TUGAS PERTEMUAN: 8

CAMERA & CHARACTER MOVEMENT

NIM	:	2118023
Nama	:	Ckristina Candra Dewi
Kelas	:	D
Asisten Lab	:	Natasya Octavia (2118034)

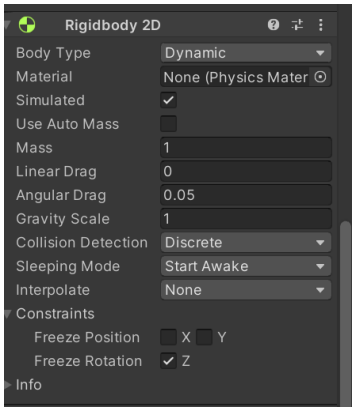
8.1 Tugas 1 : Membuat Character Movement, Detect Ground, Jumping

A. Membuat Pergerakan Player

1. Buka project unity pada bab 7

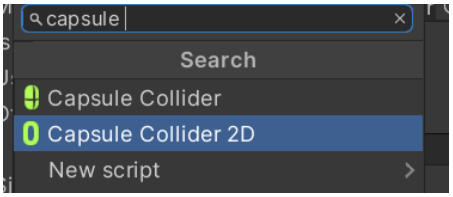
Gambar 8.1 Tampilan Project Bab 7

2. Pilih player-idle 1, lalu pada Rigidbody 2D gentang bagian freeze Rotation Z.



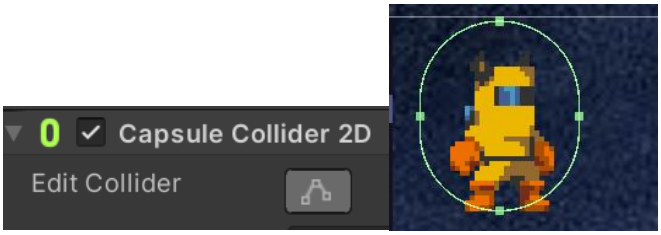
Gambar 8.2 Tampilan Rigidbody 2D

3. Tambahkan komponen Capsule Collider 2D pada player-idle 1.



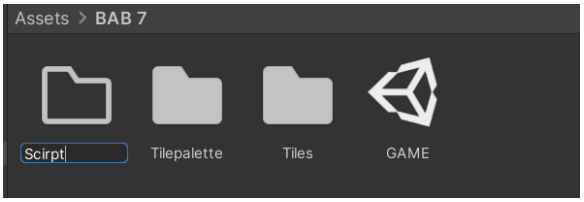
Gambar 8.3 Tampilan Komponen Baru

4. Pada Capsule Collider 2D pilih edit collider, lalu sesuaikan garis oval dengan ukuran player-idle 1



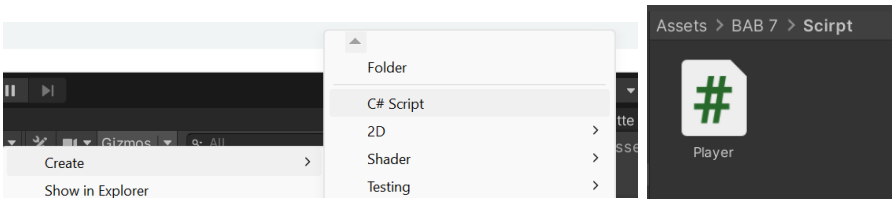
Gambar 8.4 Tampilan Capsule Collider 2D

5. Buka folder bab 7 lalu, tambahkan folder baru dengan nama script.



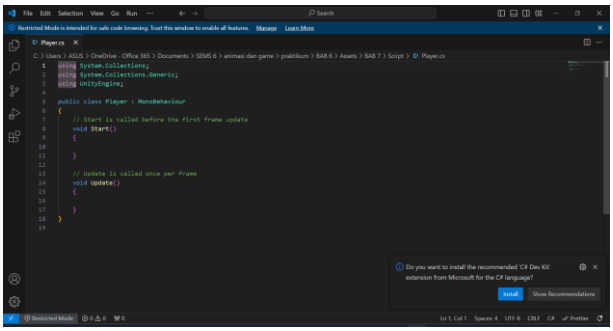
Gambar 8.5 Tampilan Folder Script

6. Masuk ke dalam folder script, lalu klik kanan pilih create dan pilih C# Script. Beri nama player.



Gambar 8.6 Tampilan Membuat Script

7. Drag and drop script player ke player-idle 1, lalu klik 2 kali pada script player untuk masuk ke text editor.



Gambar 8.7 Tampilan Text Editor Player

8. Tambahkan source code berikut pada Player.cs

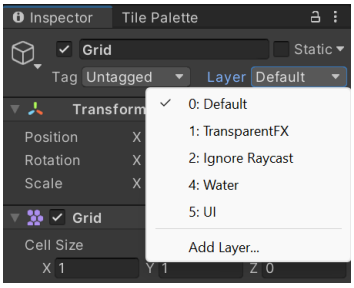
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

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
    }
}
```

9. Jalankan game untuk mengetahui apakah source code berjalan dengan baik atau tidak. Tekan A atau left arrow untuk kea rah kiri dan D atau right arrow untuk kea rah kanan.

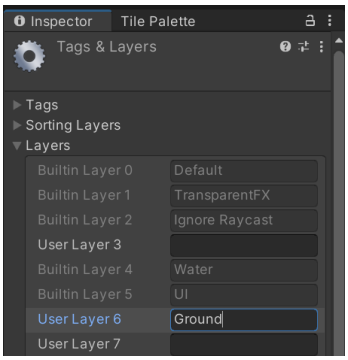
Gambar 8.8 Tampilan Hasil

10. Untuk membuat player bisa loncat menggunakan spasi, perlu membuat GorundCheck dengan cara klik grid pada hierarch. Pada inspector pilih layer lalu add layer.



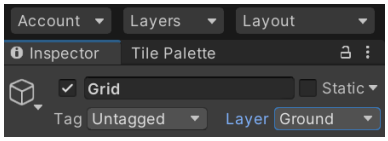
Gambar 8.9 Tampilan Add Layer

11. Pilih user layer 6 lalu isi dengan Ground.



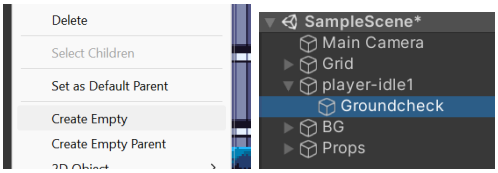
Gambar 8.10 Tampilan User Layer

12. Ubah layer menjadi Ground, jika muncul pop up pilih yes.



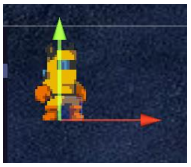
Gambar 8.11 Tampilan Layer

13. Pilih player-idle 1 pada hierarchy, klik kanan lalu pilih Create Empty dan beri nama Groundcheck.



Gambar 8.12 Tampilan Groundcheck

14. Pada hirarki groundcheck, ubah ke move tool lalu geser panah ke bawah.



Gambar 8.13 Tampilan Tampilan Groundcheck

15. Lalu tambahkan source code berikut

```
public class Player : MonoBehaviour
{
  Rigidbody2D rb;

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

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

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

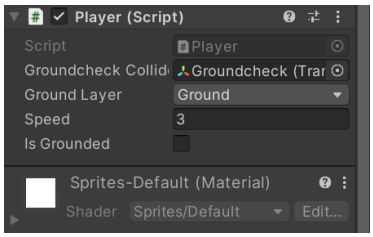
Gambar 8.14 Tampilan Source Code

16. Buat void ground check lalu lalu tambahkan source code berikut

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

void GroundCheck()
{
isGrounded = false;
Collider2D[] colliders = Physics2D.OverlapCircleAll(groundcheckCollider.position, groundCheckRadius, groundLayer);
if (colliders.Length > 0)
isGrounded = true;
}
```

Gambar 8.15 Tampilan Source Code

17. pilih player-idle 1, pada inspector ubah groundcheck, layer dan speed seperti pada gambar.



Gambar 8.16 Tampilan Groundcheck

18. Tambahkan float jumpower dan bool jump.

```
const float groundCheckRadius = 0.2f; // +
[SerializeField] float speed = 1;
[SerializeField] float jumpPower = 100;
float horizontalValue;

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

Gambar 8.17 Tampilan Source Code

19. Tambahkan soid source code if pada void update

```
void Update ()
{
  horizontalValue = Input.GetAxisRaw("Horizontal");
  if (Input.GetButtonDown("Jump"))
  jump = true;
  else if (Input.GetButtonUp("Jump"))
  jump = false;
}
```

Gambar 8.18 Tampilan Source Code

20. Tambahkan parameter jump pada move di void FixedUpdate

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

Gambar 8.19 Tampilan Source Code

21. Tambahkan source code berikut pada void move

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

Gambar 8.20 Tampilan Source Code

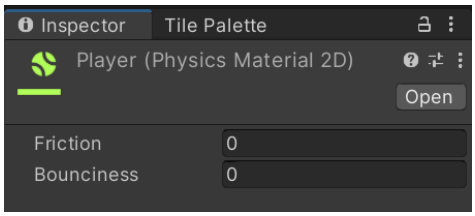
22. Buat folder baru dengan nama Physic

Gambar 8.21 Tampilan Folder Physic

23. Buka folder Physic, lalu klik kanan create > 2D > Physic Material 2D. Ubah namanya menjadi Player.

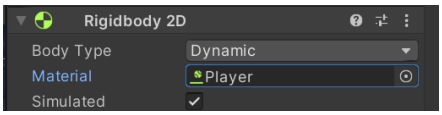
Gambar 8.22 Tampilan Physiv Material 2D

24. Klik player yang baru dibuat, pada inspector ubah friction dan bounciness menjadi 0.



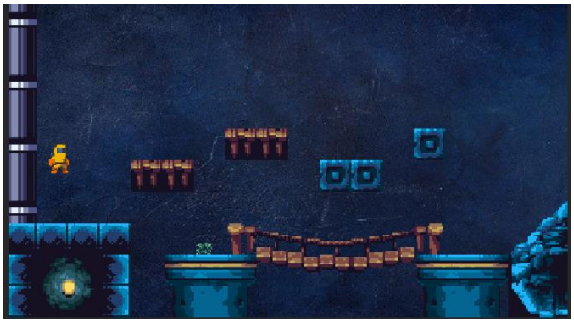
Gambar 8.23 Tampilan Player (Physics Material 2D)

25. Pada player-idle 1, Pada Rigidbody 2D ubah material menjadi Player



Gambar 8.24 Tampilan Rigidbody 2D

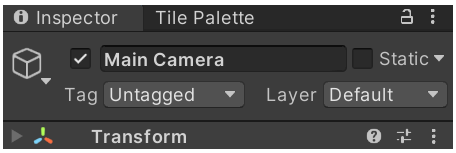
26. Jalankan game, untuk memastikan apakah game berjalan dengan baik atau tidak.



Gambar 8.25 Tampilan Hasil

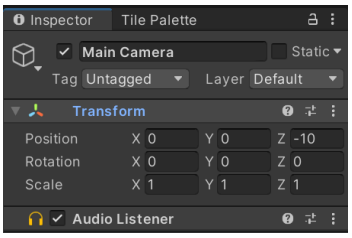
B. Camera Movement

1. Pada bagian Main Camera ubah tag menjadi Untagged.



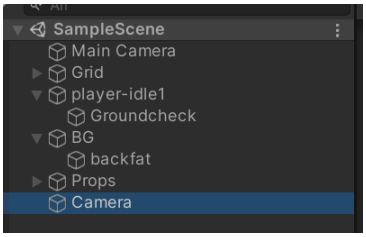
Gambar 8.26 Tampilan Props

2. Lalu remove component camera, dengan cara klik kanan lalu pilih remove component



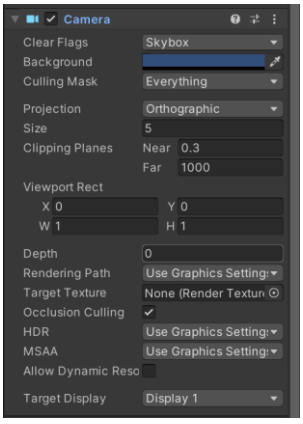
Gambar 8.27 Tampilan Remove Component Camera

3. Pada hierarchy klik kanan lalu create empty dan ubah namanya menjadi camera.



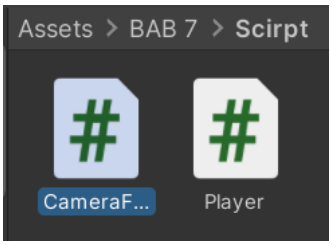
Gambar 8.28 Tampilan Membuat Camera

4. Klik pada camera yang baru dibuat lalu, pada component camera sesuaikan dengan gambar berikut.



Gambar 8.29 Tampilan Component Camera

5. Buka folder BAB 7 ke script lalu buat script baru dengan nama CameraFollow.



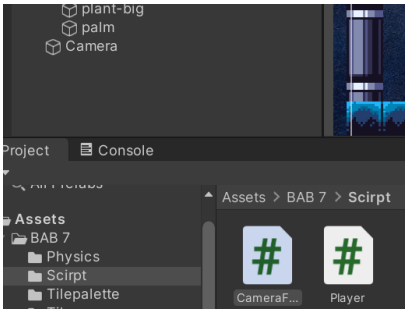
Gambar 8.30 Tampilan Folder Baru

6. Tambahkan source code 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,
```

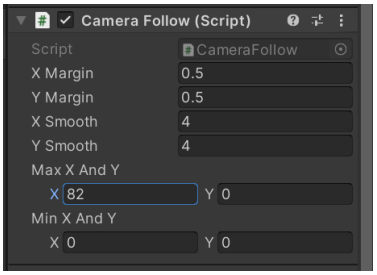
```
            xSmooth * Time.deltaTime);
        if (CheckYMargin())
            targetY = Mathf.Lerp(transform.position.y,
player.position.y,
            ySmooth * Time.deltaTime);
            targetX = Mathf.Clamp(targetX, minXAndY.x,
maxXAndY.x); targetY =
            Mathf.Clamp(targetY,            minXAndY.y,
maxXAndY.y); transform.position = new
            Vector3(targetX,                   targetY,
transform.position.z);
    }
}
```

7. Drag and drop script CameraFollow ke Camera.



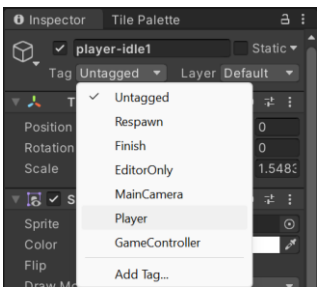
Gambar 8.31 Tampilan Drag and Drop Camera

8. Pada inspector camera, ubah nilai max X menjadi 82.



Gambar 8.32 Tampilan Camera

9. Pilih player-idle 1 pada hierarchy, kemudian ke inspector dan ubah tag menjadi player.



Gambar 8.33 Tampilan Player-Idle 1

10. Jalankan game agar mengetahui apakah camera bisa berfungsi dengan baik atau tidak.

Gambar 8.34 Tampilan Camera

C. KUIS CameraFollow

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class CameraFollow: MonoBehaviour
{
  [SerializeField] private Transform player;
  void Update () {
    transform.position       =       Vector3       (player.
Position.x,transform.position.y, transform.position.z);
  }
}
```

Penjelasan :

Source code diatas merupakan script unity yang digunakan untuk mengatur camera dalam game agar mengikuti arah gerakan player. Pada baris pertama hingga ketiga digunakan untuk import namespace yang akan digunakan dalam script camerafollow. Lalu, pada baris keempat terdapat deklarsi CameraFollow yang mewarisi MonoBehavior, kelas tersebut merupakan kelas dasar untuk semua script unity. Pada baris kelima terdapat variabel privat bertipe Transform dengan nama player. Variable tersebut ditandai dengan atribut SerializeField, yang mana nilainya dapat diedit di Unity Inspector. Variabel ini akan digunakan untuk menyimpan referensi ke transform objek pemain. Void update dipanggil setiap frame oleh Unity dan digunakan untuk memperbarui status objek. Baris ketujuh digunakan untuk mengatur posisi camera menjadi sama dengan posisi pemain. Kamera akan selalu mengikuti pemain disepanjang sumbu x , tetapi ketinggian dan kedalaman kamera tidak berubah.

D. Link Github Pengumpulan

https://github.com/kristinacandra/2118023_PRAK_ANIGAME.git