



## TUGAS PERTEMUAN: 8

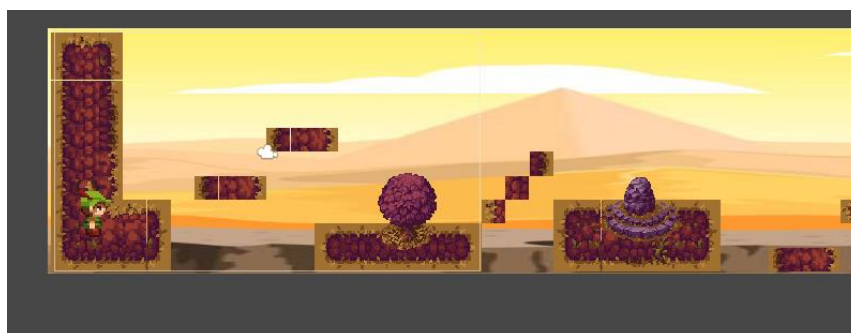
### CAMERA & CHARACTER MOVEMENT

NIM	:	2118076
Nama	:	Nabila Paramita Hapsari
Kelas	:	B
Asisten Lab	:	Maria Avrilliana (2218096)

#### 7.1 Tugas 1 : Membuat Tilemap

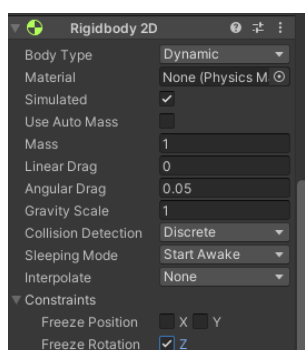
##### A. Membuat Pergerakan Layer

1. Buka file proyek Unity sebelumnya pada bab 7 untuk digunakan kembali Tambahkan player bernama player-id, pilih yang idle, Import kedalam Hirarki



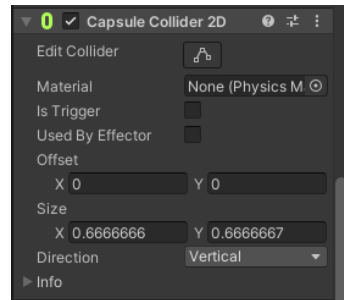
Gambar 7.1 Membuka Projek Unity

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



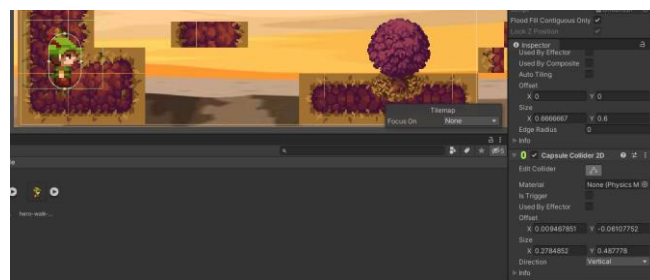
Gambar 7.2 *Freeze Rotation Z*

3. Lalu tambahkan komponen *Capsule Colider* di *player-idle-1*, lalu klik icon sebelah kanan edit colider



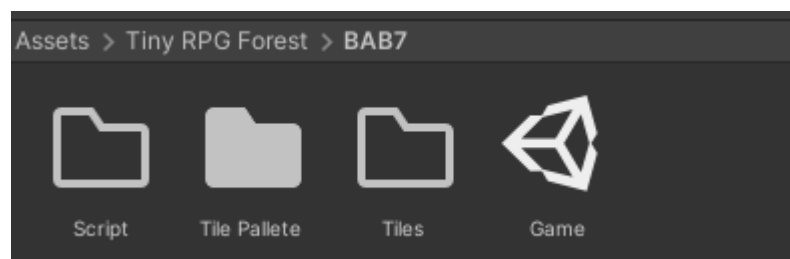
Gambar 7.3 Capsule Collider

4. Lalu cockan garis oval degan karakternya atau bisa di inputkan Offset X, Y dan juga Size X, Y nya



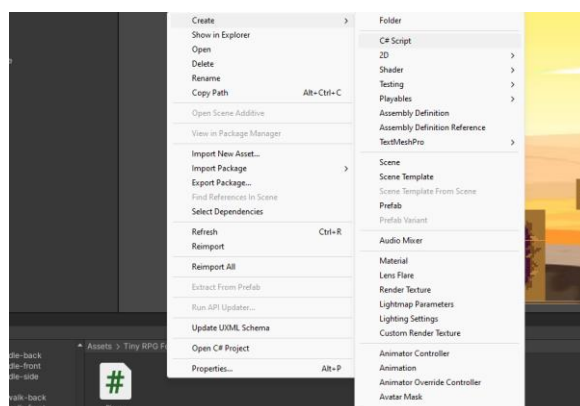
Gambar 7.4 Capsule Collider

5. Buka Folder BAB7, lalu buat folder baru bernama Script



Gambar 7.5 Membuat folder Script

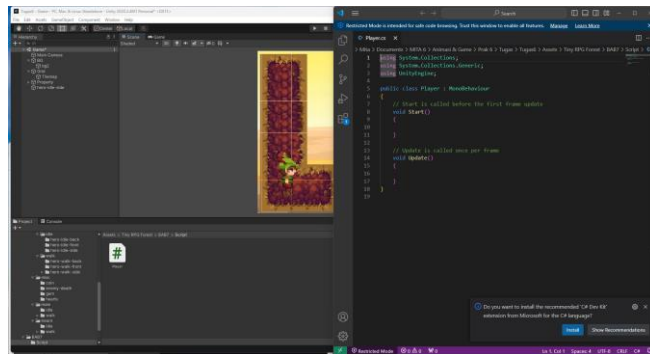
6. Masuk kedalam folder Script, lalu buat C# Script, beri nama Player



Gambar 7.6 Membuat Script



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



Gambar 7.7 Script Player

8. Masukan *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;

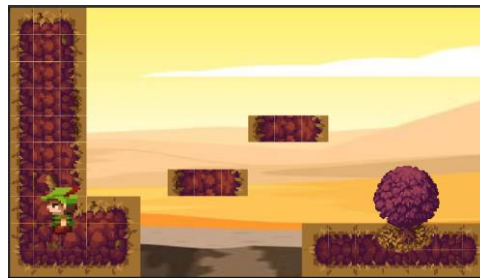
        if (facingRight && dir < 0)
        {
```



```
// ukuran player
transform.localScale = new Vector3(-4, 4, 4);
facingRight = false;
}

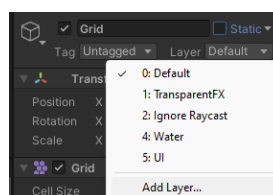
else if (!facingRight && dir > 0)
{
    // ukuran player
    transform.localScale = new Vector3(4, 4, 4);
    facingRight = true;
}
#endregion
}
```

9. 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



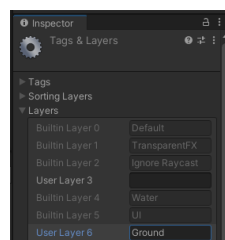
Gambar 7.8 Tampilan Hasil

10. Untuk membuat player loncat menggunakan spasi, kita perlu membuat *GroundCheck* dengan cara, klik Grid pada Hierarchy, pergi ke inspector, pilih Layer, Klik Add Layer.



Gambar 7.9 Layer Ground

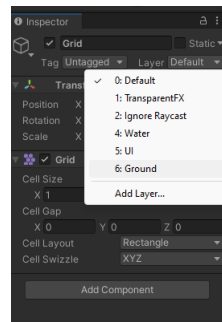
11. Lalu isi “Ground” pada user layer 6.



Gambar 7.10 Layer Ground

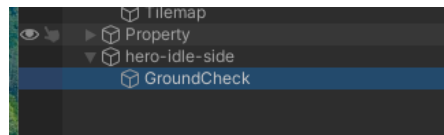


12. Ubah Layer menjadi Ground, jika muncul pop up Change Layer, klik yes saja



Gambar 7.11 *Layer Ground*

13. Klik kanan pada *player-idle-1*, lalu *Create empty*, beri nama *GorundCheck*



Gambar 7.12 *Layer Ground*

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



Gambar 7.13 *Layer Ground*

15. 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;  
  
[SerializeField] bool isGrounded; // +  
bool facingRight;
```

16. Buat void ground check dibawah void fixedUpdate & tambahkan GroundCheck(); pada void fixedUpdate.

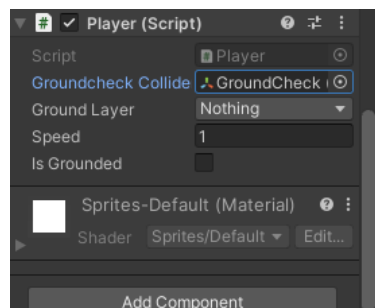
```
void FixedUpdate()  
{
```



```
GroundCheck();
Move(horizontalValue);
}

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

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



Gambar 7. 14 Layer Ground

18. Lalu untuk membuat player melompat tambahkan script berikut

```
[SerializeField] Transform groundcheckCollider;
[SerializeField] LayerMask groundLayer;
const float groundCheckRadius = 0.2f; // +
[SerializeField] float speed = 1;
[SerializeField] float jumpPower = 100;

float horizontalValue;

[SerializeField] bool isGrounded; // +
bool facingRight;
bool jump;
```

19. Tambahkan juga script berikut di bagian void update.

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

20. Tambahkan juga jump pada parameter Move.

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

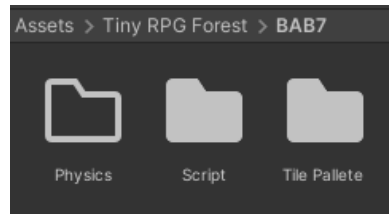
21. Tambahkan script berikut pada void Move.

```
bool jumpflag
```



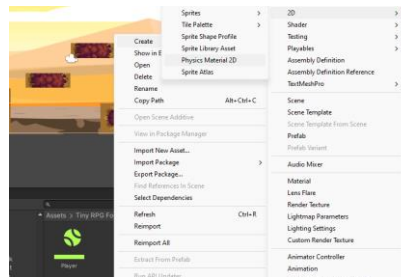
```
if(isGrounded && jumpflag)
{
    isGrounded = false;
    jumpflag = false;
    rb.AddForce(new Vector2(0f, jumpPower));
}
```

22. Buat folder baru di praktikum bernama “Physics”



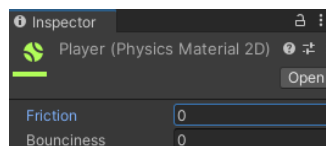
Gambar 7.15 Folder *Physics*

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



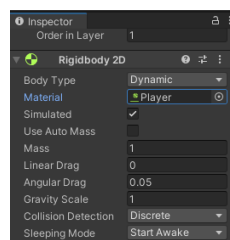
Gambar 7.16 Folder *Physics*

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



Gambar 7.17 Folder *Physics*

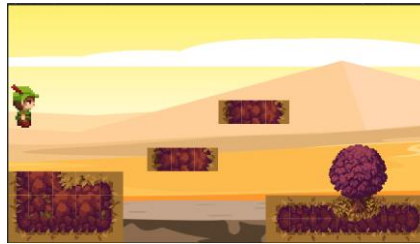
25. Klik Hierarchy pilih layer player idle 1, pada Inspector Cari Rigidbody 2D lalu klik icon untuk membuka box select physics material 2d , lalu pilih asset Player yang sudah kita buat tadi.



Gambar 7.18 *Rigidbody 2D*



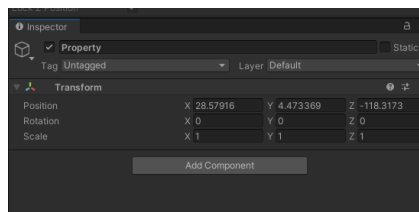
26. Tekan play, maka player bisa melompat dengan menekan spasi.



Gambar 7.19 Tampilan Hasil

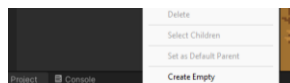
## B. Camera Movement

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



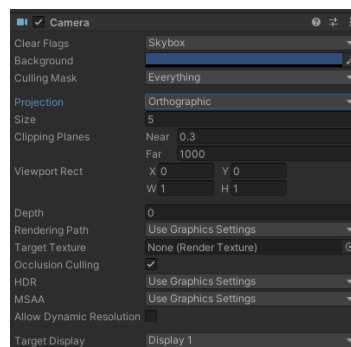
Gambar 7.20 Property

2. Create Empty pada Hirarki, dan Rename Menjadi Camera



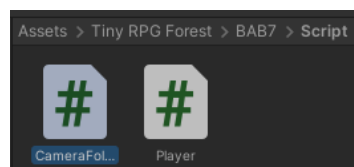
Gambar 7.21 Create Empty

3. Sesuaikan Setting Layer Camera seperti gambar dibawah ini



Gambar 7.21 Camera

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



Gambar 7.22 Camera Follow





## 5. 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);
    }
}
```

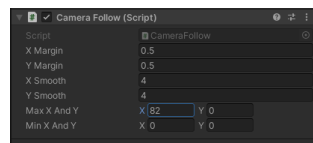


6. Drag & drop script CameraFollow Kedalam Layer Camera



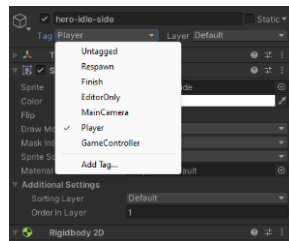
Gambar 7.23 Camera Follow

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



Gambar 7.24 Camera Follow

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



Gambar 7.25 player

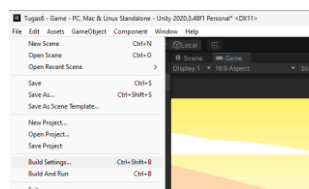
9. Hasil Tampilan



Gambar 7.26 Tampilan Hasil

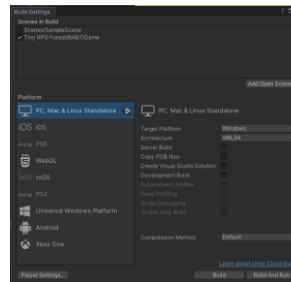
## C. Render

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



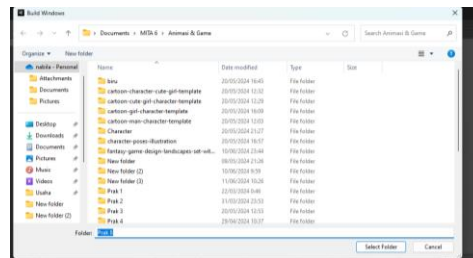
Gambar 7.27 Render

2. Pada Setting Build ini pilih PC, Mac & Linux, Tekan Build, pastikan pada menu Scene in Build berada pada project Tugas Kalian



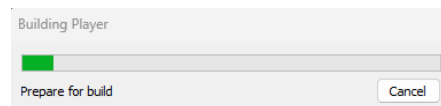
Gambar 7.28 Render

3. Pilih dimana Project disimpan, dan tunggu hasilnya



Gambar 7.29 Render

4. Tunggu



Gambar 7.30 Render

5. Tampilan Hasil



Gambar 7.31 Hasil Tampilan

#### D. Kuis CameraFollow

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

public class CameraFollow : 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** :  
Import namespace yang sering digunakan dalam pengembangan game.

**using UnityEngine** : Import namespace Unity, yang diperlukan untuk mengakses API Unity.

**public class CameraFollow : MonoBehaviour** : Deklarasi kelas CameraFollow yang mewarisi dari MonoBehaviour.

**[SerializeField] private Transform player;** : Deklarasi variabel player dengan type Transform. Transform mengacu pada komponen transform dari pemain yang akan diikuti oleh kamera.

**void Update()** : Method ini dipanggil sekali per frame, sehingga cocok untuk tugas yang memerlukan update seperti mengikuti pemain.

**transform.position = new Vector3(player.position.x, transform.position.y, transform.position.z);** : Posisi kamera diatur ulang tiap frame. Posisi X kamera diatur agar sama dengan posisi X pemain, sementara posisi Y dan Z kamera tetap sama dengan posisi saat ini. Sehingga kamera hanya mengikuti pergerakan pemain di sumbu X.

#### E. Link Github

[https://github.com/NabilaParamitaHapsari/2118076\\_PRAK\\_ANIGAME.git](https://github.com/NabilaParamitaHapsari/2118076_PRAK_ANIGAME.git)