



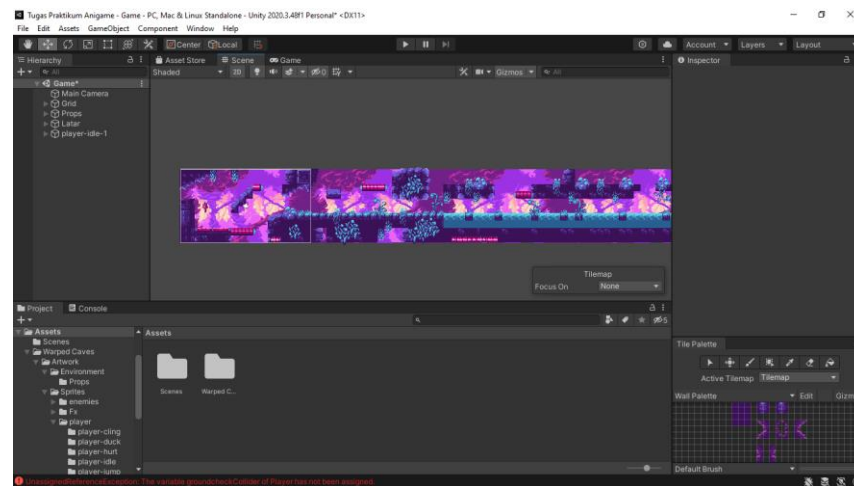
TUGAS PERTEMUAN: 8

CAMERA & CHARACTER MOVEMENT

NIM	:	2118030
Nama	:	Putra Prasetya Utama
Kelas	:	D
Asisten Lab	:	Aprillia Dwi Dyah S. (2118143)

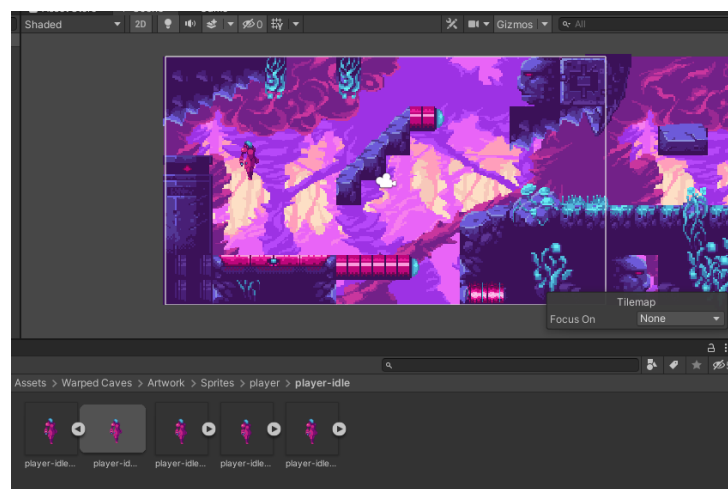
8.1 Tugas 1 : Langkah-langkah Membuat Pergerakan Karakter

1. Buka file unity sebelumnya yang sudah dibuatnya *platform tile*.



Gambar 8.1 Membuka File Unity Sebelumnya

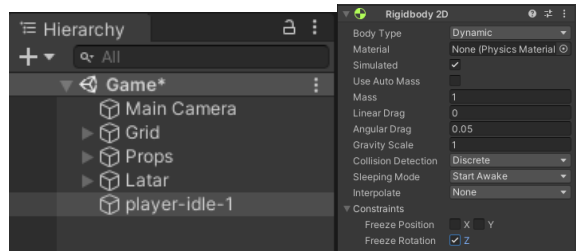
2. Lalu hapus *player* yang sudah dibuat dan tambahkan lagi *player idle* nya dengan cara *drag & drop*.



Gambar 8.2 Menambahkan Player Idle Baru

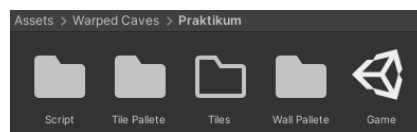


3. Lalu klik pada player yang ditambahkan lalu tambahkan komponen Rigidbody 2D dan centang *Freeze Rotation Z*.



Gambar 8.3 Membuat Komponen Rigidbody 2D Pada Player

4. Lalu buat folder Script yang terletak didalam folder Praktikum.



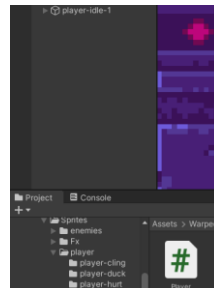
Gambar 8.4 Membuat Folder Script

5. Didalam folder script buat C# Script dan beri nama Player.



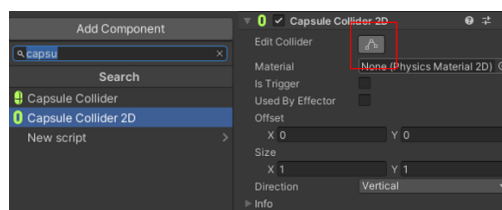
Gambar 8.5 Membuat C# Script Baru Dengan Nama Player

6. Lalu *drag & drop* file C# Script pada player yang ada di hirarki.



Gambar 8.6 Drag & Drop Script Kedalam Player Yang Ada Di Hirarki

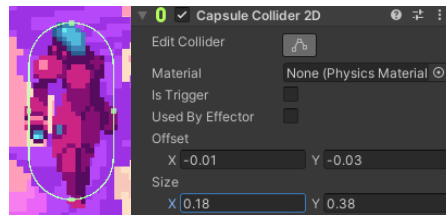
7. Tambahkan komponen *Capsule Colider* 2D lalu klik icon disebelah kanan *Edit Collider*.



Gambar 8.7 Menambahkan Capsule Colider 2D Pada Player

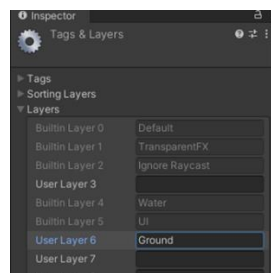


8. Lalu cockan garis oval degan karakternya atau bisa di inputkan Offset X, Y dan juga Size X, Y nya.



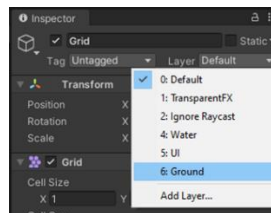
Gambar 8.8 Mengatur Offset dan Size

9. Klik *Grid* pada *Hierarchy*, pergi ke *inspector*, pilih *Layer*, Klik *Add Layer*. Lalu isi “*Ground*” pada *User Layer 6*.



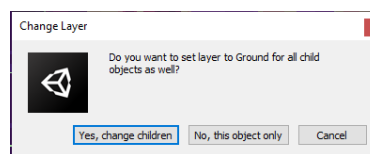
Gambar 8.9 Menambahkan Layer Ground Pada Layer 6

10. Setelah itu ubah *layer* menjadi *ground*



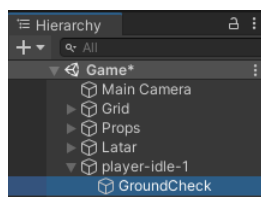
Gambar 8.10 Menambah Layer Baru Dan Mengubah Layer Pada Grid

11. Klik *yes* jika terdapat *pop up Change Layer*.



Gambar 8.11 Klik Yes Pada Saat Pop Up Change Layer

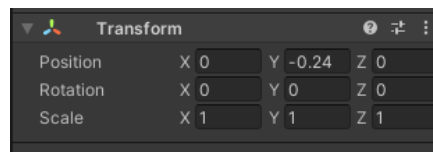
12. Klik kanan pada *Player*, lalu *Create empty*, beri nama *GroundCheck*.



Gambar 8.12 Membuat GroundCheck

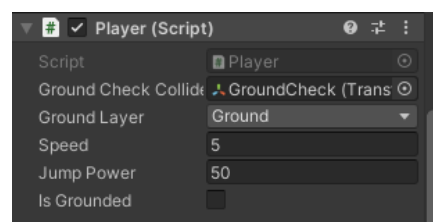


13. Klik pada Hirarki *GorundCheck*. Lalu ubah *transform* seperti gambar dibawah.



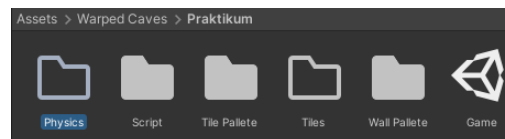
Gambar 8.13 Mengubah Posisi Sumbu Y Pada GroundCheck

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



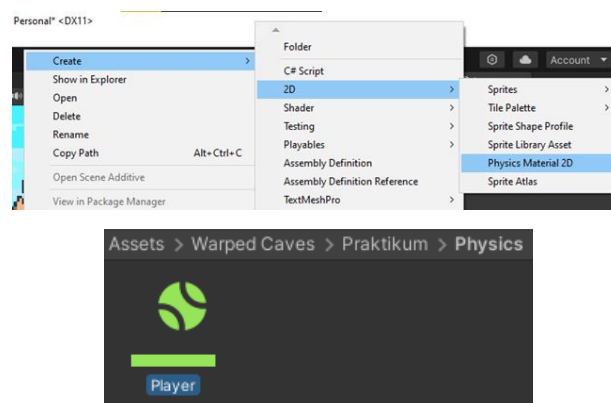
Gambar 8.14 Memilih GroundCheck Pada GroundCheck Collider

15. Buat folder baru di folder *Praktikum* bernama *Physics*.



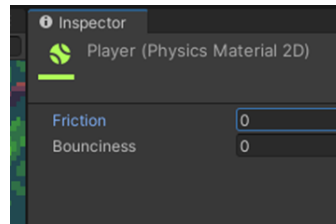
Gambar 8.15 Membuat Folder Physics

16. Lalu masuk kedalam folder *Physics* yang sudah dibuat dan buat *Physics Material 2D* dan beri nama *Player*.



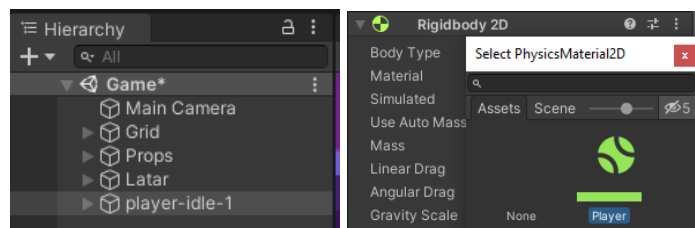
Gambar 8.16 Membuat Physics Material 2D Untuk Player

17. Lalu klik pada *physics* yang sudah dibuat, pergi ke *inspector* lalu ubah *friction* menjadi 0.



Gambar 8.17 Mengubah Friction Menjadi 9

18. 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 8.18 Menambahkan PhysicMaterial2D Pada Player

19. Tambahkan *source code* dibawah ini agar *player* dapat bergerak menggunakan *keyboard* A dan D atau arah panah kiri, kanan. Juga agar *player* dapat melompat menggunakan spasi *keyboard*.

```
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; // Radius untuk
    ground check
    [SerializeField] float speed = 2; // Kecepatan gerak
    [SerializeField] float jumpPower = 2; // Kekuatan
    lompat

    bool jump;
    float horizontalValue;

    [SerializeField] bool isGrounded; // Status apakah di
    tanah atau tidak
    bool facingRight = true; // Mengatur arah menghadap
    player, default menghadap kanan

    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;
    }
}

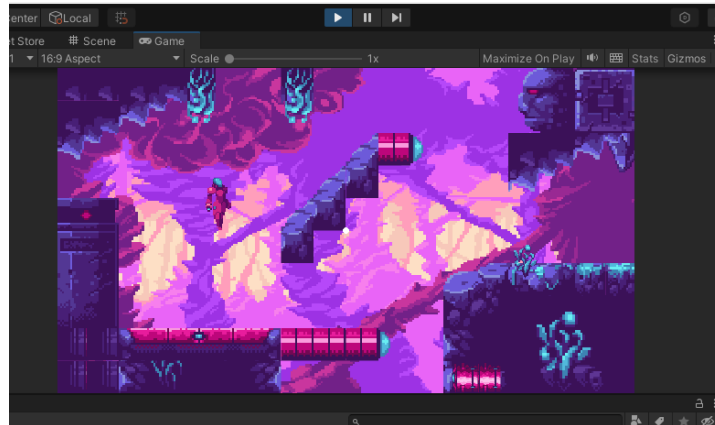
void Move(float dir, bool jumpFlag)
{
    if (isGrounded && jumpFlag)
    {
        isGrounded = false;
        rb.AddForce(new Vector2(0f, jumpPower),
ForceMode2D.Impulse); // Menggunakan ForceMode2D.Impulse
untuk lompat yang lebih realistis
    }

    // Gerakan ke kanan dan kiri
    float xVal = dir * speed;
    Vector2 targetVelocity = new Vector2(xVal,
rb.velocity.y);
    rb.velocity = targetVelocity;

    // Mengubah arah menghadap player
    if (facingRight && dir < 0){
        transform.localScale = new Vector3(-4, 4, 4);
        facingRight = false;
    }else if (!facingRight && dir > 0){
        transform.localScale = new Vector3(4, 4, 4);
        facingRight = true;
    }
}
}
```



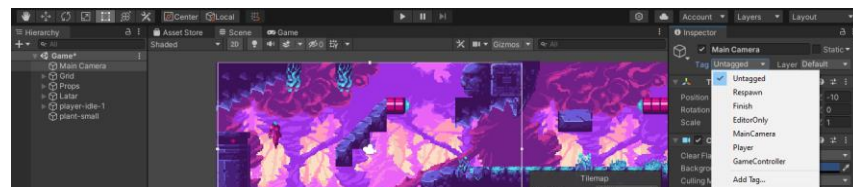
20. Klik play dan coba jalankan agar *Player* dapat bergerak dan melompat.



Gambar 8.19 Mencoba Player Bergerak dan Melompat

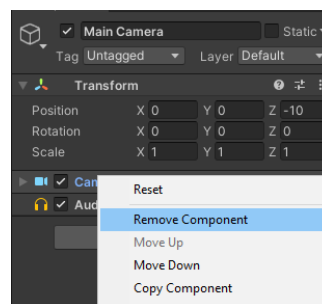
8.2 Tugas 2 : Langkah-langkah Membuat Camera Movement

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



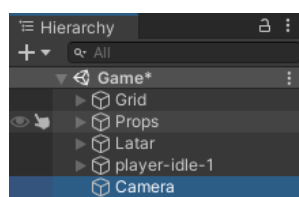
Gambar 8.20 Mengubah Tag Main Camera

2. Pada Effect Camera pilih Remove Component. Lalu hapus juga *main camera* pada hirarki.



Gambar 8.21 Remove Component Camera

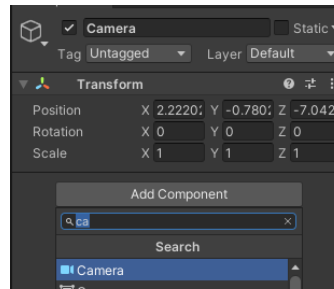
3. *Create Empty* pada Hirarki, dan *Rename* Menjadi *Camera*.



Gambar 8.22 Membuat Camera Baru



4. Tambahkan komponen camera.



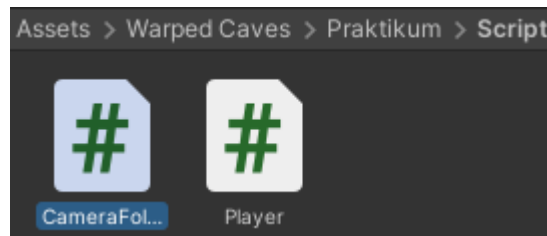
Gambar 8.23 Menambahkan Komponen Kamera

5. Sesuaikan *Camera* seperti gambar dibawah ini.



Gambar 8.24 Menyesuaikan Camera

6. Buat file *script* baru di folder *Script* dengan nama *CameraFollow*



Gambar 8.25 Membuat C# Script Baru

7. Klik 2 kali pada *CameraFollow* dan tambahkan *source code* dibawah ini.

Setelah itu *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;
}

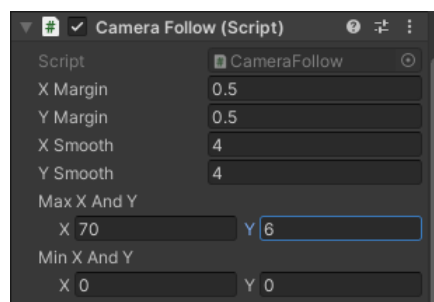
bool CheckXMargin() {
    return Mathf.Abs(transform.position.x
player.position.x) > xMargin;
}

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);
}
```

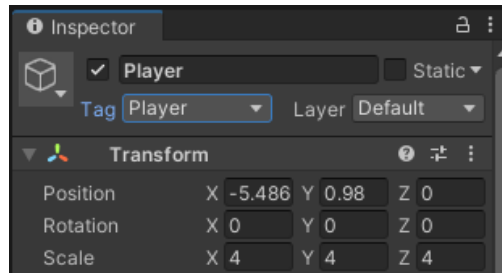
8. Lalu klik pada camera, buka inspector Pada bagian Camera Follow (Script)
Ubah Bagian Max X menjadi 70 dan Max Y menjadi 6.



Gambar 8.26 Mengubah Max X dan Y

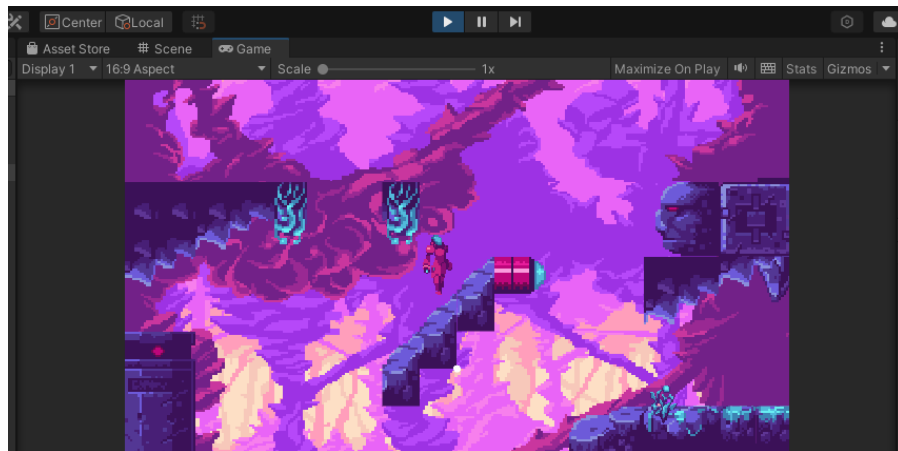


9. Ubah tag di *Player Untagged* menjadi "*Player*"



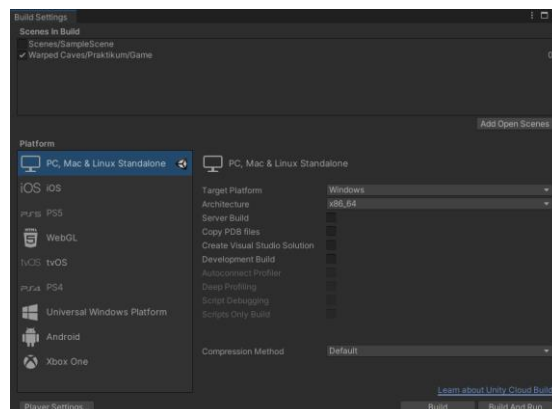
Gambar 8.27 Mengubah Tag Player

10. Jalankan maka kamera akan mengikuti pergerakan *Player*.



Gambar 8.28 Menjalankan Game

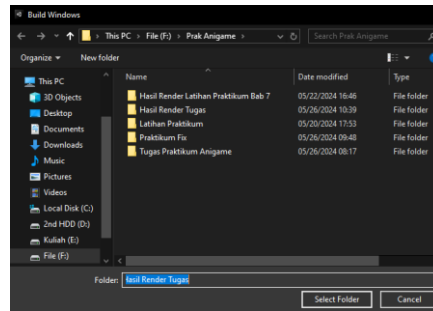
11. Pergi ke *menu File* kemudian pilih *Build Setting* (Ctrl + Shift + B). Lalu Pada Setting *Build* ini pilih PC, Mac & Linux, Tekan *Build*, pastikan pada menu *Scene in Build* berada pada *project* yang sudah dibuat sebelumnya. Lalu klik *Build and Run*.



Gambar 8.29 Mengatur Build Setting

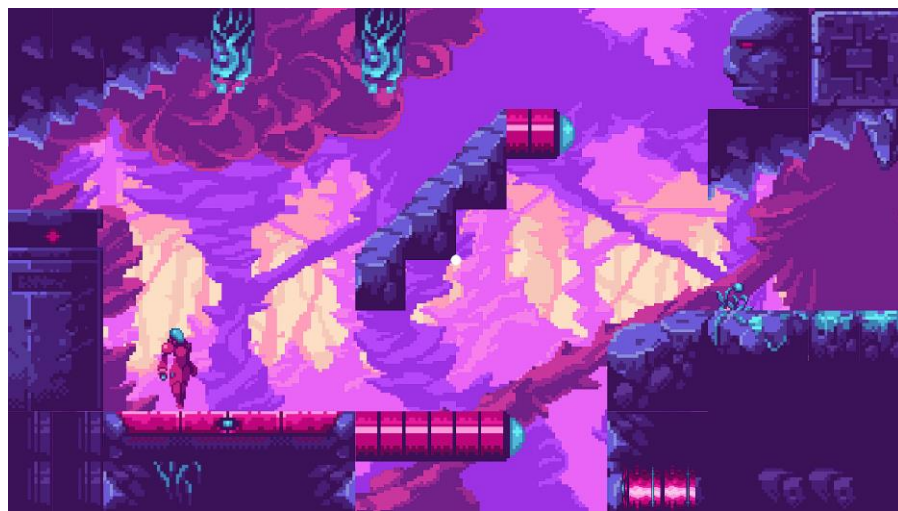


12. Tempatkan hasil render sesuai dengan keinginan.



Gambar 8.30 Menempatkan Hasil Render

13. Hasil ketika sudah di render.



Gambar 8.31 Hasil Setelah Render

8.3 Kuis CameraFollow

```
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:

Pertama dilakukannya impor *namespace* dari sistem untuk menggunakan C# dan *UnityEngine* untuk fitur dari unity agar dapat digunakan. Lalu dibuat *class* tipe *public* dengan nama *CameraFollow* dengan turunan dari kelas ini yaitu *MonoBehaviour*. Lalu dibuat variabel *player* dengan tipenya yaitu *transform*



yang variabel ini bersifat privat dengan *SerializeField* agar *player* dapat diatur dari *inspector* meskipun bersifat privat. Lalu terakhir terdapat metode *update* yang didalamnya digunakan untuk mengatur posisi kamera akan selalu mengikuti sumbu x dan sedangkan untuk sumbu y dan z kamera akan tetap pada posisinya.