



TUGAS PERTEMUAN: 8

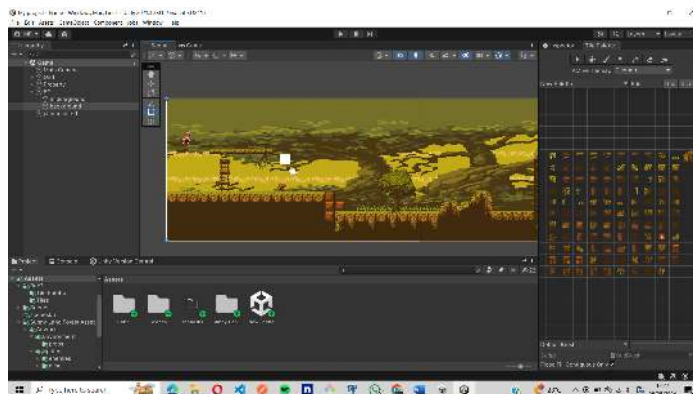
CHARACTER MOVEMENT

NIM	:	2118100
Nama	:	Moch Arif Rochmanullah
Kelas	:	C
Asisten Lab	:	RIFAL RIFQI RHOMADON (2218106)
Baju Adat	:	
Referensi	:	

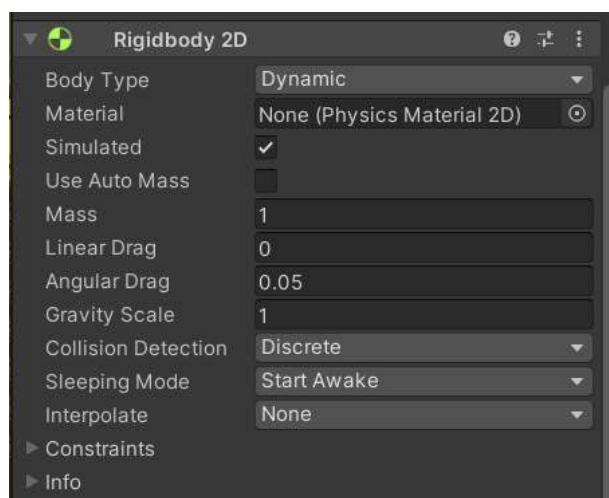
8.1 Tugas 8 : Membuat pergerakan layer sesuai asset sebelumnya

A. Membuat Pergerakan Player

1. Buka project bab 7

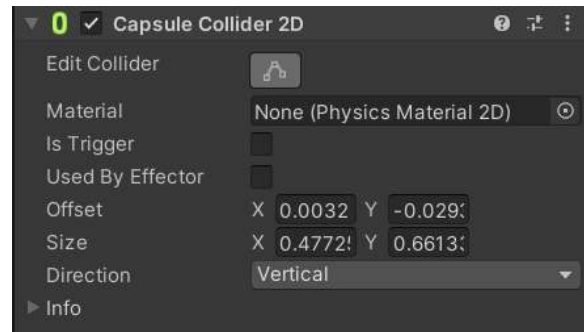


2. Klik player-idle-1 tambahkan Component Rigidbody 2D, sesuaikan settingannya seperti gambar berikut, Centang pada Freeze Rotation Z

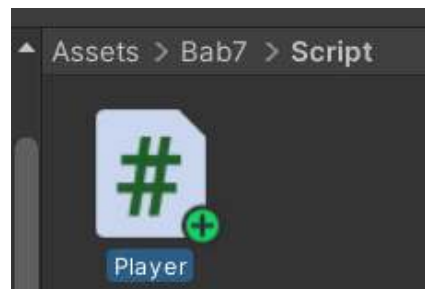




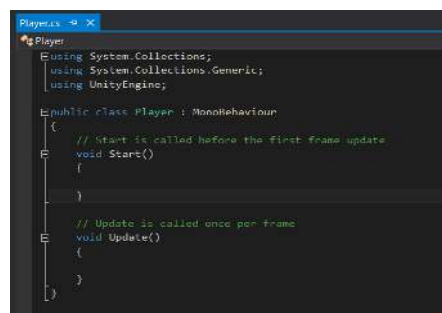
3. Lalu tambahkan komponen Capsule Colider di player-idle-1, lalu klik icon sebelah kanan edit collider. Lalu cockan garis oval degan karakternya atau bisa di inputkan Offset X, Y dan juga Size X, Y nya



4. Buka Folder praktikum, lalu bikin folder baru bernama Script. Masuk kedalam folder Script, lalu buat C# Script, beri nama Player



5. Drag & drop script player kedalam Hirarki player-idle-1, lalu klik 2x pada script player maka akan masuk kedalam text editor seperti ini



6. Masukkan source code dibawah ini, pastikan nama public class harus sama dengan nama file yang dibuat.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Player : MonoBehaviour
{
    Rigidbody2D rb;

    [SerializeField] float speed = 1;
    float horizontalValue;
```



```
bool facingRight;

private void Awake()
{
    rb = GetComponent<Rigidbody2D>();
}

void Update ()
{
    horizontalValue = Input.GetAxisRaw("Horizontal");
}

void FixedUpdate()
{
    Move(horizontalValue);
}

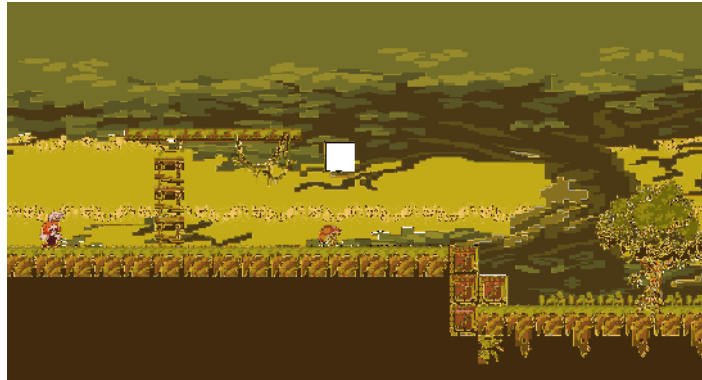
void Move(float dir)
{
    #region gerak kanan kiri
    float xVal = dir * speed * 100 * Time.fixedDeltaTime;
    Vector2 targetVelocity = new Vector2(xVal, rb.velocity.y);
    rb.velocity = targetVelocity;

    if (facingRight && dir < 0)
    {
        // ukuran player
        transform.localScale = new Vector3(-1, 1, 1);
        facingRight = false;
    }

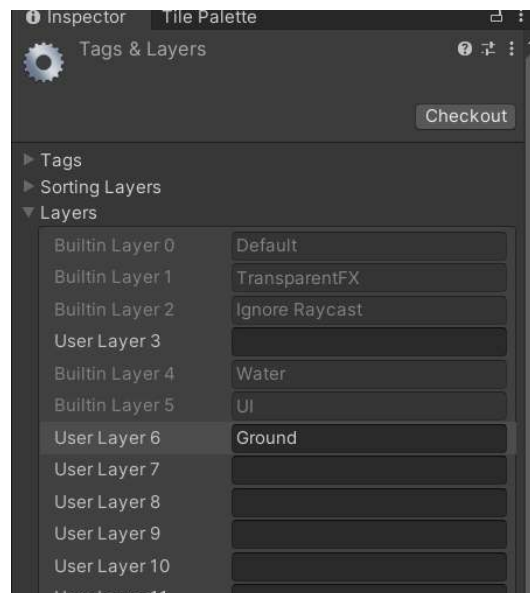
    else if (!facingRight && dir > 0)
    {
        // ukuran player
        transform.localScale = new Vector3(1, 1, 1);
        facingRight = true;
    }

    #endregion
}
}
```

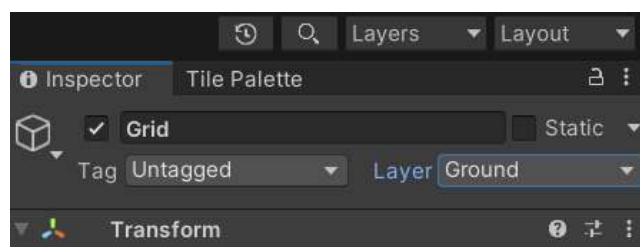
7. Untuk mencoba Source code diatas berhasil, Tekan dikeyboard “a” atau “left arrow” untuk ke arah kiri, tekan “d” atau “right arrow” untuk ke arah kanan



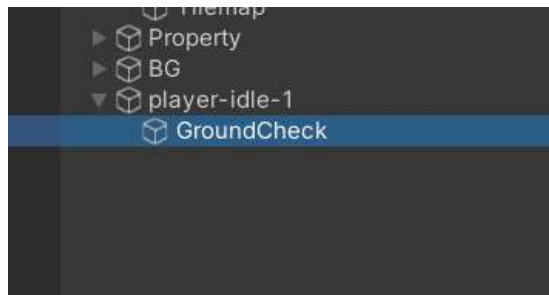
8. Untuk membuat player loncat menggunakan spasi, kita perlu membuat GroundCheck dengan cara, klik Grid pada Hierarchy, pergi ke inspector, pilih Layer, Klik Add Layer. lalu isi “Ground” pada User Layer 6



9. Ubah layer menjadi Ground, jika muncul pop up change layer, klik yes



10. klik kanan pada player-idle-1, lalu Create empty, beri nama GroundCheck



11. Klik pada Hirarki GorundCheck, lalu gunakan “Move Tools” untuk memindahkan ke bagian bawah Player seperti gambar berikut



12. Kembali ke script Player tambahkan source code seperti ini

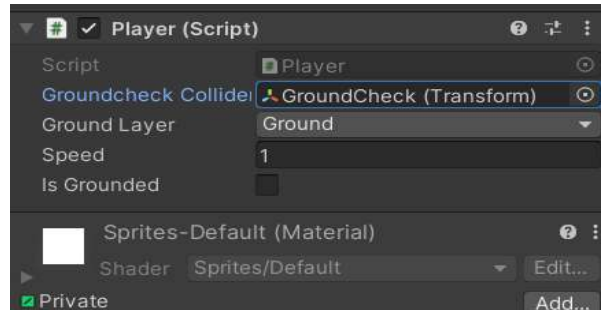
```
[SerializeField] Transform groundcheckCollider;  
[SerializeField] LayerMask groundLayer;  
  
const float groundCheckRadius = 0.2f; // +  
[SerializeField] float speed = 1;  
float horizontalValue;
```

13. Buat void ground check dibawah void fixedUpdate & tambahkan GorunCheck(); pada void fixedUpdate

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



14. Klik player-idle-1, lalu ke inspector ke effect Player script di bagian “Goruncheck collider” tekan icon lalu pilih yang GorundCheck Transform, dan pada Ground Layer pilih Ground



15. Lalu untuk membuat player melompat tambahkan script berikut

```
[SerializeField] float jumpPower = 100;

bool jump;
```

16. Tambahkan juga script berikut di bagian void update

```
if (Input.GetButtonDown("Jump"))
    jump = true;
else if (Input.GetButtonUp("Jump"))
    jump = false;
```

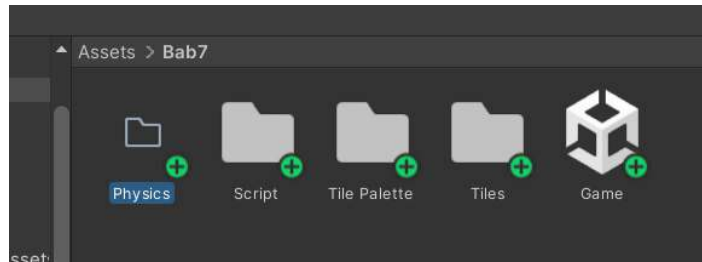
17. Tambahkan juga jump pada parameter Move

```
void FixedUpdate()
{
    GroundCheck();
    Move(horizontalValue, jump);
}
```

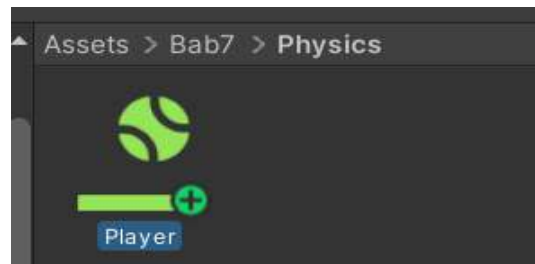
18. Tambahkan script berikut pada void Move

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

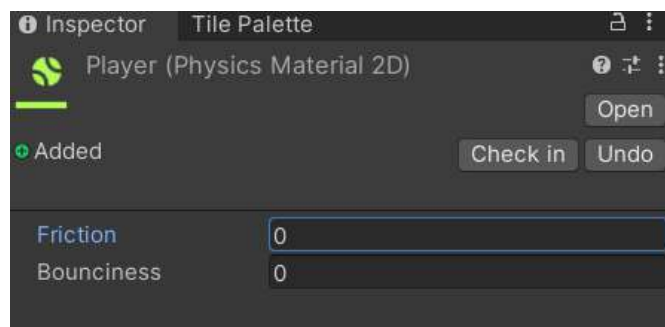
19. Buat folder baru di praktikum bernama “Physics”



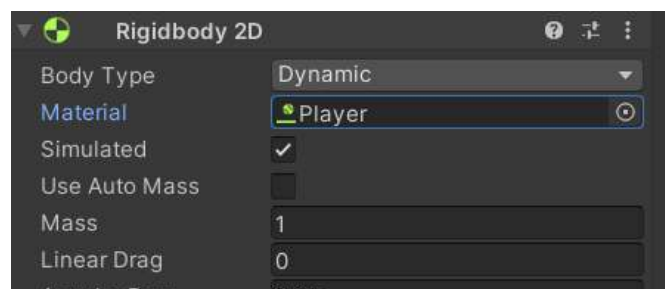
20. Didalam folder Pyshics create > 2d > physical material 2d , bernama
“Player”



21. Klik Player (Physics Material 2D), dibagian menu inspector, friction &
bounces ubah menjadi 0

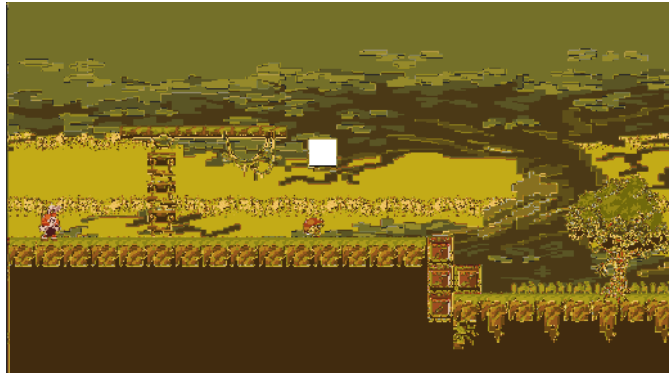


22. Klik Hierarchy pilih layer player idle 1, pada Inspector Cari Rigidbody
2D lalu klik icon untuk membuka box select physhics material 2d , lalu
pilih asset Player yang sudah kita buat tadi

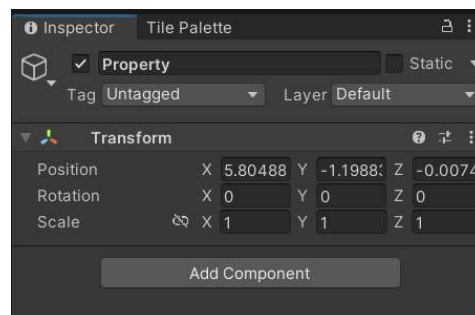




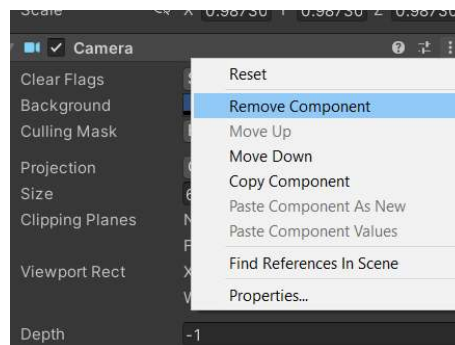
23. Tekan play, maka player bisa melompat dengan menekan spasi



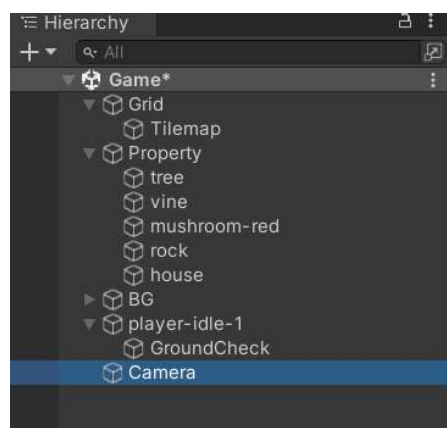
24. Pada Hirarki Property Ubah Inspector pada tag Main camera Menjadi untagged



25. Pada Effect Camera pilih Remove Component

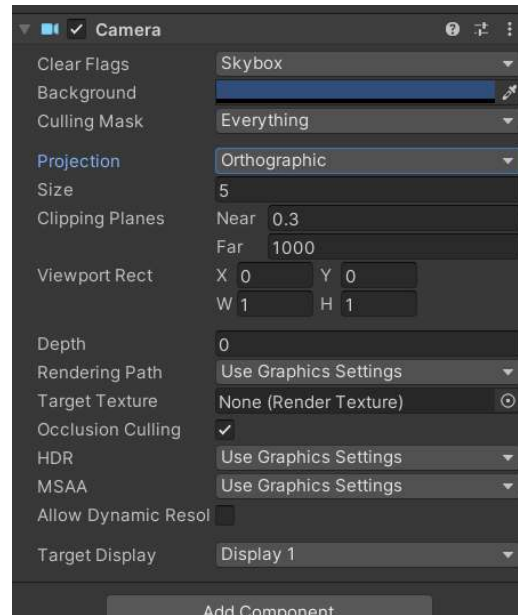


26. Create Empty pada Hirarki, dan Rename Menjadi Camera





27. Sesuaikan Setting Layer Camera seperti gambar dibawah ini



28. Buat file script baru di folder Script dengan nama "CameraFollow"



29. Lalu tuliskan script berikut ini

```
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()
```



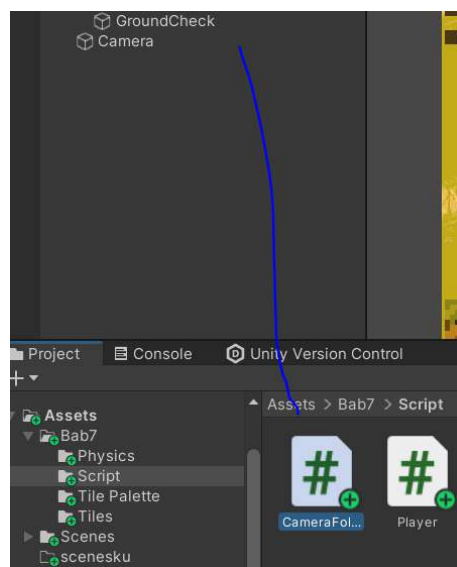
```
{
    return Mathf.Abs(transform.position.x -
player.position.x) > xMargin;
}

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);
}
}
```

30. Drag & drop script CameraFollow Kedalam Layer Camera

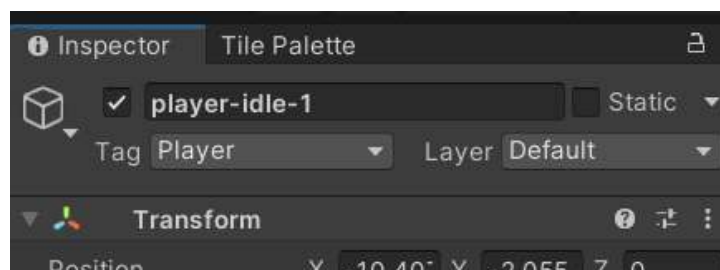




31. Lalu klik pada camera, buka inspector Pada bagian Camera Follow (Script) Ubah Bagian Max X dan Max Y nya



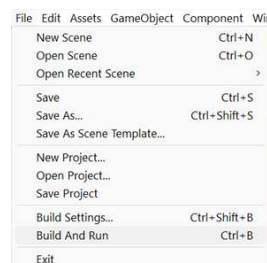
32. Ubah tag di player-idle-1 Untagged menjadi "Player"



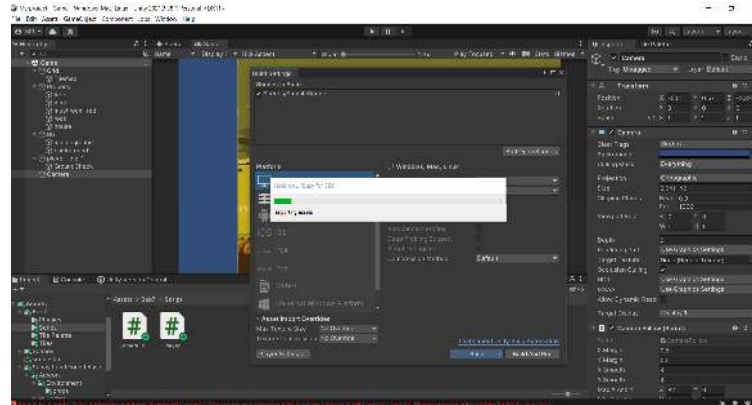
33. Tekan play untuk menjalankan, maka sekarang kamera akan mengikuti pergerakan karakter



34. Pergi ke menu File kemudian pilih Build Setting (Ctrl + Shift + B)



35. Pada Setting Build ini pilih PC, Mac & Linux, Tekan Build, pastikan pada menu Scene in Build berada pada project Tugas Kalian pilih dimana project disimpan dan tunggu hasilnya



B. Quiz

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class CameraFollower : MonoBehaviour
{
    [SerializeField] private Transform player;
    void update() {
        transform.position = new Vector3 (player.position.x,
        transform.position.y, transform.position.z);
    }
}
```

Penjelasan

using System.Collections dan using System.Collections.Generic digunakan untuk mengimpor koleksi generik di C#. using UnityEngine digunakan untuk mengimpor semua kelas dan fungsi yang terkait dengan Unity. kemudian Mendefinisikan kelas CameraFollower yang merupakan turunan dari MonoBehaviour, kelas dasar untuk semua skrip yang terpasang pada GameObject di Unity. [SerializeField] memungkinkan variabel player untuk muncul di Inspector Unity, sehingga Anda dapat menyeret dan melepas GameObject yang ingin diikuti kamera. private Transform player mendeklarasikan variabel player bertipe Transform, yang akan menyimpan referensi ke posisi, rotasi, dan skala dari GameObject pemain. void update() adalah fungsi yang dipanggil sekali per frame. Biasanya digunakan untuk logika yang perlu diperiksa atau diperbarui setiap frame.

C. Link Github Pengumpulan

https://github.com/Rachmanullah/2118100_PRAK_ANIGAME.git