



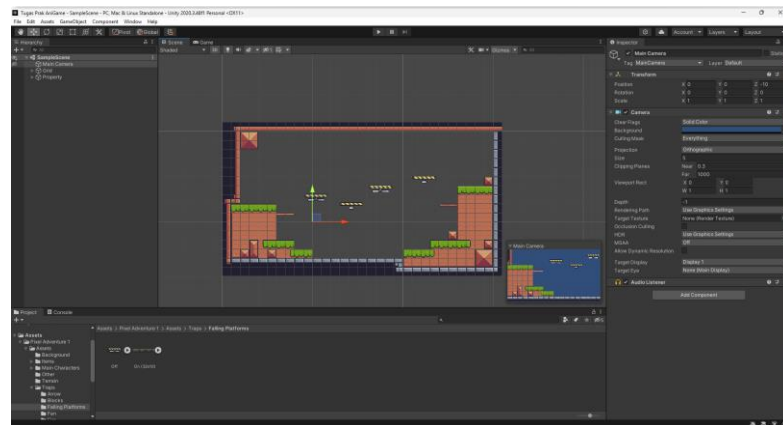
TUGAS PERTEMUAN: 8

Membuat Character Movement, Detect Ground, Jumping, & Camera Movement

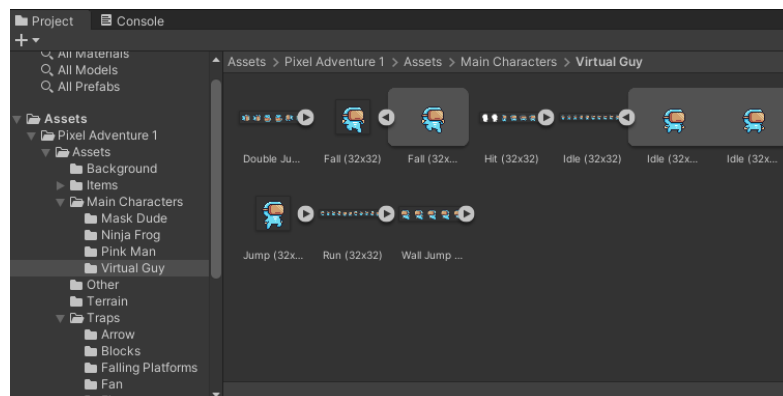
NIM	:	2118014
Nama	:	Muhammad Ridho Putra Syalabi
Kelas	:	D
Asisten Lab	:	NAUFAL DHIAURRAFIF (2218059)

8.1 Tugas 1 : Membuat Character Movement

1. Buka Project sebelumnya

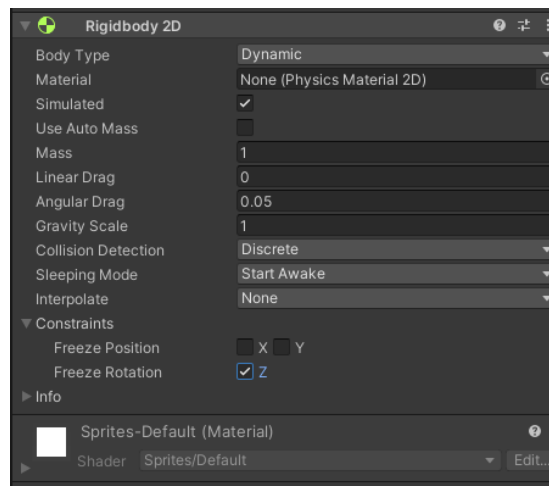


2. Tambahkan *Player*, pilih yang *idle*, import ke dalam hirarki

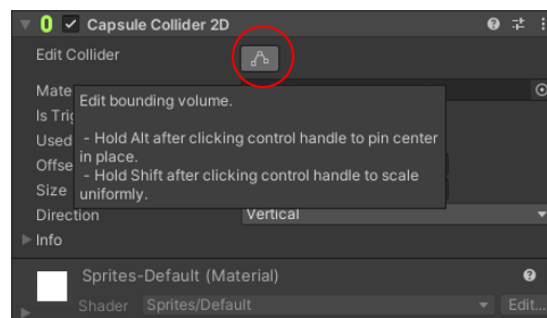




3. Pilih *player_idle_0* lalu tambahkan *Component Rigidbody 2D*. Centang pada Freeze Rotation Z



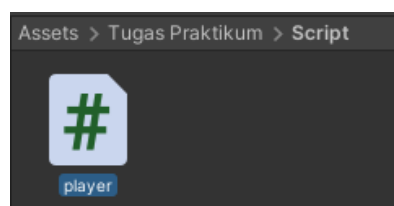
4. Kemudian tambahkan komponen *Capsule Colider* di *player-idle-0*



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



6. Kemudian buat folder baru untuk menyimpan Script, kemudian buat c# script untk player.





7. Kemudian *drag and drop* ke *player_idle_0* script *player*, kemudian tambahkan code berikut.

```
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(-5, 5, 1);
            facingRight = false;
        }

        else if (!facingRight && dir > 0)
        {
            // ukuran player
            transform.localScale = new Vector3(5, 5, 1);
            facingRight = true;
        }

        #endregion
    }
}
```

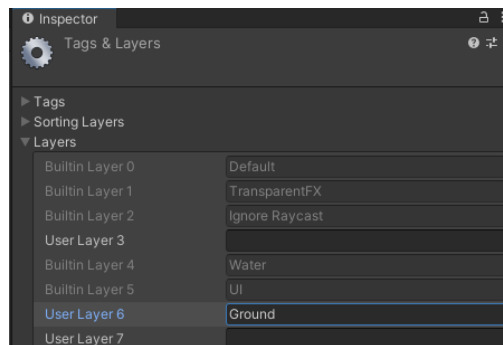
8. Asd



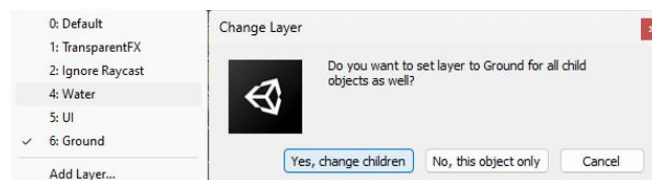
9. Asd
10. Asd

8.2 Tugas 2 : Membuat Detect Ground

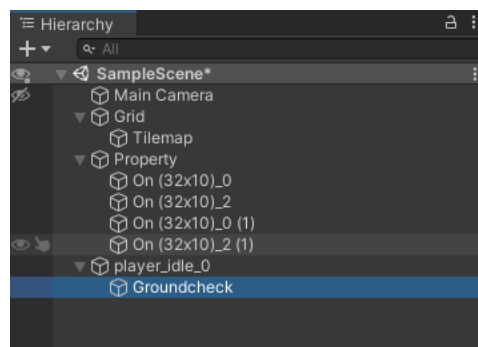
1. Untuk membuat player melompat menggunakan spasi, kita perlu membuat GroundCheck dengan cara, klik Grid pada Hierarchy, pergi ke inspector, pilih Layer, Klik Add Layer, lalu isi *ground* pada *user layer 6*



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

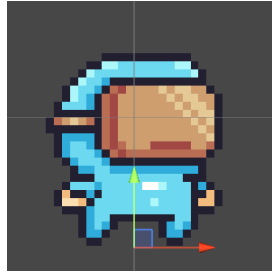


3. Klik kanan pada *player_idle_0*, lalu *Create empty*, beri nama *Groundcheck*





4. Klik pada Hirarki *GroundCheck*, lalu gunakan “Move Tools” untuk memindahkan ke bagian bawah *Player*



5. Kemudian pada script player tambahkan code berikut:

```
[SerializeField] Transform groundcheckCollider;
[SerializeField] LayerMask groundLayer;

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

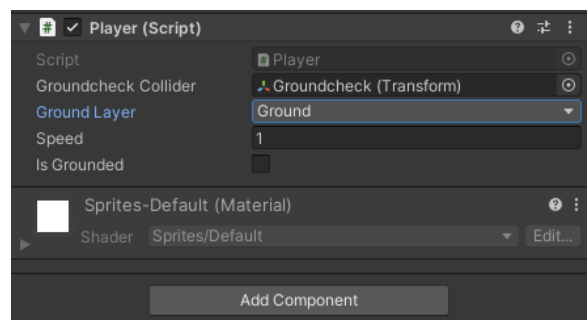
[SerializeField] bool isGrounded; //+
bool facingRight;
```

6. 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;
}
```

7. Klik player_idle_0, lalu ke *inspector* ke *effect Player script* di bagian “Grouncheck collider” tekan icon lalu pilih yang *GorundCheck Transform*, dan pada *Ground Layer* pilih *Ground*.





8.3 Tugas 3 : Membuat Movement Character Jump

1. Tambahkan code berikut agar karakter dapat melompat:

```
[SerializeField] float jumpPower = 100;

bool jump;
```

2. Pada void update, tambahkan code beriku:

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

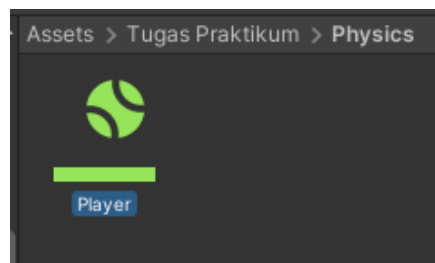
3. Tambahkan parameter jump pada function move yang dipanggil di void fixedupdate()

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

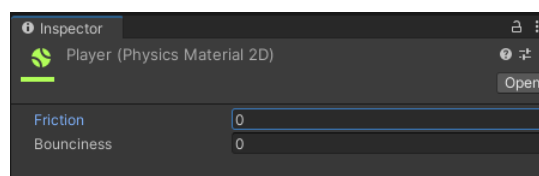
4. Kemudian modifikasi code pada void move()

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

5. Buat folder baru “physics”, Didalam folder Pysics create > 2d > physical material 2d , bernama “Player”.

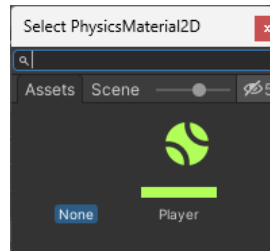


6. Klik Player (Physics Material 2D), dibagian menu inspector, friction & bounces ubah menjadi 0.



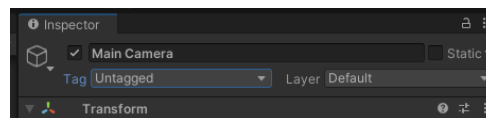


7. Klik Hierarchy pilih *player_idle_0*, pada *Inspector* Cari *Rigidbody 2D* lalu klik *icon* untuk membuka *box select physics* material 2d , lalu pilih *asset Player* yang sudah kita buat tadi.

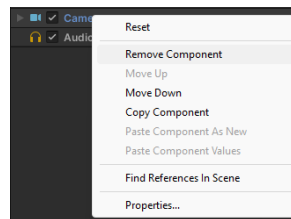


8.4 Tugas 4 : Membuat Camera Movement

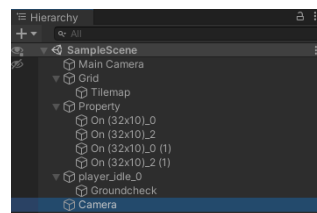
1. Pada Hirarki Property Ubah Inspector pada tag Main camera Menjadi untaged.



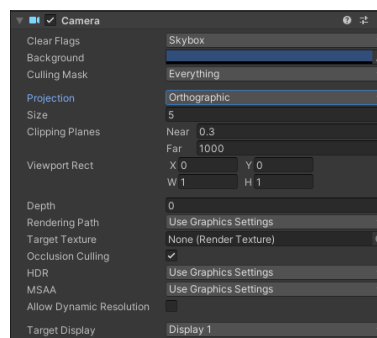
2. Pada Effect Camera pilih Remove Component.



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



4. Tambahkan component camera, kemudian ubah projection ke orthographic





5. Buat scrip baru untuk *CameraFollow*, gunakan code dibawah ini, kemudian Drag & drop script CameraFollow Kedalam Layer Camera

```
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;
    }

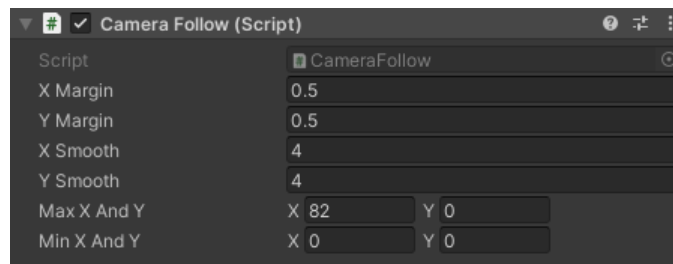
    bool CheckYMargin()    {
        return    Mathf.Abs(transform.position.y
player.position.y) > yMargin;
    }

    void FixedUpdate()
    {
        TrackPlayer();
    }

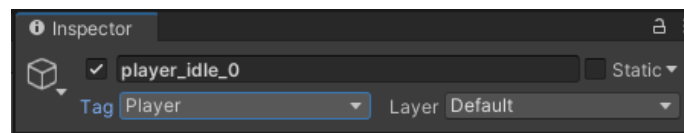
    void TrackPlayer()
    {
        float targetX = transform.position.x;
        float targetY = transform.position.y;
        if (CheckXMargin())
            targetX    =    Mathf.Lerp(transform.position.x,
player.position.x,
            xSmooth * Time.deltaTime);
        if (CheckYMargin())
            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. Lalu klik pada camera, buka inspector Pada bagian Camera Follow (Script)
Ubah Bagian Max X dan Max Y nya.



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





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 :

Kode ini mengimpor pustaka `System.Collections` dan `System.Collections.Generic` untuk koleksi dasar C#, meskipun tidak digunakan dalam kode, serta pustaka `UnityEngine` yang mencakup semua kelas dan fungsi dasar Unity seperti `MonoBehaviour`, `Transform`, dan `Vector3`. Kelas `CameraFollow` bertanggung jawab untuk membuat kamera mengikuti pemain dan diturunkan dari `MonoBehaviour`, yang memungkinkan penggunaan fungsi seperti `Update()` dan interaksi dengan komponen lainnya di Unity. Atribut `[SerializeField]` memungkinkan variabel `player` untuk diedit di Inspector Unity tanpa harus membuatnya `public`, dan variabel `private Transform player` menyimpan referensi ke posisi pemain atau objek yang akan diikuti oleh kamera. Fungsi `Update()` dipanggil setiap frame dan merupakan tempat yang tepat untuk menempatkan logika yang harus diperiksa atau diperbarui secara terus-menerus. Di dalamnya, baris `transform.position = new Vector3(player.position.x, transform.position.y, transform.position.z);` mengatur posisi kamera untuk mengikuti posisi `x` dari pemain sambil mempertahankan posisi `y` dan `z` aslinya.