



TUGAS PERTEMUAN: 8

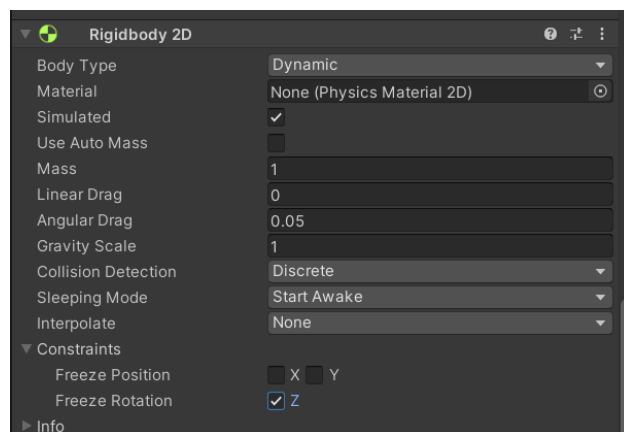
CAMERA & CHARACTER MOVEMENT

NIM	:	2118006
Nama	:	Mikhael Christian
Kelas	:	A
Asisten Lab	:	Naufal Dhiaurrafif (2218059)

8.1 Tugas 8 : Membuat Langkah-Langkah

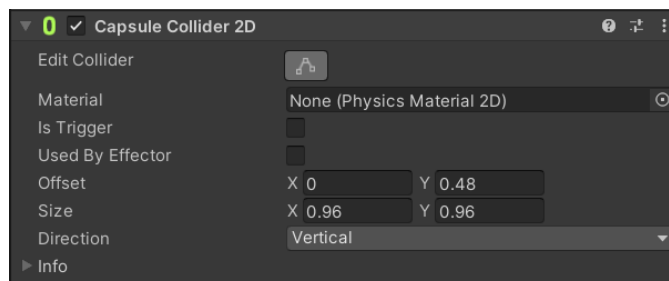
A. Langkah-Langkah Membuat Pergerakan Player

1. Buka *Project Unity* sebelumnya kemudian klik *Astronaut-idle* tambahkan *Component Rigidbody 2D*, sesuaikan settingannya seperti gambar berikut.



Gambar 8.1 Tambahkan *Player* ke *Hierarchy*

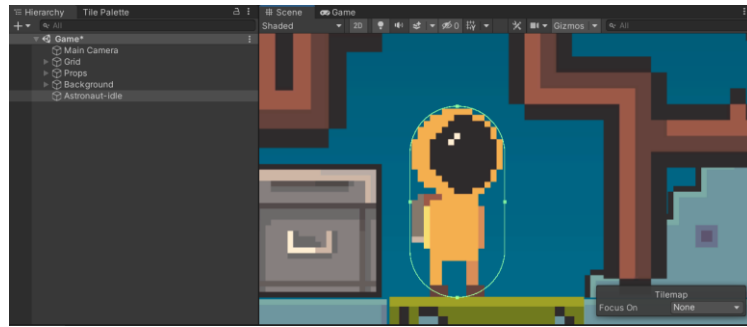
2. Kemudian tambahkan *Capsule Colider 2D* di *Astronaut-idle*, lalu klik *icon* sebelah kanan *Edit Colider*.



Gambar 8.2 Tambahkan *Capsule Collider 2D*

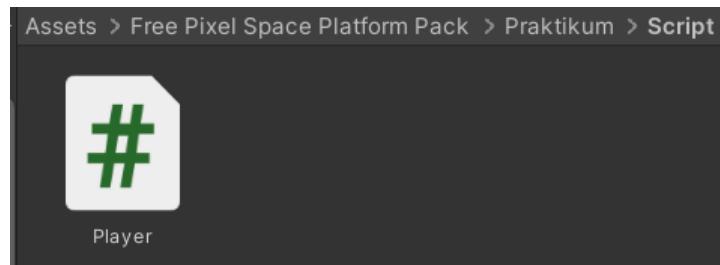


3. Lalu cocokkan garis oval dengan karakter.



Gambar 8.3 Atur Garis Oval

4. Buka folder Praktikum, lalu bikin folder baru bernama *Script*. Masuk ke dalam folder *Script*, lalu buat C# *Script*, beri nama “Player”.



Gambar 8.4 Buat C# *Script*

5. *Drag & drop script player* ke dalam *Hierarchy Astronaut-idle*, lalu klik 2x pada *script* dan masukkan *source code* di bawah ini.

```
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);
    }

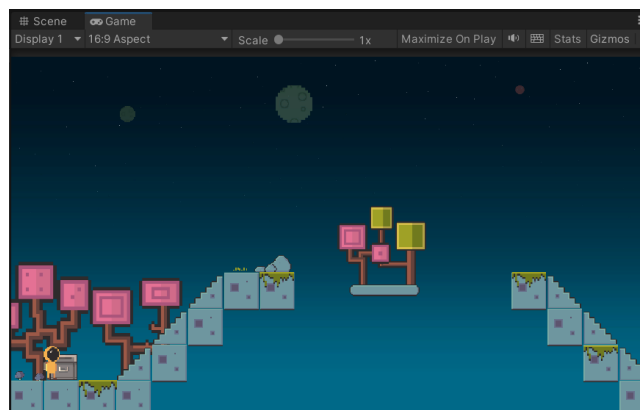
    void Move(float dir)
    {
        #region gerak kanan kiri
        float xVal = dir * speed * 100 *
Time.fixedDeltaTime;
        Vector2 targetVelocity = new Vector2(xVal,
rb.velocity.y);
        rb.velocity = targetVelocity;

        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
    }
}
```

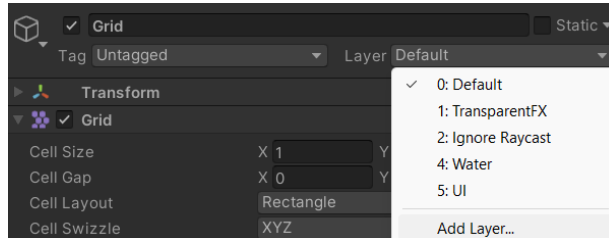
6. Untuk mencoba *Source code* di atas berhasil, tekan *keyboard* “a” atau “*left arrow*” untuk ke arah kiri, tekan “d” atau “*right arrow*” untuk ke arah kanan.



Gambar 8.6 Test Source Code

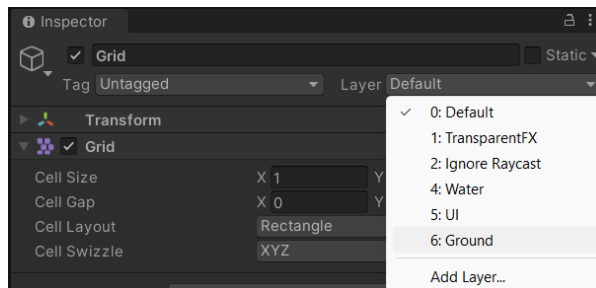


7. Untuk membuat *player* loncat menggunakan spasi, kita perlu membuat *GroundCheck* dengan cara, klik *Grid* pada *Hierarchy*, pergi ke *inspector*, pilih *Layer*, klik *Add Layer*. Kemudian isi “Ground” pada *User Layer* 6.



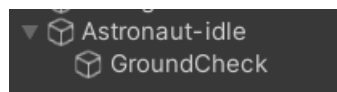
Gambar 8.6 Tambahkan *Layer* Baru

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



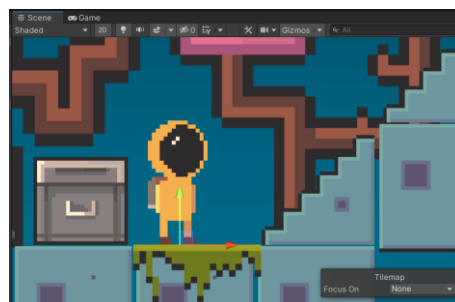
Gambar 8.7 Pilih *Layer Ground*

9. Klik kanan pada *Astronaut-idle*, lalu *Create Empty*, beri nama “GroundCheck”.



Gambar 8.8 Buat *GroundCheck*

10. Klik pada *Hierarchy GroundCheck*, lalu gunakan *Move Tools* untuk memindahkan ke bagian bawah *Player* seperti gambar berikut.



Gambar 8.9 Pindahkan *GroundCheck*



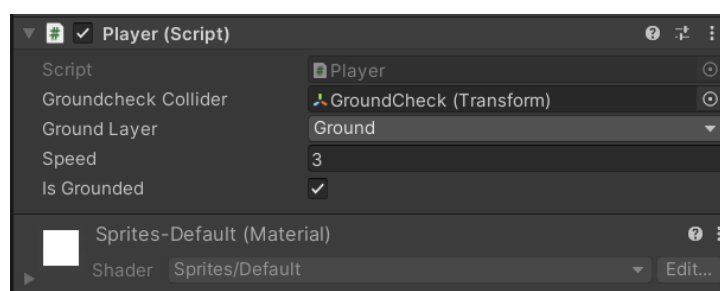
11. Kembali ke script *Player* tambahkan *source code* seperti ini.

```
[SerializeField] Transform groundcheckCollider;  
[SerializeField] LayerMask groundLayer;  
  
const float groundCheckRadius = 0.2f; // +  
[SerializeField] float speed = 1;  
float horizontalValue;  
  
[SerializeField] bool isGrounded; // +  
bool facingRight;
```

12. Buat *void ground check* di bawah *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;  
}
```

13. Klik *Astronaut-idle*, 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 8.10 Atur *Script Player* pada *Inspector*

14. Lalu untuk membuat *player* melompat tambahkan *script* berikut.

```
[SerializeField] float jumpPower = 100;  
  
bool jump;
```



15. Tambahkan juga *script* berikut di bagian *void update*.

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

16. Tambahkan juga *jump* pada parameter *Move*.

```
36 void FixedUpdate()
37 {
38     GroundCheck();
39     Move(horizontalValue, jump);
40 }
```

Gambar 8.11 Tambahkan *Script*

17. Tambahkan *script* berikut pada *void Move*.

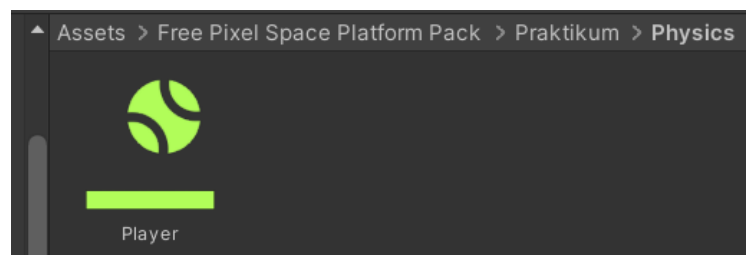
```
void Move(float dir, bool jumpflag)
{
    if(isGrounded && jumpflag)
    {
        isGrounded = false;
        jumpflag = false;
        rb.AddForce(new Vector2(0f, jumpPower));
    }
}
```

Gambar 8.12 Tambahkan *Script*

```
bool jumpflag

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

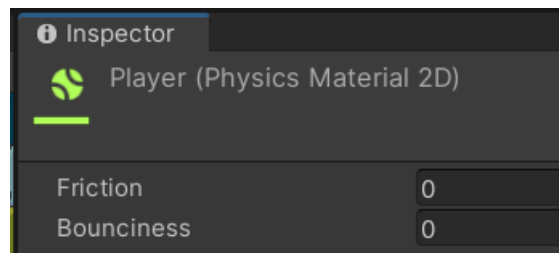
18. Buat folder baru di praktikum bernama “Physics”, kemudian di dalam folder “Pyshics” *Create* > 2D > *Physical Material* 2D, bernama “Player”.



Gambar 8.13 Tambahkan *Physical Material* 2D

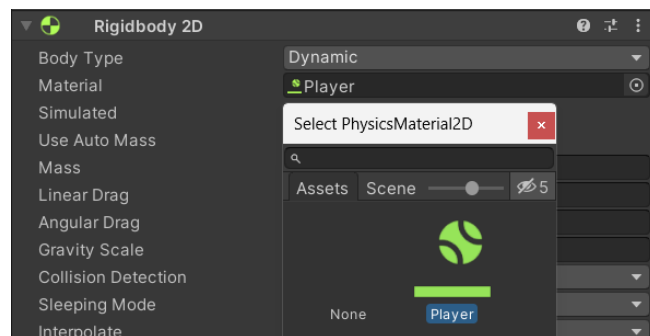


19. Klik *Player (Physics Material 2D)*, di bagian menu *Inspector*, *Friction* & *Bounciness* ubah menjadi 0.



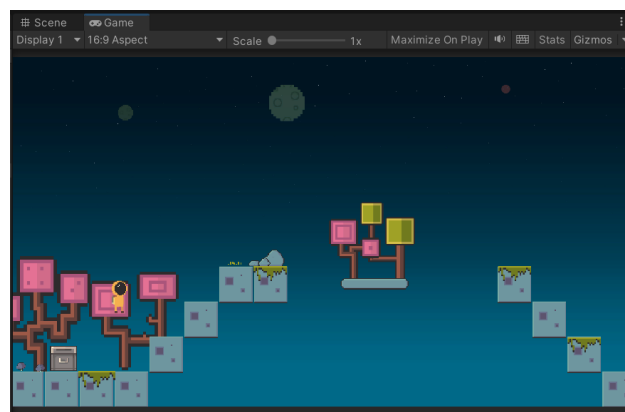
Gambar 8.14 Atur *Physics*

20. Klik *Hierarchy* pilih *layer* *Astronaut-idle*, pada *Inspector* Cari *Rigidbody 2D* lalu klik *icon* untuk membuka *box Select Physics Material 2D*, lalu pilih *asset Player* yang sudah dibuat.



Gambar 8.15 Pilih *Physics Player*

21. Tekan *play*, maka *player* bisa melompat dengan menekan spasi.

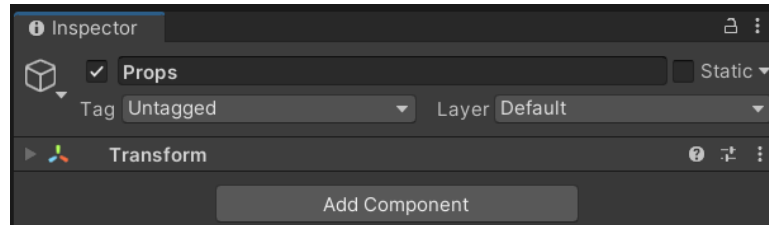


Gambar 8.16 Tekan *Play*



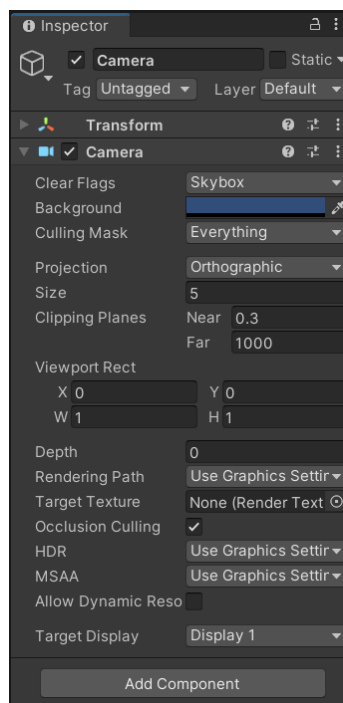
B. Langkah-Langkah Membuat Camera Movement

1. Pada *Hierarchy Props* ubah *Inspector* pada *Tag Main camera* menjadi *untagged*.



Gambar 8.17 Atur *Inspector Props*

2. *Create Empty* pada *Hierarchy*, dan *Rename* menjadi “Camera”, kemudian sesuaikan *Setting Layer Camera* seperti di bawah ini.



Gambar 8.18 Atur *Inspector Camera*

3. Buat file *script* baru di folder *Script* dengan nama “CameraFollow”, lalu tuliskan *script* 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;
}

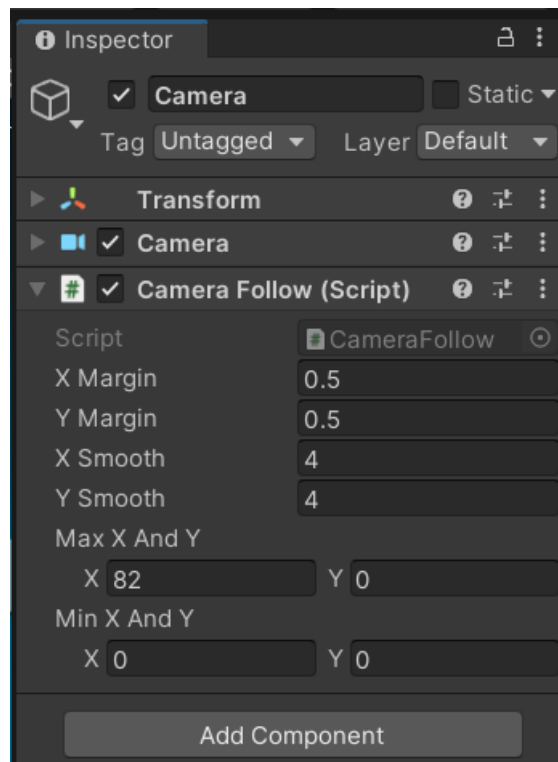
bool CheckYMargin()
{
    return Mathf.Abs(transform.position.y -
player.position.y) > yMargin;
}

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);
}
}
```

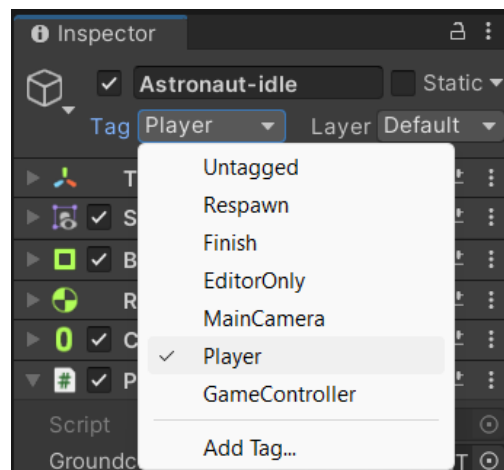


4. *Drag & drop script CameraFollow ke dalam Layer Camera. Lalu klik pada camera, buka inspector setting menjadi berikut.*



Gambar 8.19 Atur *Inspector Camera*

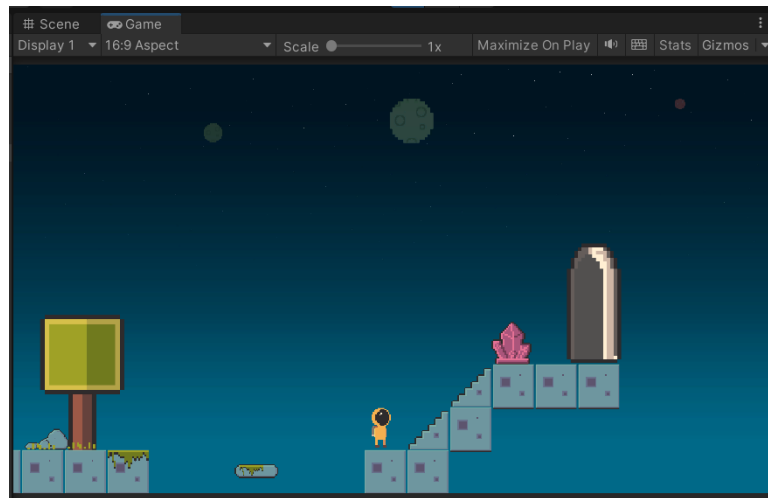
5. *Ubah Tag di Astronaut-idle Untagged menjadi "Player"*



Gambar 8.20 Atur *Tag Astronaut-idle*



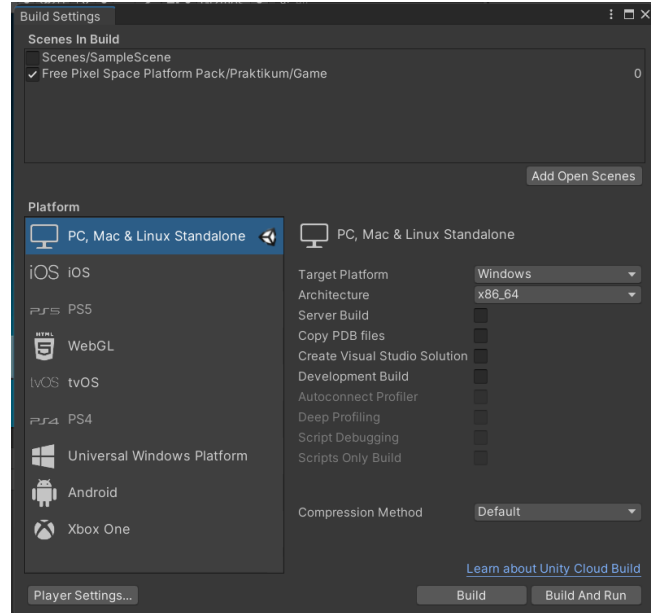
6. Tekan *play* untuk menjalankan, maka sekarang kamera akan mengikuti pergerakan karakter.



Gambar 8.21 Tekan *Play*

C. Langkah-Langkah Render

1. Pergi ke menu *File* kemudian pilih *Build Setting* (*Ctrl + Shift + B*). Pada *Setting Build* ini pilih *PC, Mac & Linux*, tekan *Build*, pastikan pada menu *Scene in Build* berada pada *project* yang sedang dikerjakan.



Gambar 8.22 Atur *Build Settings*



2. Kemudian pilih dimana *Project* disimpan dan tunggu.

Name	Date modified	Type	Size
MonoBleedingEdge	30/05/2024 0:17	File folder	
Tugas_Data	30/05/2024 0:17	File folder	
Tugas	09/05/2023 8:25	Application	639 KB
UnityCrashHandler64	09/05/2023 8:32	Application	1.204 KB
UnityPlayer.dll	09/05/2023 8:32	Application extens...	27.871 KB

Gambar 8.23 Tampilan Folder *Game*

D. 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);
    }
}
```

Penjelasan:

Source code di atas merupakan script agar kamera dapat mengikuti gerakan player. Kamera tersebut akan mengikuti gerakan player pada sumbu x. Namun terdapat beberapa kesalahan penulisan, seperti pada penulisan “Void” seharusnya menjadi “void” dan seharusnya tidak ada spasi atau jarak pada parameter Vector3. Seharusnya “player.position.x”.