

# PRAKTIKUM ANIMASI DAN GAME PERTEMUAN :8

## Camera & Character Movement

NIM	:	1918039
NAMA	:	M.Sofian Attasauri
MATERI	:	Camera & Character Movement
KELAS	:	D
TUGAS	:	Membuat pergerakan karakter player

### 8.1 Tujuan

1. Praktikan dapat mengetahui *Game* 2D dan *Game* 3D.
2. Praktikan dapat mengetahui jenis *Game*.
3. Praktikan dapat menerapkan *Game* pada Unity.

### 8.2 Alat dan Bahan

1. Laptop/pc.
2. Modul Praktikum Animasi dan *Game* 2024.
3. Unity 2017.

### 8.3 Langkah-Langkah Membuat Tugas

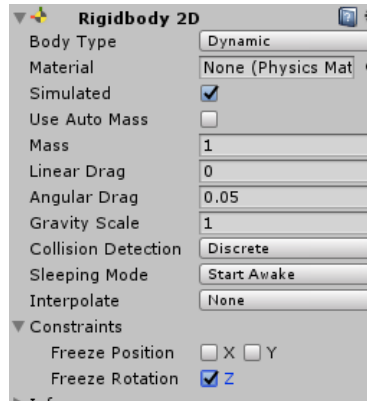
#### A. Membuat Pergerakan Player

1. Buat file *projek Unity* Tugas 7



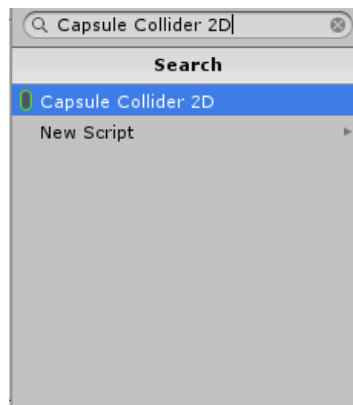
Gambar 8.1 Buka projek tugas 7

2. Pada properties *Rigidbody 2D*, klik *Constraint*, dan centang pada bagian Z, *Freeze Rotation*, kotak tersebut dicentang agar saat karakter menginjak tanah tidak terguling



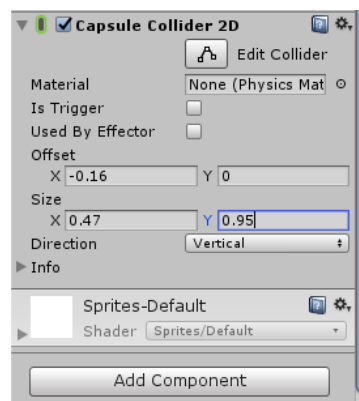
Gambar 8.2 Tampilan *Freeze Rotation*

3. Tambahkan lagi satu komponen bernama *Capsule Collider 2D*



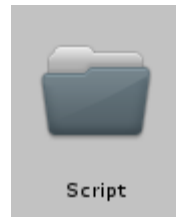
Gambar 8.3 *Capsule Collider 2D*

4. Pada properties *Capsule Collider 2D*, isikan nilai *Offset* dan *Size*



Gambar 8.4 Mengatur nilai *Offset* dan *Size*

5. Klik kanan pada asset lalu buat folder baru dan beri nama *Script*



Gambar 8.5 Membuar folder *Scirpt*

6. Pada folder *script* klik kanan pilih *C# script*



Gambar 8.6 Membuat script untuk warrior

7. Lalu pada script karakter masukkan Source Berikut

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class karakter : MonoBehaviour
{
    Rigidbody2D rb;
    [SerializeField]
    Transform groundcheckCollider;
    [SerializeField]
    LayerMask groundLayer;
    const float groundCheckRadius = 0.2f;
    [SerializeField]
    bool isGrounded;
    [SerializeField]
    float speed = 1;
    [SerializeField]
    float jumpPower = 100;
    float horizontalValue;
    bool facingRight;
    bool jump;
    private void Awake()
    {
        rb = GetComponent<Rigidbody2D>();
    }
    // Update is called once per frame
    void Update()
    {
        horizontalValue
        Input.GetAxisRaw("Horizontal");
        if (Input.GetButtonDown("Jump"))
            jump = true;
```

```

        else if (Input.GetButtonUp("Jump"))
            jump = false;
    }
    void FixedUpdate()
    {
        GroundCheck();
        Move(horizontalValue, jump);
    }

    void GroundCheck()
    {
        isGrounded = false;
        Collider2D[] colliders =
Physics2D.OverlapCircleAll(groundcheckCollider.position, groundCheckRadius,
        groundLayer);
        if (colliders.Length > 0)
            isGrounded = true;
    }

    void Move(float dir, bool jumpflag)
    {
        if (isGrounded && jumpflag)
        {
            isGrounded = false;
            jumpflag = false;
            rb.AddForce(new Vector2(0f, jumpPower));
        }

        #region gerak kanan kiri
        float xVal = dir * speed * 100 *
Time.fixedDeltaTime;
        Vector2 targetVelocity = new Vector2(xVal,
rb.velocity.y);
        rb.velocity = targetVelocity;
        if (facingRight && dir < 0)
        {
            transform.localScale = new Vector3(-1, 1,
1);
            facingRight = false;
        }
        else if (!facingRight && dir > 0)
        {
            transform.localScale = new Vector3(1, 1, 1);
            facingRight = true;
        }
        #endregion
    }
}

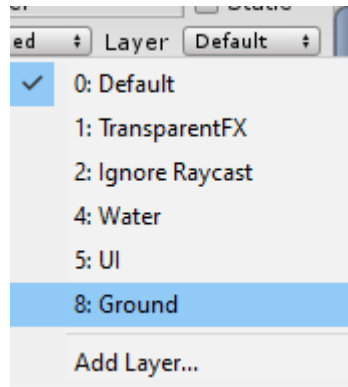
```

8. Isikan *User Layer 8* dengan nama *Ground*



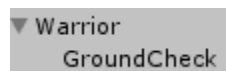
Gambar 8.7 Mengisi *layer 8* menjadi *Ground*

9. Ganti *Layer Default* menjadi *Ground*



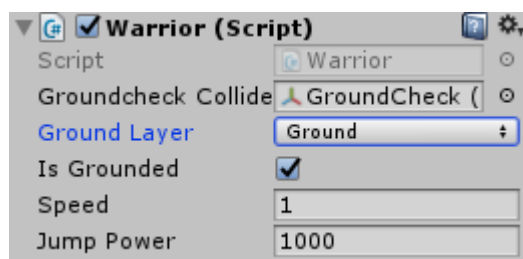
Gambar 8.8 Mengganti layer menjadi *Ground*

10. Klik kanan karakter kemudian pilih *Create Empty* dan ubah namanya menjadi *GroundCheck*



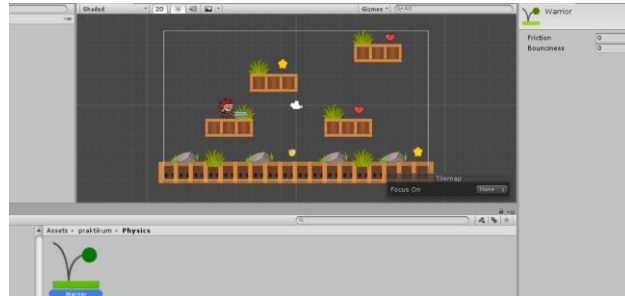
Gambar 8.9 Membuat *GroundCheck*

11. drag *GroundCheck* arahkan kedalam *Groundcheck Collider* yang ada pada *Inspector*



Gambar 8.10 Penyesuaian *GroundCheck*

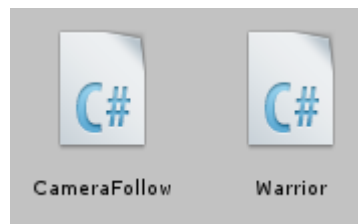
12. Klik kanan folder “*Physics*” pilih *Create>Physics Material 2D* dan beri nama “*karakter*” lalu ubah nilai *Friction* dan *Bounciness* menjadi 0



Gambar 8.11 Tampilan *Physics* karakter

## B. Camera Movement

1. Tambahkan script dalam folder *Script* dan beri nama “*CameraFollow.cs*”



Gambar 8.12 Membuat dan Menambahkan *Source Code*

2. Tambahkan *Source Code* berikut pada Script *CameraFollow*

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class CameraFollow : MonoBehaviour
{
    public float xMargin = 0.5f;
    public float yMargin = 0.5f;
    public float xSmooth = 4f;
    public float ySmooth = 4f;
    public Vector2 maxXAndY;
    public Vector2 minXAndY;
    private Transform player;
    void Awake()
    {
        player =
        GameObject.FindGameObjectWithTag("Player").transform;
    }
    bool CheckXMargin()
    {
        return Mathf.Abs(transform.position.x -
        player.position.x) > xMargin;
    }
    bool CheckYMargin()
    {

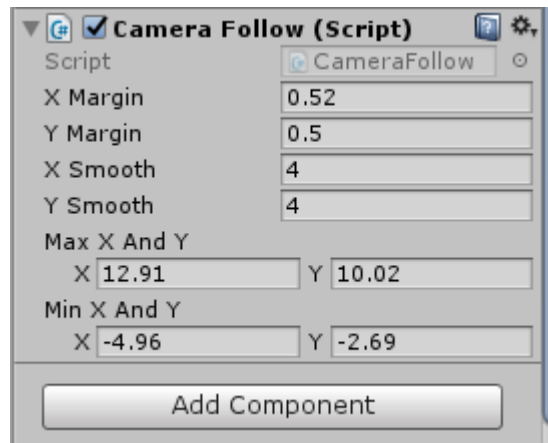
```

```

        return Mathf.Abs(transform.position.y -
player.position.y) > yMargin;
    }
    void FixedUpdate()
    {
        TrackPlayer();
    }
    void TrackPlayer()
    {
        float targetX = transform.position.x;
        float targetY = transform.position.y;
        if (CheckXMargin())
            targetX = Mathf.Lerp(transform.position.x,
player.position.x,
            xSmooth * Time.deltaTime);
        if (CheckYMargin())
            targetY = Mathf.Lerp(transform.position.y,
player.position.y,
            ySmooth * Time.deltaTime);
        targetX = Mathf.Clamp(targetX, minXAndY.x,
maxXAndY.x); targetY =
        Mathf.Clamp(targetY, minXAndY.y, maxXAndY.y);
        transform.position = new
            Vector3(targetX, targetY,
transform.position.z);
    }
}

```

3. Pergi ke *Inspector Main Camera* kemudian lakukan setting *Camera Follow* seperti gambar berikut



Gambar 8.13 Mensetting *Camera Follow*

4. Jika di *Play*, maka camera akan mengikuti pergerakan setiap karakter.



Gambar 8.14 Tampilan game ketika di Run

#### 8.4 Kesimpulan

1. Bahasa C# adalah sebuah Bahasa pemrograman modern yang bersifat generalpurpose, berorientasi objek, yang dapat digunakan untuk membuat program diatas arsitektur Microsoft .Net. Framework. Dalam Unity Bahasa C# adalah Bahasa pemrograman default untuk penegmabangan game menggunakan Unity
2. Komponen Rigidbody2D menempatkan objek dibawah kendali sifat fisika. Objek hanya dapat bergerak di bidang XY dan hanya dapat memutar pada sumbu yang tegak lurus terhadap bidang itu
3. Camera adalah perangkat yang menangkap dan menampilkan dunia kepada pemain. Dengan menyesuaikan dan memanipulasi kamera. Setiap game dapat memiliki jumlah kamera yang tida terbatas dalam sebuah scene. Meraka dapat diatur untuk menyajikan dalam urutan apapun

Nilai	Asisten Laboratorium
	(M.Rafi Faddilani) 2118144