



TUGAS PERTEMUAN: 8

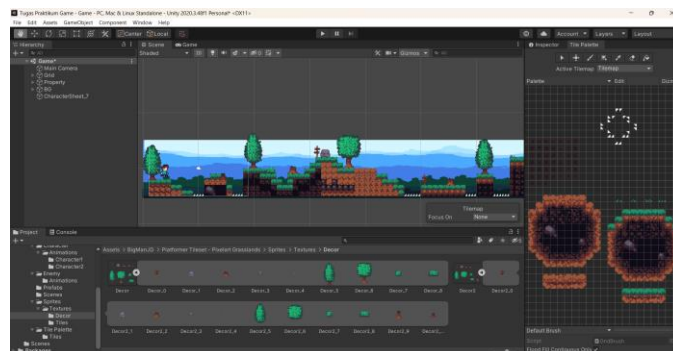
Camera & Character Movement

NIM	:	2118099
Nama	:	Indra Syaputra Stiansyah
Kelas	:	E
Asisten Lab	:	Nur Aria Hibnastiar (2118078)
Baju Adat	:	
Referensi	:	

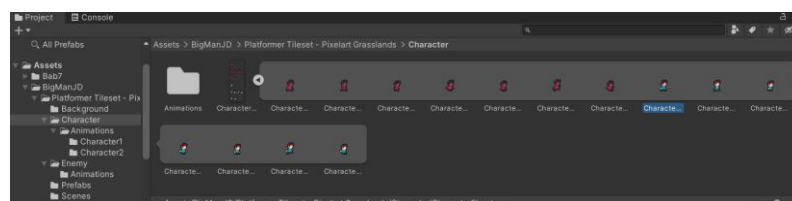
8.1 Tugas 8 : Membuat Camera & Character Movement

A. Membuat Pergerakan Player

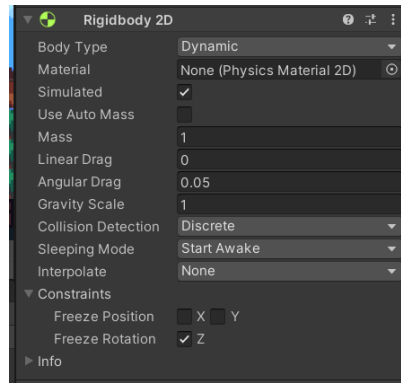
1. Buka Project unity sebelumnya yang telah diimpor dengan asset dari unity asset store bisa dilihat dari gambar dibawah ini.



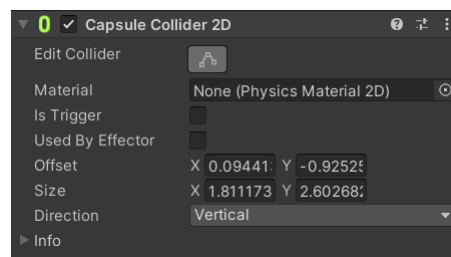
2. Tambahkan player Bernama character sheet_7, Import kedalam Hirarki.



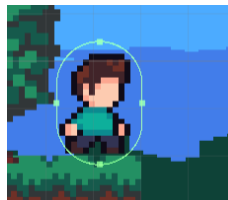
3. Klik character sheet_7 tambahkan Component Rigidbody 2D, sesuaikan settingannya seperti gambar berikut, Centang pada Freeze Rotation Z



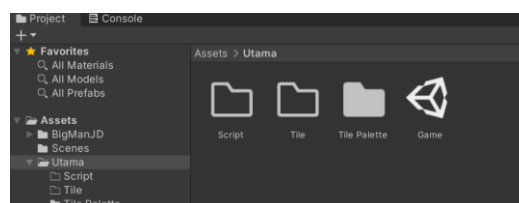
4. Lalu tambahkan komponen Capsule Colider di player-idle-1, lalu klik icon sebelah kanan edit colider.



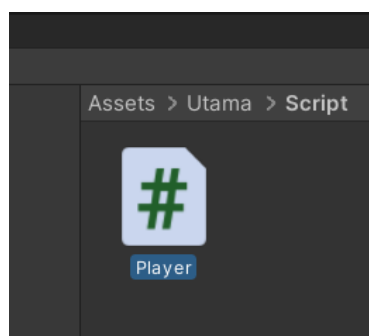
5. Lalu cockan garis oval degan karakternya atau bisa di inputkan Offset X, Y dan juga Size X, Y nya.



6. Buka Folder utama, lalu bikin folder baru bernama Script.

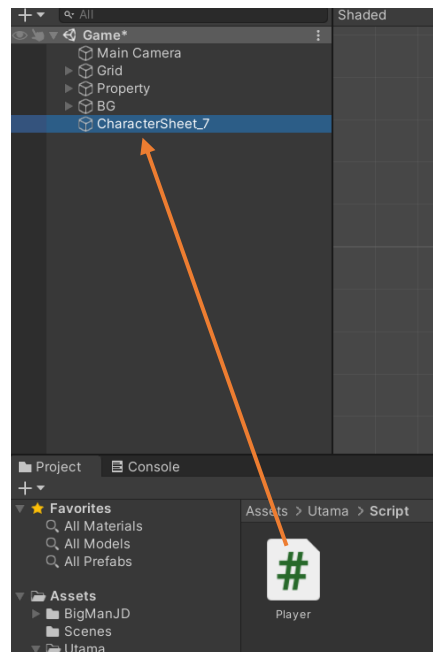


7. Masuk kedalam folder Script, lalu buat C# Script, beri nama Player.





8. Drag & drop script player kedalam Hirarki character sheet_7, lalu klik 2x pada script player maka akan masuk kedalam text editor.



9. 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)
    {

```



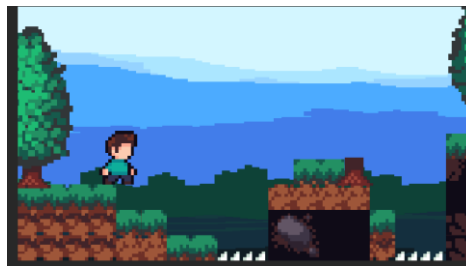
```
#region gerak kanan kiri
float xVal = dir * speed * 100 *
Time.fixedDeltaTime;
Vector2 targetVelocity = new Vector2(xVal,
rb.velocity.y);
rb.velocity = targetVelocity;

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

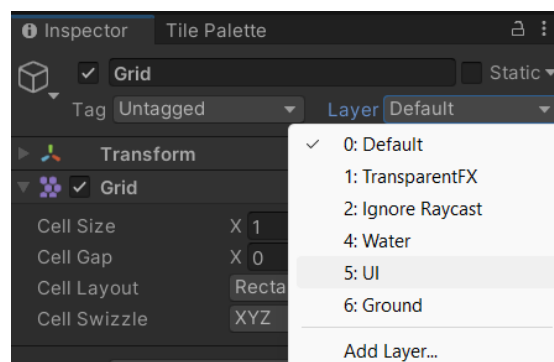
else if (!facingRight && dir > 0)
{
    // ukuran player
    transform.localScale = new Vector3(1, 1, 1);
    facingRight = true;
}

#endregion
}
}
```

10. 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.

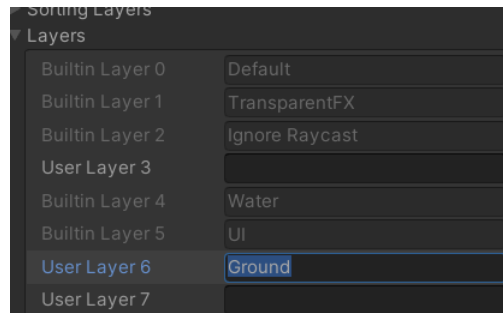


11. Untuk membuat player loncat menggunakan spasi, kita perlu membuat GroundCheck dengan cara, klik Grid pada Hierarchy, pergi ke inspector, pilih Layer, Klik Add Layer.

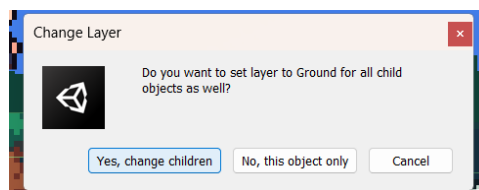




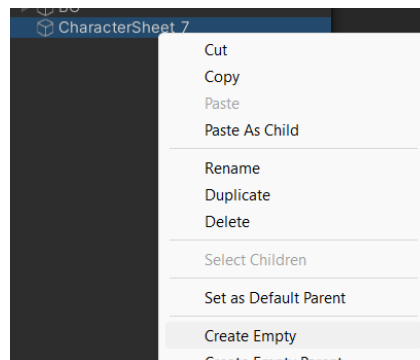
12. Lalu isi “Ground” pada User Layer 6.



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



14. Klik kanan pada character sheet_7, lalu Create empty, beri nama GorundCheck.



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



16. 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;

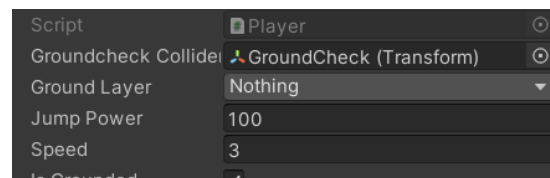
[SerializeField] bool isGrounded; // +
bool facingRight;
```

17. 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.positi
on, groundCheckRadius, groundLayer);
    if (colliders.Length > 0)
        isGrounded = true;
}
```

18. Klik character sheet_7, lalu ke inspector ke effect Player script di bagian “Goruncheck collider” tekan icon lalu pilih yang GorundCheck Transform, dan pada Ground Layer pilih Ground.



19. Lalu untuk membuat player melompat tambahkan script berikut.

```
[SerializeField] float jumpPower = 100;

bool jump;
```

20. Tambahkan juga script berikut di bagian void update

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

21. Tambahkan juga jump pada parameter Move.

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

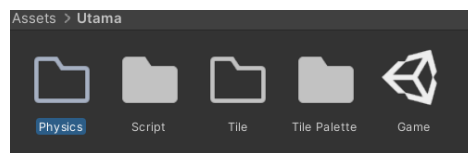


22. Tambahkan script berikut pada void Move

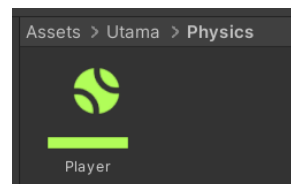
```
reference  
void Move(float dir, bool jumpflag)  
{  
    #region gerak kanan kiri
```

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

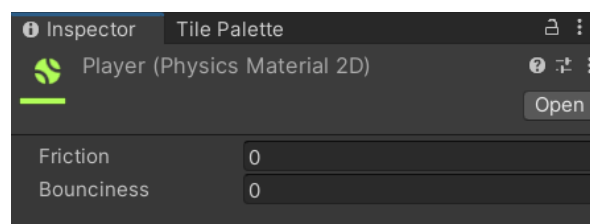
23. Buat folder baru di utama bernama “Physics”.



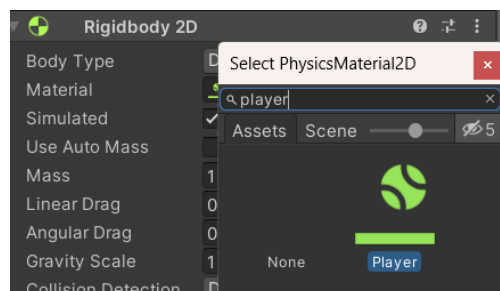
24. Didalam folder Pyshics create > 2d > physical material 2d , bernama “Player”.



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

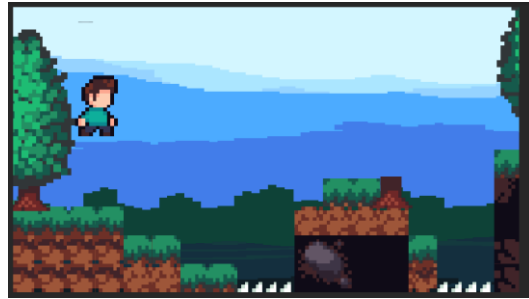


26. Klik Hierarchy pilih layer character sheet_7, pada Inspector Cari Rigidbody 2D lalu klik icon untuk membuka box select physics material 2d , lalu pilih asset Player yang sudah kita buat tadi.



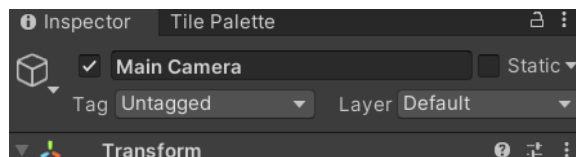


27. Tekan play, maka player bisa melompat dengan menekan spasi.

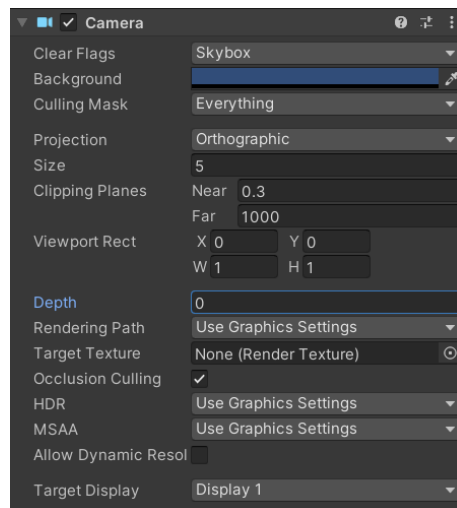


B. Camera Movement

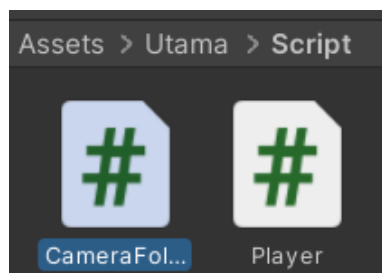
1. Pada Hirarki Property Ubah Inspector pada tag Main camera Menjadi untagged.



2. Sesuaikan Setting Layer Camera seperti gambar dibawah ini.



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



4. 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;
    }

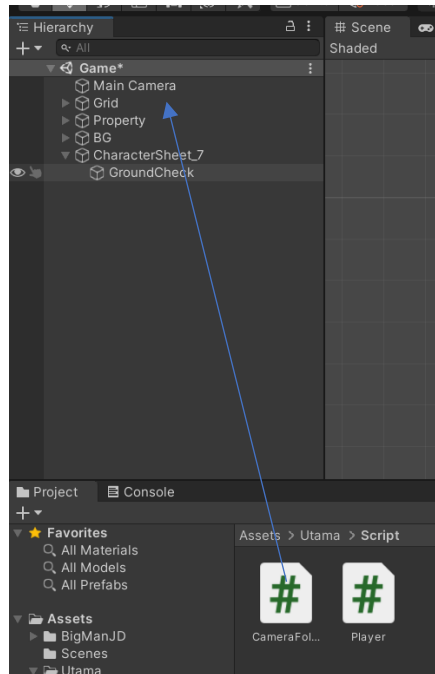
    bool CheckXMargin()
    {
        return Mathf.Abs(transform.position.x -
player.position.x) > xMargin;
    }

    bool CheckYMargin()
    {
        return Mathf.Abs(transform.position.y -
player.position.y) > yMargin;
    }

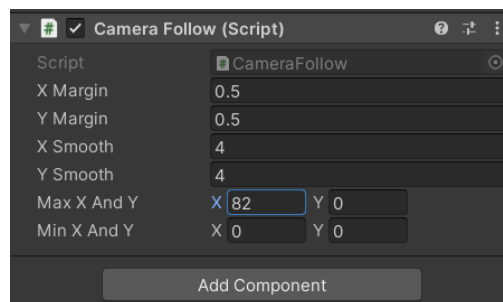
    void FixedUpdate()
    {
        TrackPlayer();
    }

    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);
    }
}
```

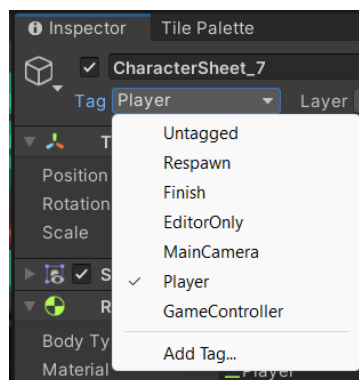
5. Drag & drop script CameraFollow Kedalam Layer Camera



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



7. Ubah tag di character sheet_7 Untagged menjadi "Player"



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



C. Quiz

```
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 diatas digunakan dalam Unity untuk membuat kamera mengikuti objek pemain. Script ini mengimpor namespace yang diperlukan seperti System.Collections, System.Collections.Generic, dan UnityEngine. Kelas CameraFollow mewarisi dari MonoBehaviour, yang memungkinkan script ini untuk dipasang pada GameObject di Unity. Variabel player bertipe Transform diberi atribut [SerializeField] agar bisa diatur di Inspector Unity tanpa perlu dijadikan public. Pada metode Update(), posisi kamera diatur ulang setiap frame agar sumbu x mengikuti posisi x pemain, sementara sumbu y dan z tetap pada posisi kamera saat ini. Pastikan Transform pemain telah diassign di Inspector agar script berfungsi dengan benar.

D. Link Github Pengumpulan

https://github.com/IndraSyaputraS/2118099_PRAK_ANIGAME