
    }
    void OnDrawGizmos()
    {
        Vector3 top_right = Vector3.zero;
        Vector3 bottom_right = Vector3.zero;
        Vector3 bottom_left = Vector3.zero;
        Vector3 top_left = Vector3.zero;
        if (corner_max && corner_min)
        {
            top_right = corner_max.position;
            bottom_left = corner_min.position;
            bottom_right = top_right;
            bottom_right.z = bottom_left.z;
            top_left = bottom_left;
            top_left.z = top_right.z;
        }
        //Set the following gizmo colors to YELLOW
        Gizmos.color = Color.yellow;
        //Draw 4 lines making a rectangle
        Gizmos.DrawLine(top_right, bottom_right);
        Gizmos.DrawLine(bottom_right, bottom_left);
        Gizmos.DrawLine(bottom_left, top_left);
        Gizmos.DrawLine(top_left, top_right);
    }
}

```

14. Pilih Cube-player pada panel Hierarchy, drag corner\_max dan corner\_min kedalam variable public yang ada didalam panel inspector.





15. Ketika dijalankan, corner\_max dan corner\_min akan membatasi gerakan dari character player dalam hal ini Cube-player.

**A. TUGAS PRAKTIKUM**

1. Tambahkan dua checkpoint lagi dari praktikum 2 yang telah dikerjakan!

--- SELAMAT BELAJAR ---