



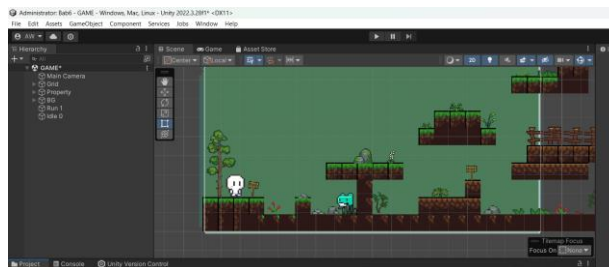
## TUGAS PERTEMUAN: 8 CAMERA & CHARACTER MOVEMENT

NIM	:	2118077
Nama	:	Abdul Wahid
Kelas	:	B
Asisten Lab	:	MARIA AVRILIANA SURAT LELAONA (2218096)
Baju Adat	:	Pangasi Baju Adat sunda (jawa barat)
Referensi	:	<a href="https://akcdn.detik.net.id/community/media/visual/2022/04/20/pakaian-adat-jawa-barat-5_43.jpeg?w=700&amp;q=90">https://akcdn.detik.net.id/community/media/visual/2022/04/20/pakaian-adat-jawa-barat-5_43.jpeg?w=700&amp;q=90</a>

### 8.1 Tugas 8 : Membuat Character Movement, Detect Ground, Jumping, & Camera Movement

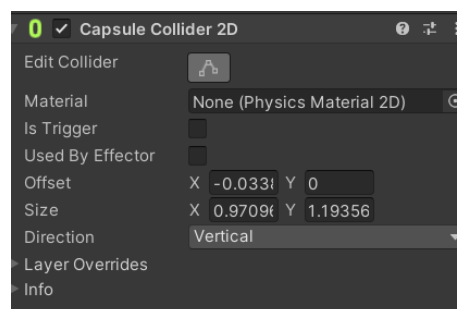
#### A. Membuat Pergerakan Player

1. Buka kembali file proyek Unity sebelumnya pada tugas bab 7 untuk digunakan kembali.



Gambar 8.1 Proyek Bab 7

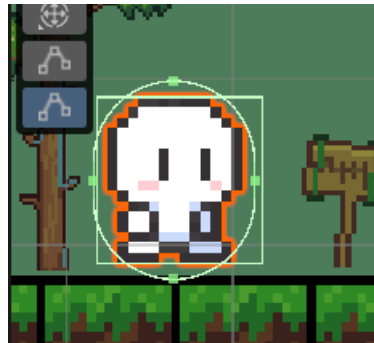
2. Lalu tambahkan komponen Capsule Colider di player-idle-1, lalu klik icon sebelah kanan edit colider.



Gambar 8.2 Capsule Collider 2D

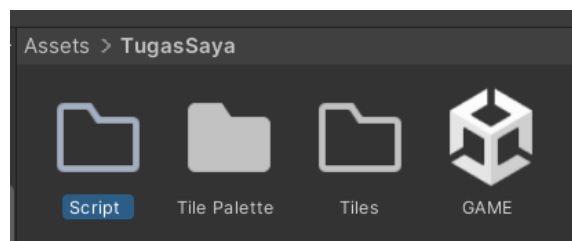


3. Selanjutnya cockan garis oval degan karakternya atau bisa di inputkan Offset X, Y dan juga Size X, Y nya.



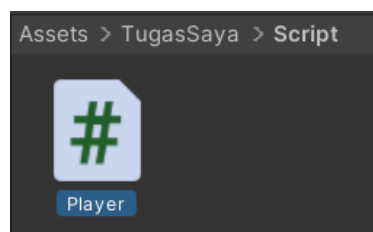
Gambar 8.3 Offset X, Y

4. Kemudian buka Folder TugasSaya, lalu bikin folder baru bernama Script.



Gambar 8.4 Folder Script

5. Lalu masuk kedalam folder Script, lalu buat C# Script, beri nama Player.



Gambar 8.5 Script Player

6. Selanjutnya drag & drop script player kedalam Hirarki player lalu klik 2x pada script player maka akan masuk kedalam text editor seperti ini kemudian Masukan source code dibawah ini.

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

public class Player : MonoBehaviour
{
    Rigidbody2D rb;
    [SerializeField] Transform groundcheckCollider;
    [SerializeField] LayerMask groundLayer;

    const float groundCheckRadius = 0.2f; // +
    [SerializeField] float speed = 1;
```



```
[SerializeField] float jumpPower = 1000;
float horizontalValue;

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

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

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)
```

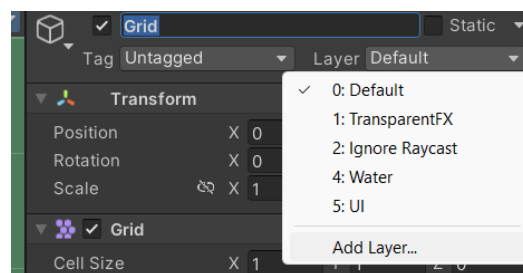


```
{
    // ukuran player
    transform.localScale = new Vector3(-1, 1, 1);
    facingRight = false;
}

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

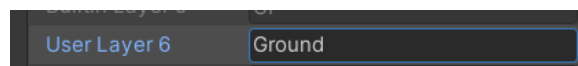
#endregion
}
```

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



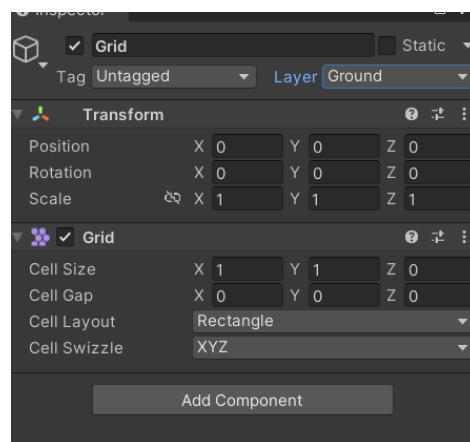
Gambar 8.6 Add Layer

8. Kemudian isi “Ground” pada User Layer 6.



Gambar 8.7 Ground

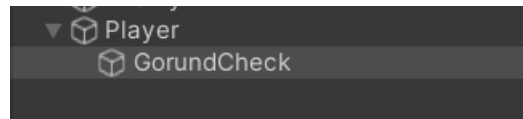
9. Lalu ubah Layer menjadi Ground.



Gambar 8.8 Layer Ground

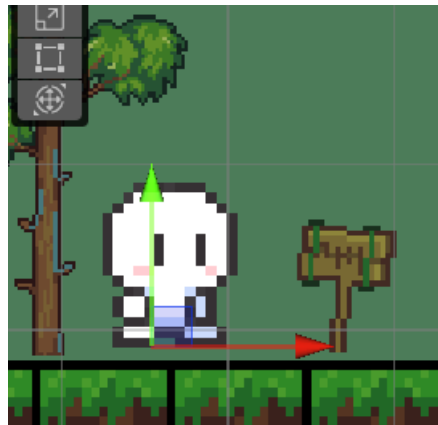


10. Selanjutnya klik kanan pada player, lalu Create empty, beri nama GroundCheck.



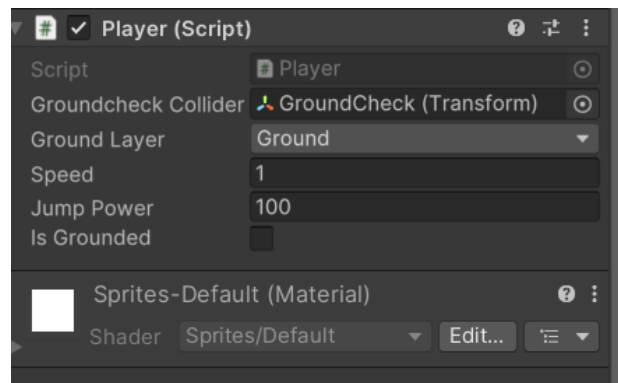
Gambar 8.9 GroundCheck

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



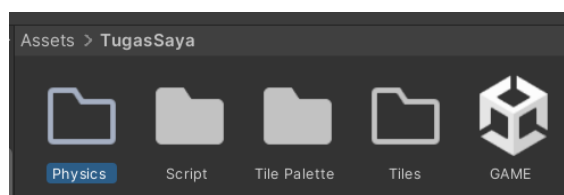
Gambar 8.10 Move GroundCheck

12. Selanjutnya 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.



Gambar 8.11 Player Script

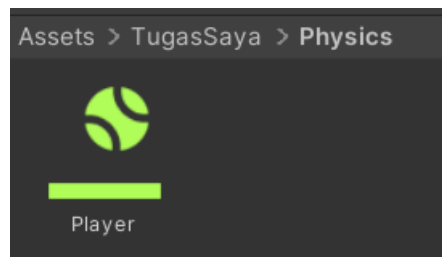
13. Lalu buat folder baru di TugasSaya bernama “Physics”.



Gambar 8.12 Physics

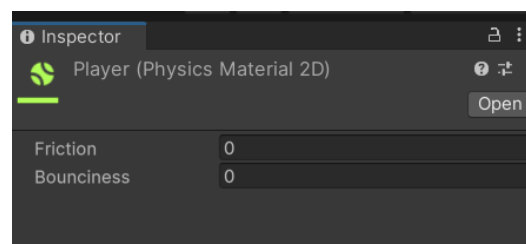


14. Selanjutnya didalam folder Pyshics create > 2d > physical material 2d ,  
berinama “Player”.



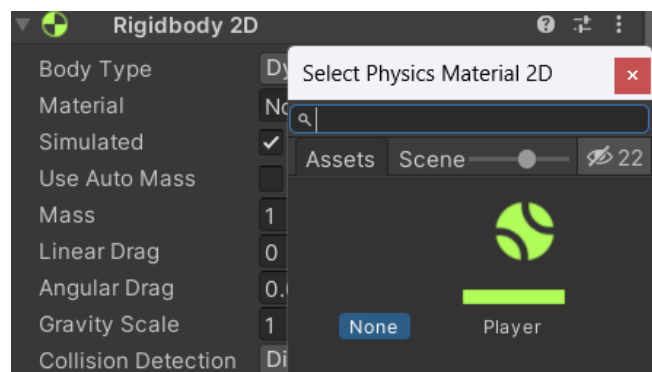
Gambar 8.13 Physics Player

15. Kemudian klik Player (Physics Material 2D), dibagian menu inspector,  
friction & bounces ubah menjadi 0.



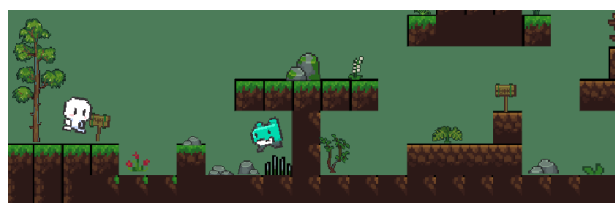
Gambar 8.14 Physics Material 2D

16. Lalu 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 8.15 Select Physics Material 2D

17. Selanjutnya tekan play, maka player bisa melompat dengan menekan  
spasi.

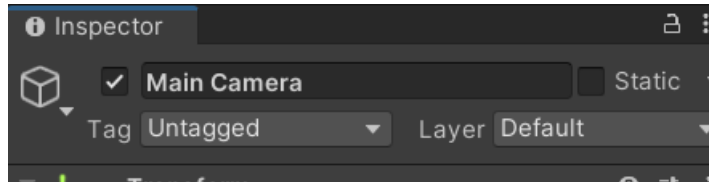


Gambar 8.16 Play



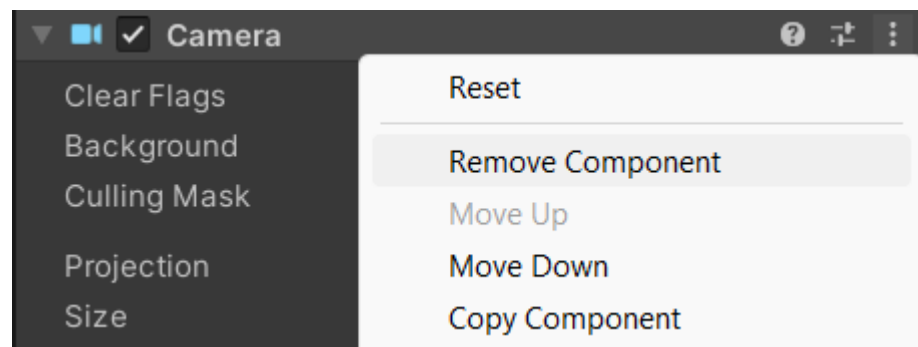
## B. Camera Movement

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



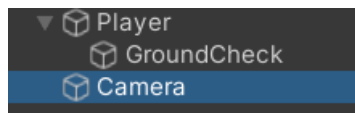
Gambar 8.17 Main camera

2. Lalu pada Effect Camera pilih Remove Component.



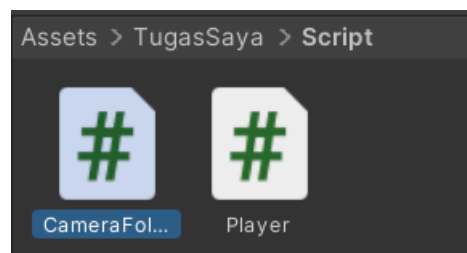
Gambar 8.18 Remove Component

3. Selanjutnya Create Empty pada Hirarki, dan Rename Menjadi Camera.



Gambar 8.19 Camera

4. Kemudian sesuaikan Setting Layer Camera seperti gambar dibawah ini.
5. Lalu buat file script baru di folder Script dengan nama "CameraFollow".



Gambar 8.20 CameraFollow

6. Selanjutnya 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);
}
}
```

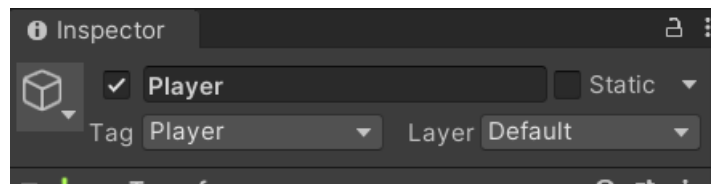
7. Kemudian drag & drop script CameraFollow Kedalam Layer Camera lalu klik pada camera, buka inspector Pada bagian Camera Follow (Script) Ubah Bagian Max X dan Max Y nya.





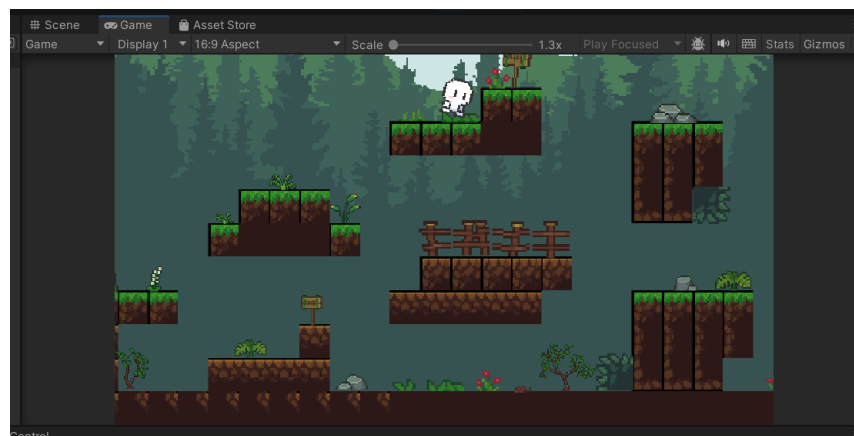
Gambar 8.21 Camera Follow Script

8. Selanjutnya ubah tag di player Untagged menjadi "Player".



Gambar 8.22 Tag Player

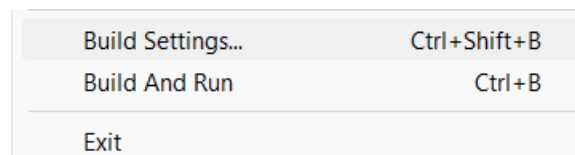
9. Kemudian tekan play untuk menjalankan, maka sekarang kamera akan mengikuti pergerakan karakter.



Gambar 8.23 play

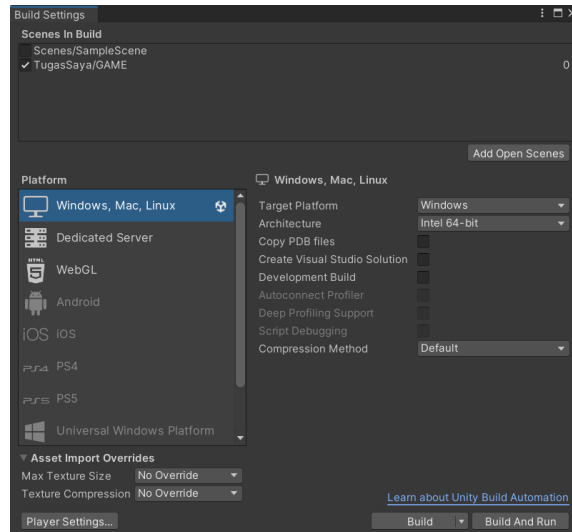
### C. Render

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



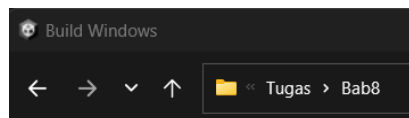
Gambar 8.24 Build Settings

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



Gambar 8.25 Setting Build

3. Selanjutnya pilih dimana Project disimpan, dan tunggu hasilnya.



Gambar 8.26 Folder Projek

4. Kemudian ini hasil nya.



Gambar 8.27 Hasil Render

## 8.2 KUIS

```
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);
    }
}
```



```
}  
}
```

Analisa Source Code :

Soutce Code Diatas berada pada Script C# CameraFollow di Unity membuat kamera mengikuti pergerakan horizontal pemain dengan memperbarui x-coordinate kamera setiap frame agar sesuai dengan x-coordinate pemain, sementara y dan z-coordinate kamera tetap konstan. Variabel player bertipe Transform diserialisasi untuk diatur melalui Unity Editor.