



TUGAS PERTEMUAN: 8

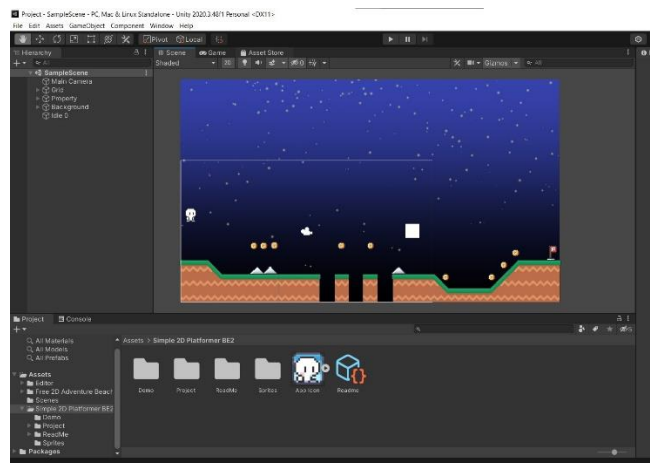
CAMERA & CHARACTER MOVEMENT

NIM	:	2118012
Nama	:	Adrianus Vianto Eban Kia
Kelas	:	A
Asisten Lab	:	Yedija Adya Vesaka (2118056)

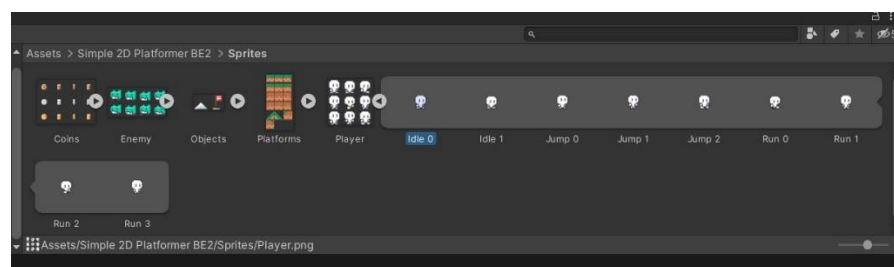
8.1 Tugas 8 : Membuat Pergerakan Player

A. Membuat Pergerakan Player pada Project

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

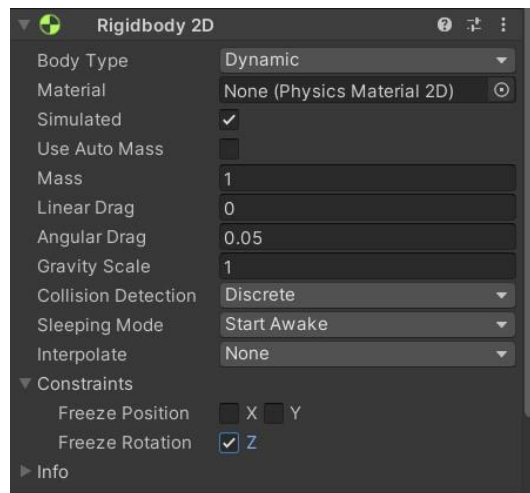


2. Tambahkan player bernama player-id, pilih yang Idle 0, Import kedalam Hirarki





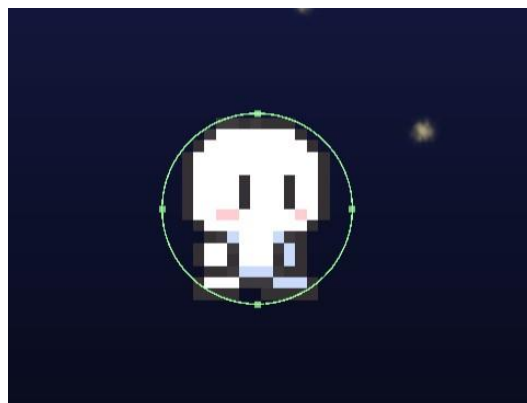
3. Klik Idle 0 tambahkan Component Rigidbody 2D, sesuaikan settingannya seperti gambar berikut, Centang pada Freeze Rotation Z



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

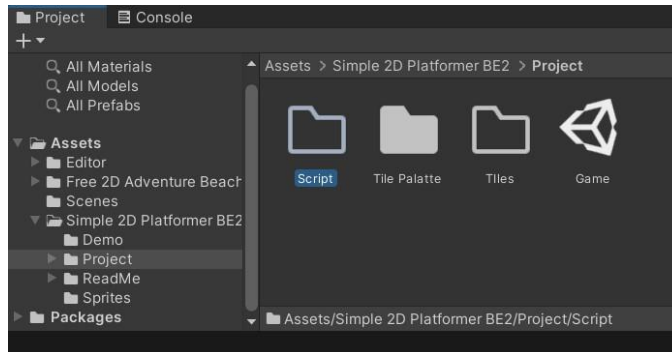


5. Lalu cockan garis oval degan karakternya atau bisa di inputkan Offset X, Y dan juga Size X, Y nya

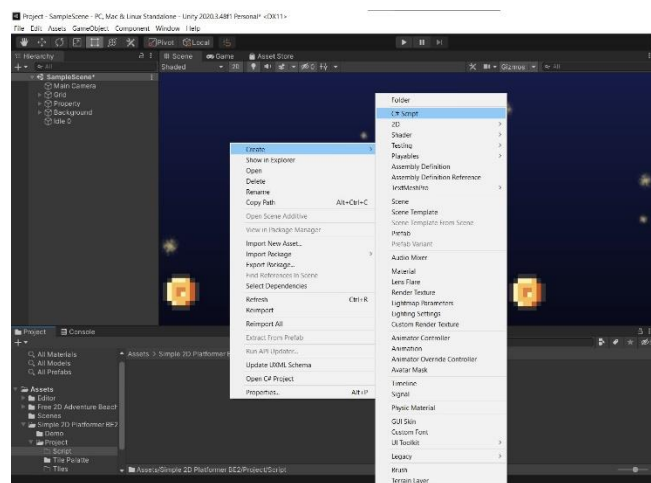




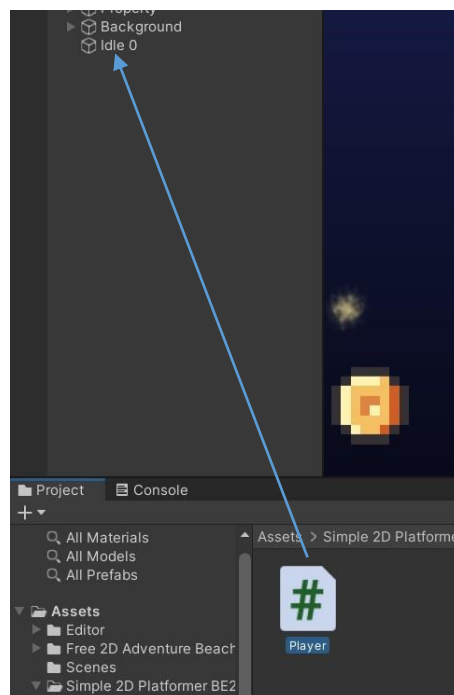
6. Buka Folder praktikum, lalu bikin folder baru bernama Script.



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



8. Drag & drop script player kedalam Hirarki Idle 0, lalu klik 2x pada script player maka akan masuk kedalam text editor seperti ini.





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;

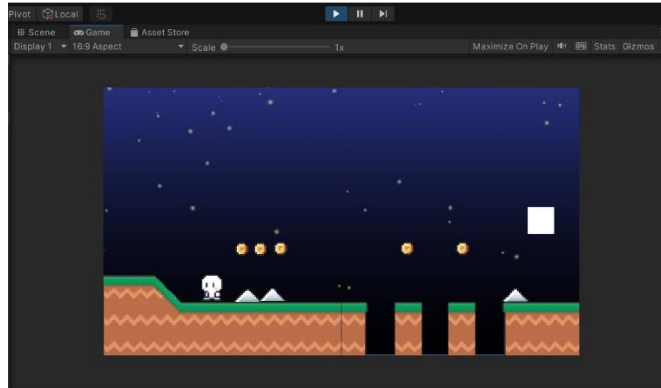
        if (facingRight && dir < 0)
        {
            // ukuran player
            transform.localScale = new Vector3(-1, 1, 1);
            facingRight = false;
        }

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

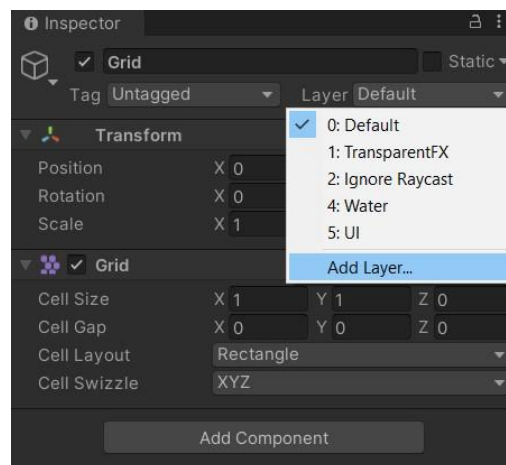
        #endregion
    }
}
```



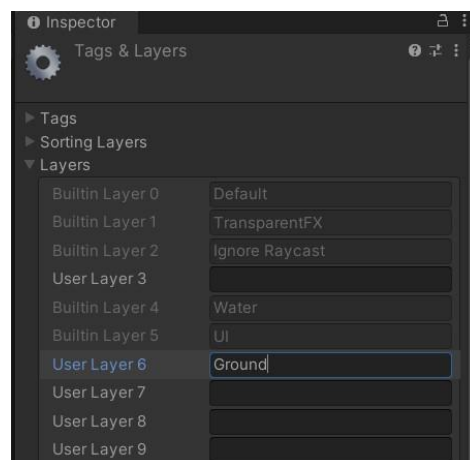
10. Untuk mencoba Source code diatas berhasil, Tekan dikeyboard “a” atau “left arrow” untuk ke arah kiri, tekan “d” atau “right arrow” untuk ke arah kanan.



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.

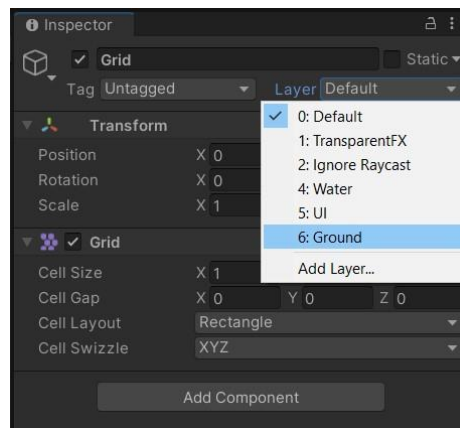


12. Lalu isi “Ground” pada User Layer 6.

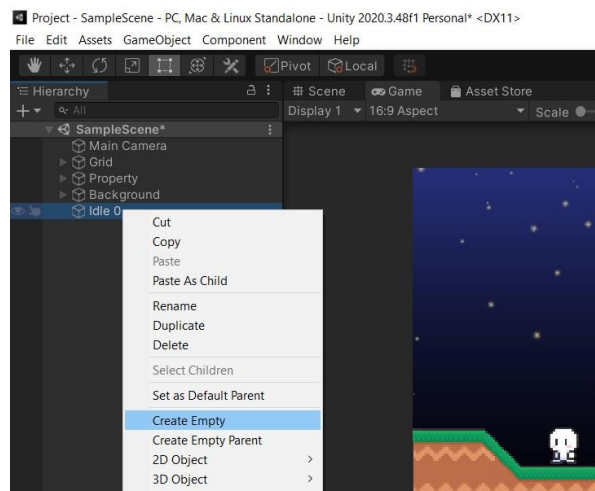




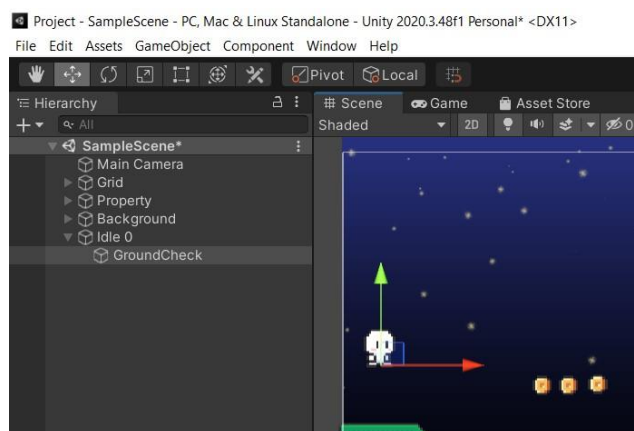
13. Ubah Layer menjadi Ground, jika muncul pop up Change Layer, klik yes saja.



14. Klik kanan pada Idle 0, 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 dibawah Rigidbody2D rb.

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

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

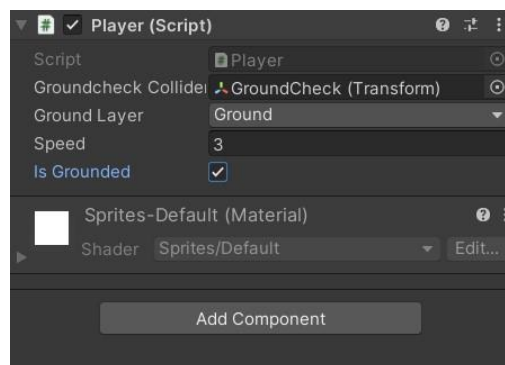
[SerializeField] bool isGrounded; // +
bool facingRight;
```

17. Buat void ground check dibawah void fixedUpdate & tambahkan GorundCheck(); 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;
}
```

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



19. Lalu untuk membuat player melompat tambahkan script berikut dibawah const float groundcheckRadius.

```
[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.

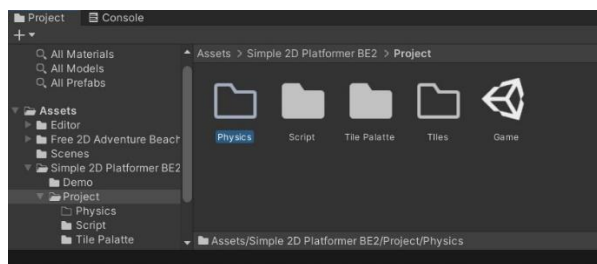
```
void FixedUpdate()
{
    GroundCheck();
    Move(horizontalValue, jump);
}
```

22. Tambahkan script berikut pada void Move.

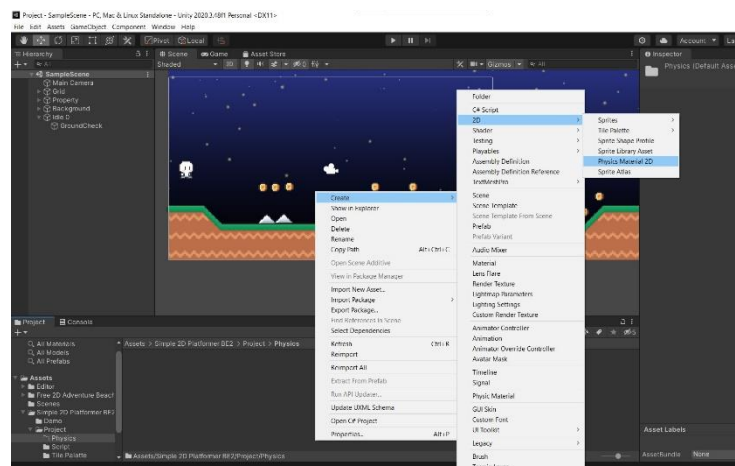
```
bool jumpflag

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

23. Buat folder baru di praktikum bernama “Physics”

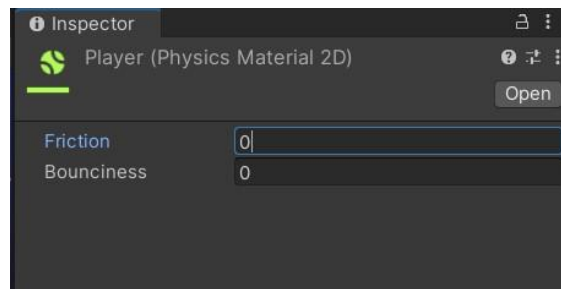


24. Didalam folder Pyshics create > 2d > physical material 2d , bernama “Player”

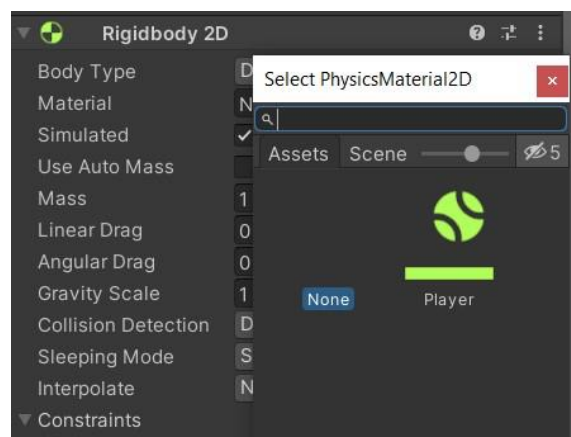




25. Klik Player (Physics Material 2D), dibagian menu inspector, friction & bounces ubah menjadi 0.

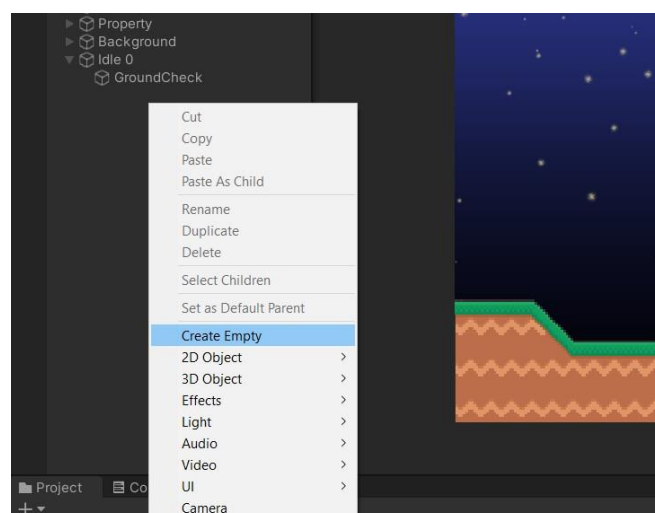


26. Klik Hierarchy pilih layer Idle 0, pada Inspector Cari Rigidbody 2D lalu klik icon untuk membuka box select physics material 2d , lalu pilih asset Player yang sudah kita buat tadi.



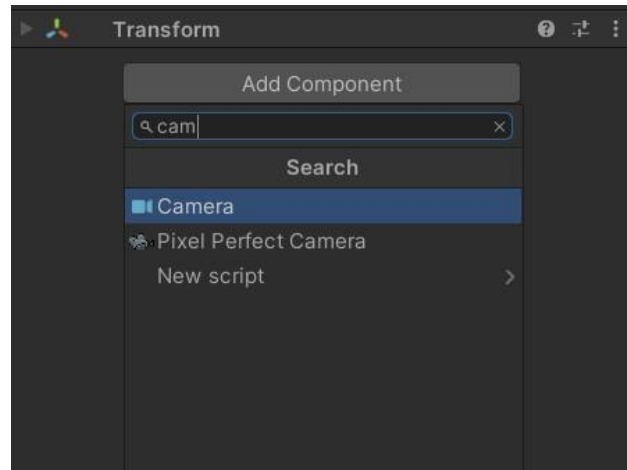
B. Membuat Pergerakan Kamera

1. Create Empty pada Hirarki, dan Rename Menjadi Camera

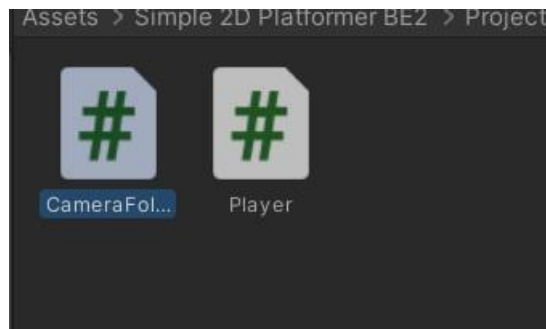




2. Pada layer Camera tambahkan komponen baru bernama Camera.



3. Lalu didalam folder Script, buat script baru bernama CameraFollow.



4. Double klik pada script CameraFollow lalu masukan codingan berikut.

```
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;
    }
}
```

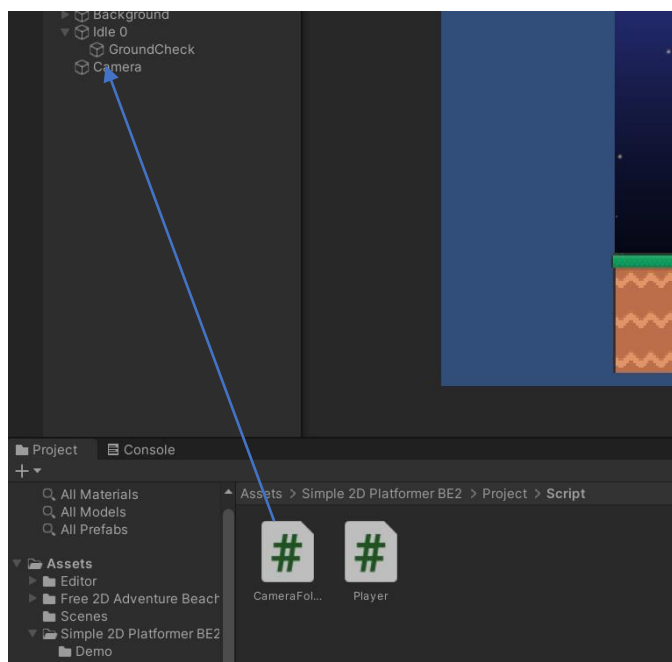


```
bool CheckYMargin()
{
    return Mathf.Abs(transform.position.y -
player.position.y) > yMargin;
}

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. Kemudian drag and drop script tadi kedalam layer Camera pada Hierarki.

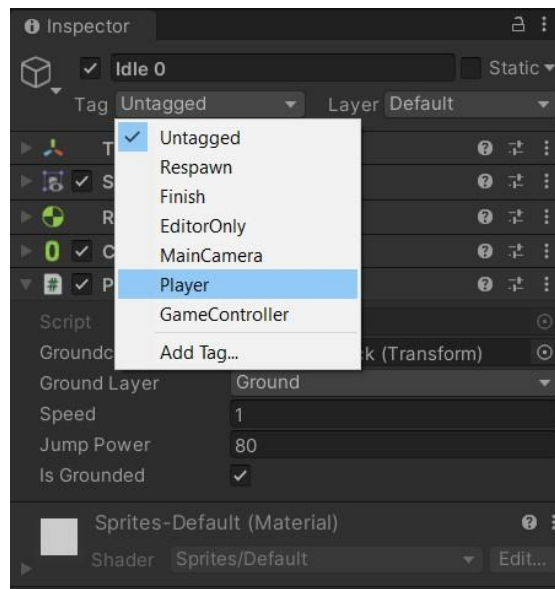




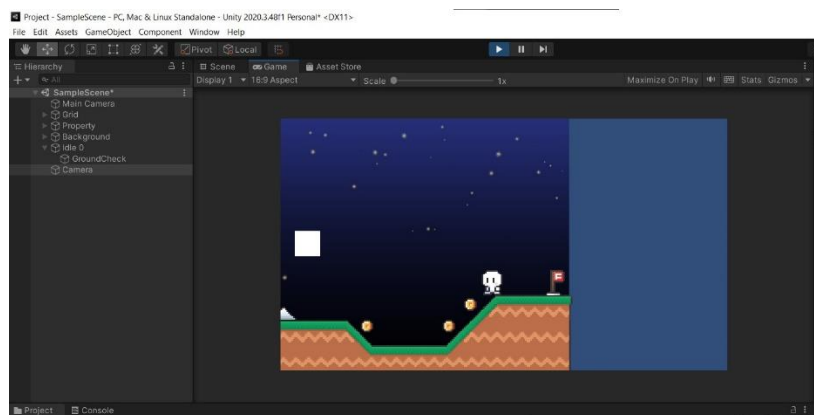
6. Lalu klik pada camera, buka inspector Pada bagian Camera Follow (Script) Ubah Bagian Max X dan Max Y nya.



7. Ubah tag di Idle 0 Untagged menjadi "Player".



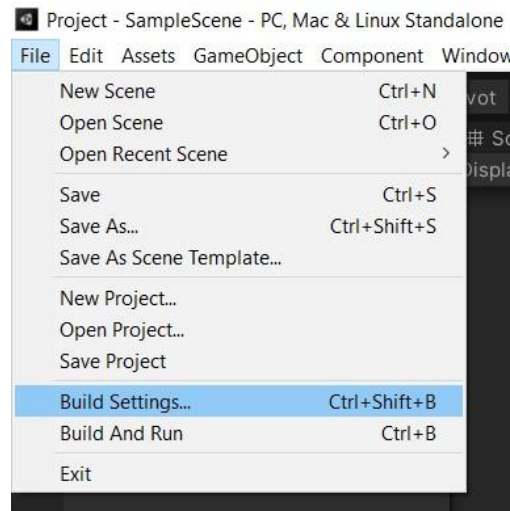
8. Tekan play untuk menjalankan, maka sekarang kamera akan mengikuti pergerakan karakter.



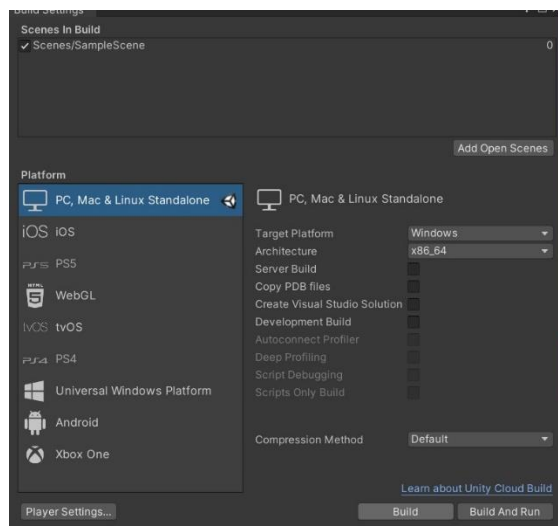


C. Rendering

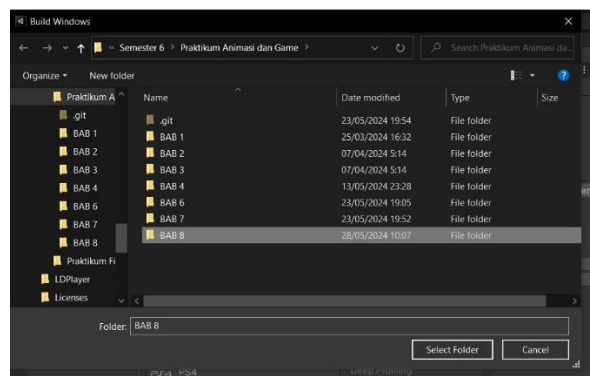
1. Pergi ke menu File kemudian pilih Build Setting (Ctrl + Shift + B).



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



3. Pilih dimana Project disimpan, dan tunggu hasilnya, maka project kita sudah berhasil dirender.





QUIZ :

Source Code :

```
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 Source Code :

Codingan diatas digunakan untuk membuat kamera mengikuti pergerakan dari pemain secara horizontal (sumbu x), SerializeField sendiri memungkinkan kita mengubah referensi dari “player” langsung dari inspector yang ada pada unity. void Update() sendiri dipanggil sekali pada setiap frame didalam unity dan posisi kamera akan diperbarui agar selalu memiliki nilai x yang sama dengan posisi pemain, sementara untuk nilai y dan z akan tetap sama.