TUGAS PERTEMUAN: 8

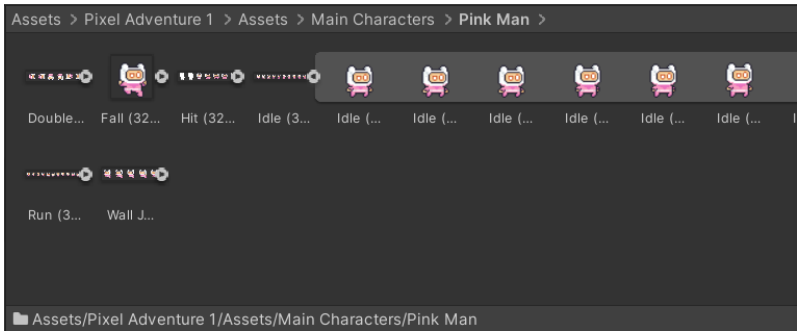
Camera & Character Movement

NIM	:	2018116
Nama	:	Ahmad Fatah Nurul Yaqin
Kelas	:	D
Asisten Lab	:	M. RAFI FADDILANI (2118144)

1.1 Tugas 1 : Menerapkan Camera & Character Movement

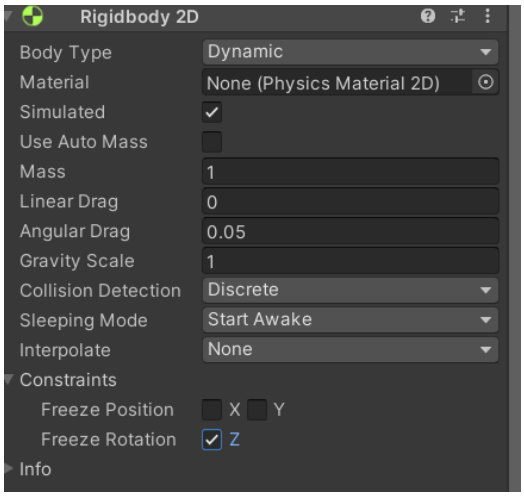
A. Menerapkan Camera & Character Movement

1. Pertama tambahkan player dan pilih pose idle import kedalam hierarki



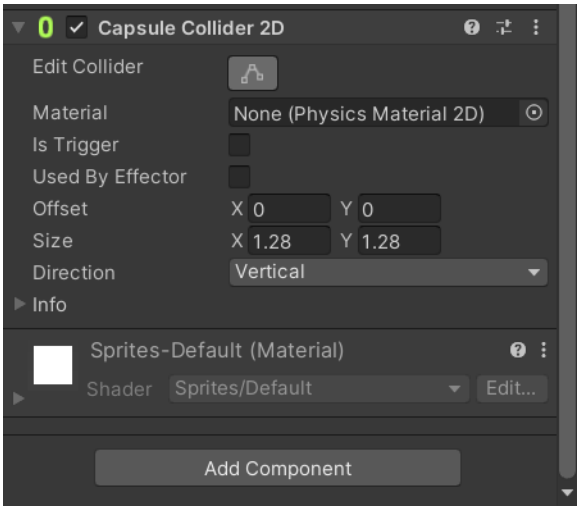
Gambar 1.1 Player idle

2. Tambahkan rigid body 2d kedalam player dengan memilih bagian inspector



Gambar 1.2 Form Register

3. Tambahkan juga collider 2D kedalam player

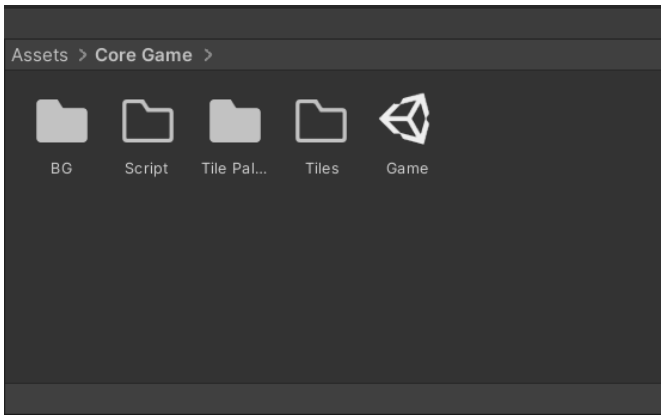


Gambar 1.3 Email

4. Atur/Edit collider sesuai dengan bentuk player agar tidak terlalu banyak ruang kosong saat bersentuhan dengan tiles

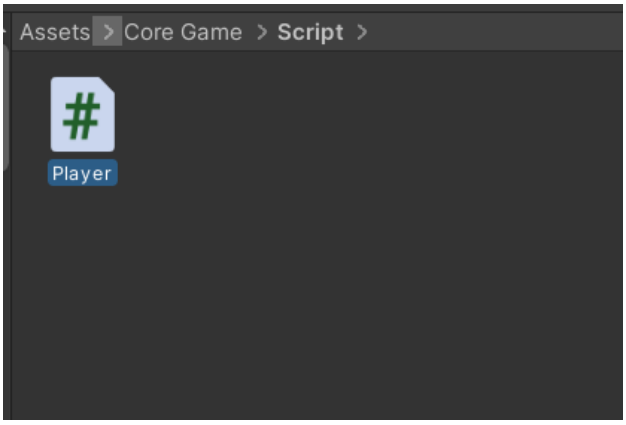
Gambar 1.4 Edit Collider

5. Buat folder baru bernama script



Gambar 1.5 Folder Script

6. Buat file C# dengan nama player untuk memberi script kepada player



Gambar 1.6 Script Player

7. Isikan script sebagai berikut:

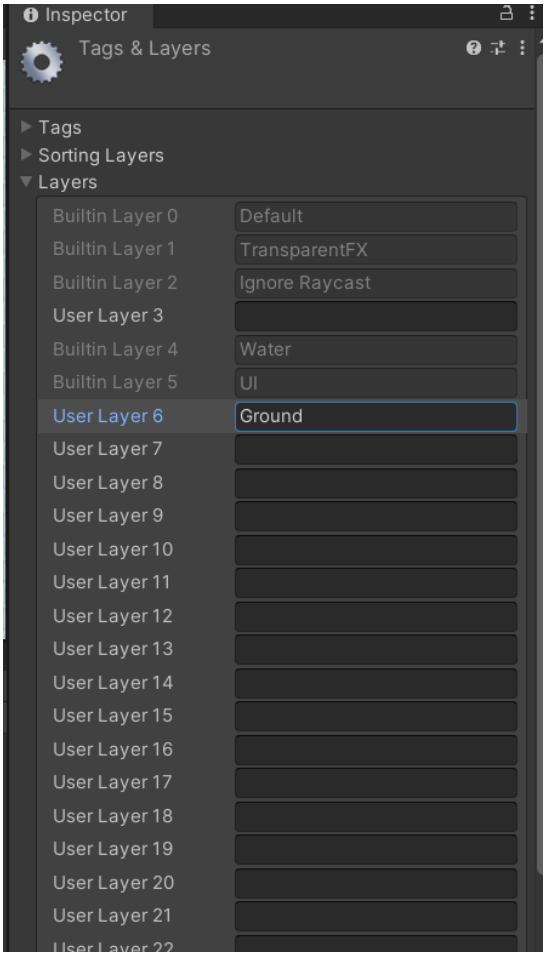
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Player : MonoBehaviour
{
  Rigidbody2D rb;

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

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

  float horizontalValue;

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

  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.po
      sition, groundCheckRadius, groundLayer);
        if (colliders.Length > 0)
        isGrounded = true;
        }

        void Move(float dir, bool jumpflag)
        {

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

          #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
        }
}
```

8. Selanjutnya coba untuk play game tersebut dan coba untuk menggerakan dengan tombol A untuk kekiri dan D untuk kekanan

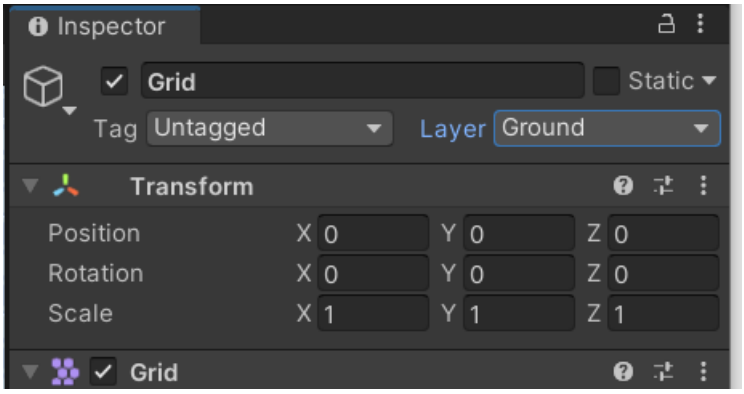
Gambar 1.8 Hasil Player

9. Buatkan list layer baru pada grid dengan nama ground



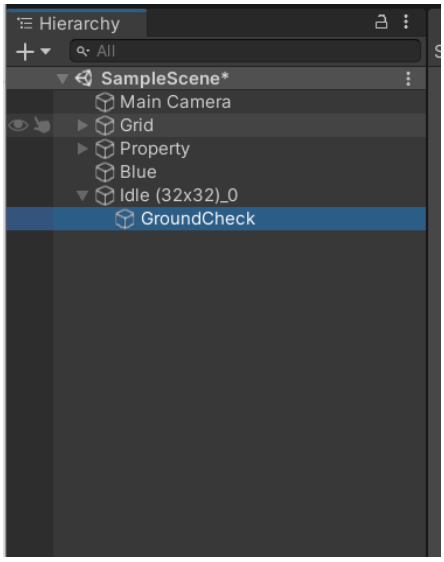
Gambar 1.9 List Layer

10. Rubah layernya menjadi ground



Gambar 1.10 Grid Layer

11. Selanjutnya Create Empty dan beri nama GroundCheck pada player



Gambar 1.11 Layer Depth

12. Klik pada hirarki groundcheck lalu gunakan move tools dan pindahkan kebawah player

Gambar 1.12 GroundCheck Point

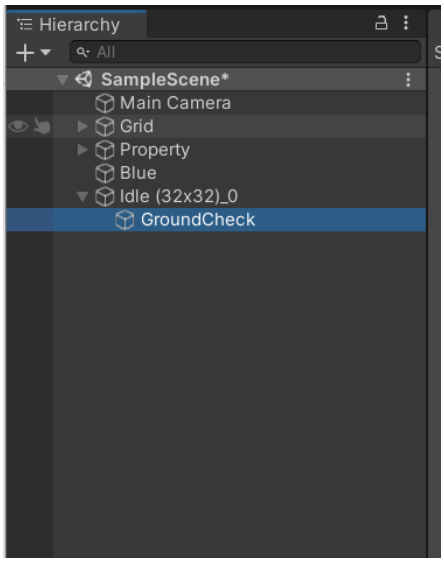
13. Masukkan script berikut kedalam script player tadi

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

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

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

14. Tambahkan juga void groundcheck dibawah void fixedupdate

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

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

15. Tarik script player sebelumnya ke hirarki groundcheck



Gambar 1.13 Memasukkan Script ke groundcheck

16. Buat fungsi untuk melompat dengan script berikut

```
      [SerializeField]    float    jumpPower    =    100;

        bool jump;
```

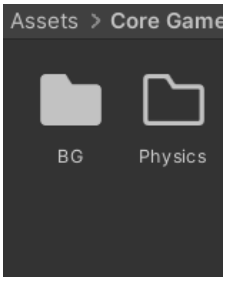
17. Tambahkan juga script dibagian void update

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

18. Tambahkan juga pada void move

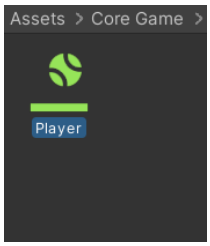
```
bool jumpflag

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

19. Buat folder physhics



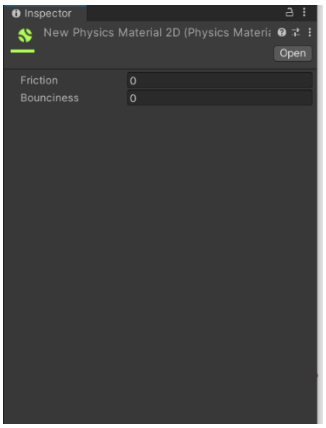
Gambar 1.14 Core Game > Physics

20. Buat physical material 2D didalam folder physics tadi



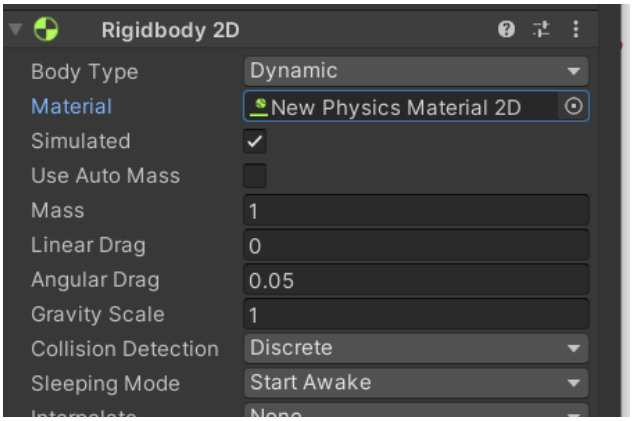
Gambar 1.15 Physical material 2D

21. Didalam inspectornya terdapat friction & bounce ubah menjadi 0



Gambar 1.16 Inspector physics

22. Masukkan material physics tadi kedalam rigid body 2D pada player

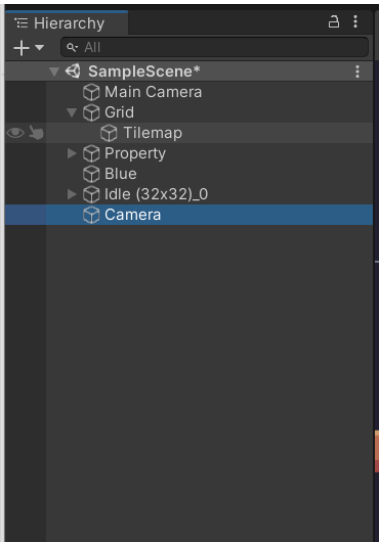


Gambar 1.17 Rigid Body 2D

23. Coba hasilnya, player akan bisa melompat dengan menekan spasi

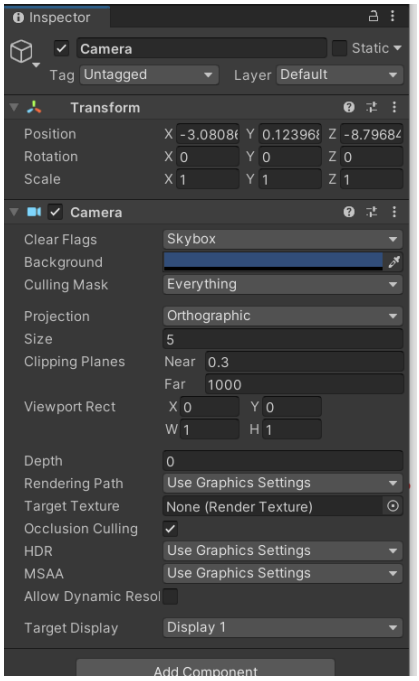
Gambar 1.18 Hasil Player Melompat

24. Langkah selanjutnya buat create empty lagi pada hirarki dengan nama Camera



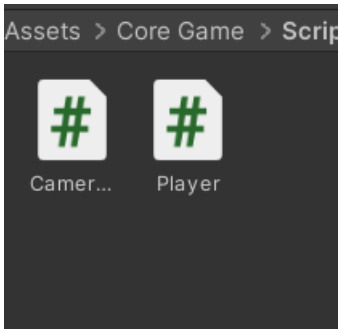
Gambar 1.19 Camera

25. Lalu pada inspector camera sesuai kan settingsnya seperti ini



Gambar 1.20 Inspector Camera

26. Buat script baru bernama CameraFollow



Gambar 1.21 Script CameraFollow

27. Lalu masukkan isi Script sebagai berikut:

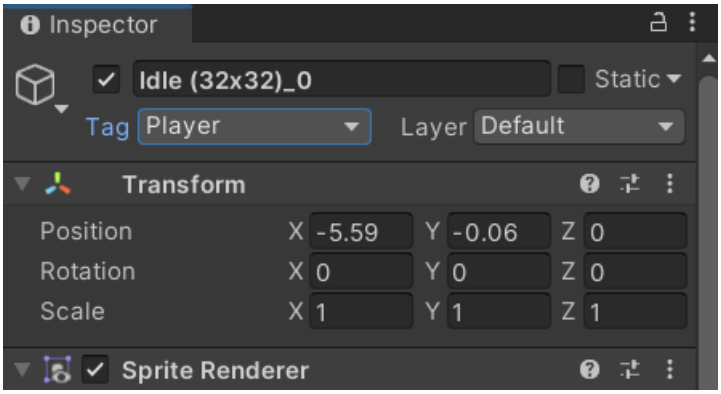
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

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").tra
nsform;
    }

    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);
    }
}
```

28. Drag and drop script tadi kedalam hirarki camera dan rubah settingan dari script camera pada inspectornya sebagai berikut:

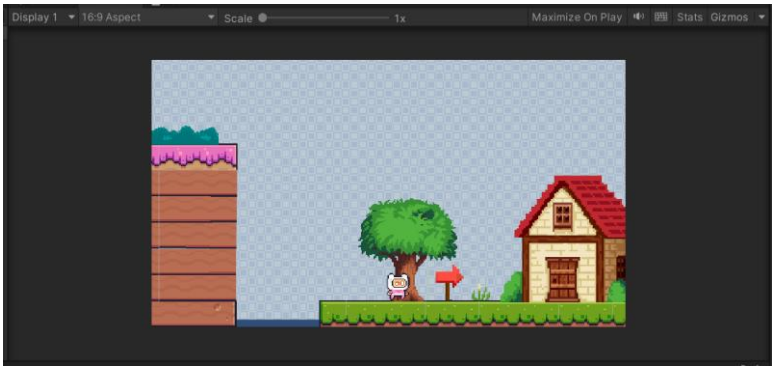
Gambar 1.22 Inspector Camera Script

29. Selanjutnya pada player idle sebelumnya ganti tagnya menjadi player



Gambar 1.23 Tag Player

30. Jika sudah maka hasilnya camera akan mengikuti gerakan character



Gambar 1.24 Hasil Camera Movement