TUGAS PERTEMUAN: 8

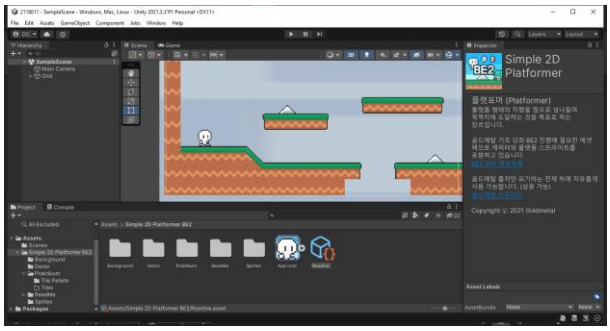
Camera & Character Movement

NIM	:	2118011
Nama	:	Dewa Chandra Agung Wibawa
Kelas	:	C
Asisten Lab	:	Naufal Dhiaurrafif (2218059)

1.1 Tugas 1 : Character Movement, Detect Ground, Jumping, & Camera Movement

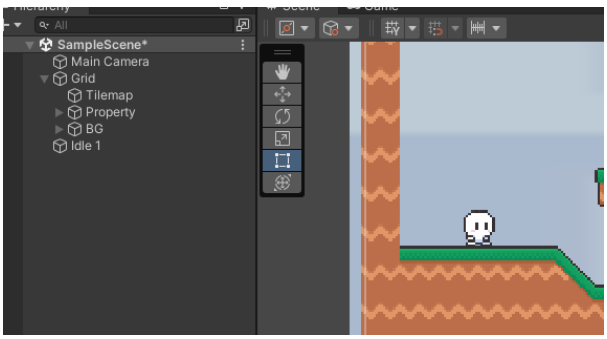
A. Membuat Character Movement

1. Pertama buka *project* yang telah dikerjakan di materi sebelumnya.



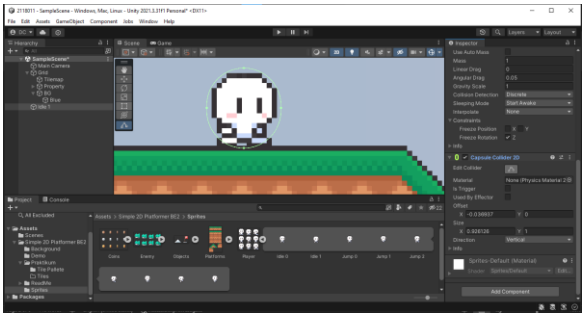
Gambar 1.1 Membuka Project

2. Setelah itu tambahkan karakter pada hirarki. Setelah ditammbahkan klik pada karakter dan tambahkan komponen *rigidbody 2d*. Lalu centang *freeze rotation z*.



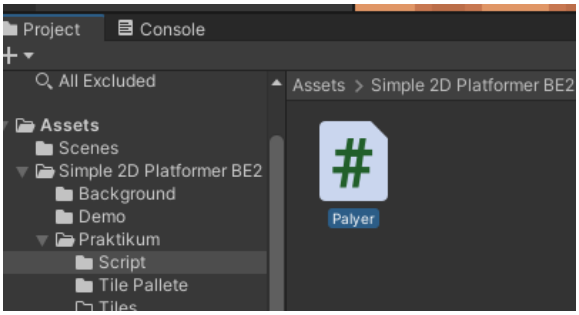
Gambar 1.2 Menambahkan Character Idle

3. Setelah itu tambahkan *capsule collider* pada karakter. Setelah itu cocokan garis oval pada karakter. Atau bisa juga dilakukan dengan meng-inputkan *offset* x, y, dan size x, y -nya.



Gambar 1.3 Menambahkan Capsule Colider

4. Setelah itu buat lagi *folder* baru Bernama “Script”. Lalu buat C# *script*. Lalu beri nama “Player”. Jika sudah maka hasilnya akan seperti gambar dibawah ini.



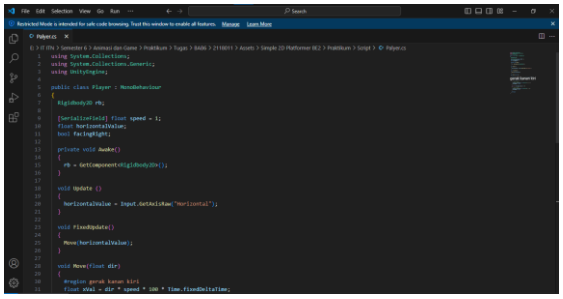
Gambar 1.4 Membuat Script Untuk Karakter

5. Berikutnya klik 2x pada script. Lalu tambahkan kode dibawah ini. Jika sudah, *drag and drop script* ke bagian karakter tadi. Jika sudah maka karakter akan bisa bergerak kekanan atau kekiri.

   Source code:

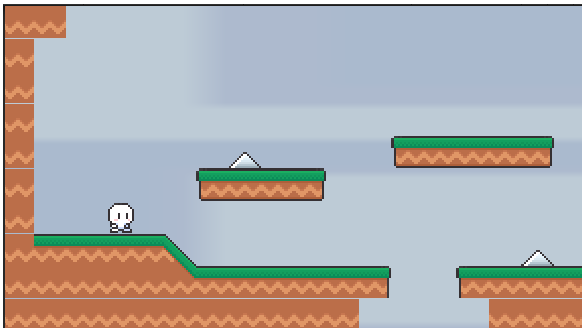
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

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
  }
}
```



Gambar 1.5 Membuat Karakter Agar Bisa Bergerak

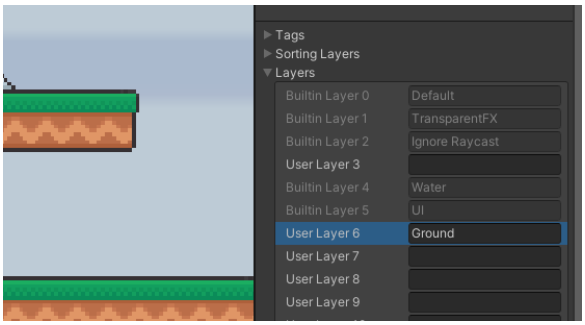
6. Setelah itu coba *source code*. Play dan tekan “a” untuk kea rah kiri. Lalu tekan “d” untuk kea rah kanan. Jika sudah maka hasilnya seperti gambar dibawah ini.



Gambar 1.6 Hasil Karakter Movement

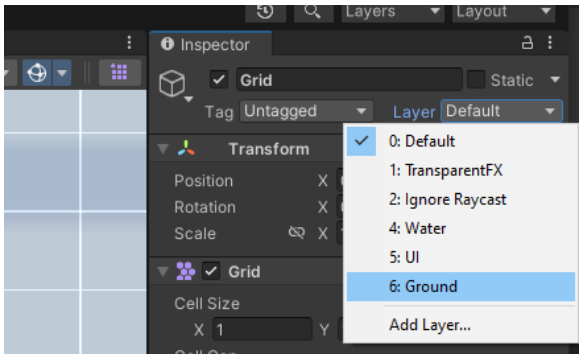
B. Detect Ground

1. Lalu buat *layer ground*. Klik bagian *grid*. Setelah itu lihat *inspector*. Setelah itu pilih *layer* dan klik *add layer*. Jika sudah maka akan seperti gambar dibawah ini.



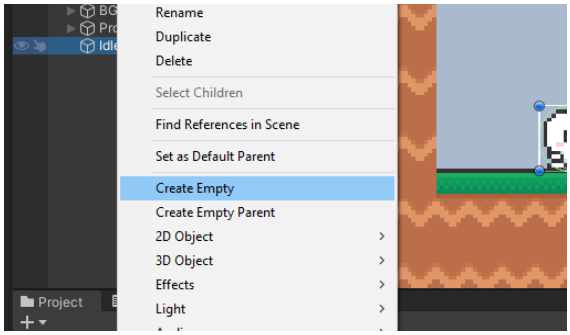
Gambar 1.7 Membuat Ground Check

2. Berikutnya, ubah layer grid menjadi *ground*. Pergi ke bagian *inspector* dan *layer ground.* Jika muncul notifikasi klik yang yes.



Gambar 1.8 Mengubah Layer Grid

3. Setelah itu klik kanan pada karakter. Lalu pilih *reate empty*. Lalu beri nama "*GroundCheck"*. Lalu pindahkan karakternya sedikit keatas dengan *move tools*.



Gambar 1.9 Membuat GroundCheck Pada Karakter

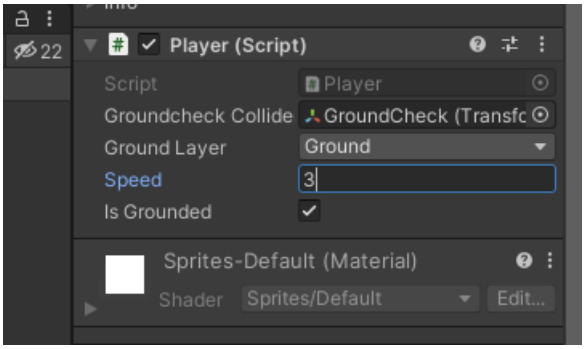
4. Lalu setelah itu kembali ke *script* karakter. Lalu Tambahkan beberapa source code berikut. Tambahkan juga *void ground check* dibawah *void fixedUpdate* & tambahkan GorunCheck(); pada *void fixedUpdate*.

Source Code:

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

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

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

Gambar 1.10 Menambahkan GroundCheck Pada Script

5. Lalu klik karakter. Lihat *inspector*, lalu ke effect Player script di bagian "*Grouncheck collider*". Setelah itu pilih *groundcheck transform*, serta pada *ground layer* pilih *ground*.



Gambar 1.11 Menjalankan GroundCheck pada Karakter

C. Membuat Jumping

1. Selanjutnya kita akan membuat karakter agar bisa meloncat. Rubah *script* seperti berikut pada *script* karakter.

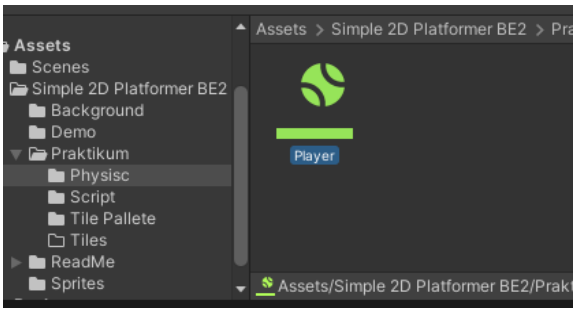
   Source code:

```
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.position,
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
```

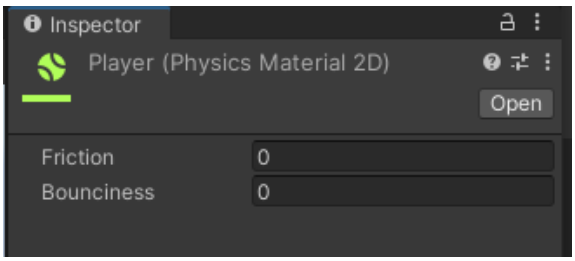
```
    }
}
```

2. Selanjutnya buat *folder* baru pada *folder* praktikum dan beri nama "Physics". Lalu didalamnya buat *physical material* 2D. Lalu beri nama "Player".



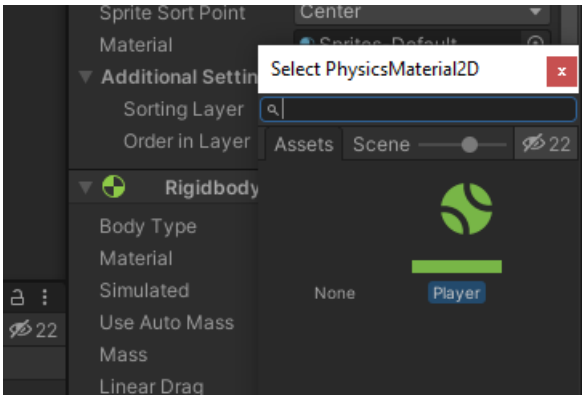
Gambar 1.12 Menambahkan Physical Material 2D

3. Selanjutnya klik *player phyisics*, lihat bagian *inspector*. Jika sudah ubah menjadi 0 pada bagian *friction* dan *bounces*nya.



Gambar 1.13 Mengubah Nilai Friction & Bounces

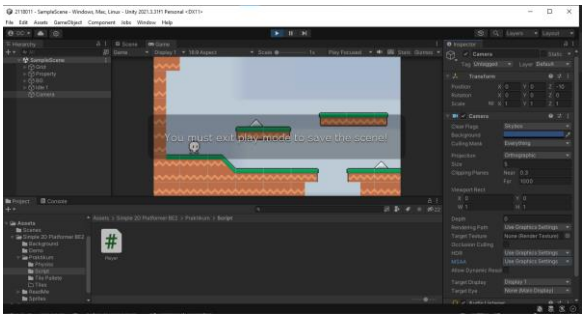
4. Selanjutnya klik hirarki dan pilih karakter. Setelah itu lihat *inspector* dan cari *rigidboy* 2D. Lalu buka *box select* pada material dan pilih *asset player* yang sudah dibuat tadi. Maka karakter sudah bisa meloncat.



Gambar 1.14 Menambahkan Karakter

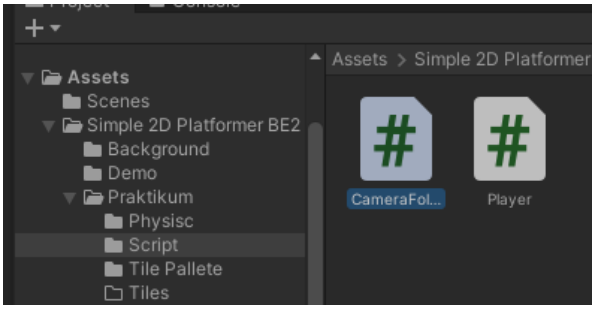
D. Camera Movement

1. Setelah itu tambahkan kamera pada hirarki. Lalu setting kameranya. Ubah tag *main camera* menjadi *untagged*. Lalu ubah settingannya seperti gambar dibawah ini.



Gambar 1.15 Mengatur Kamera

2. Lalu buat *cript* baru. Pada *folder script* buat *script* baru. Setelah itu beri nama *script* "CameraFollow". Jika sudah maka klik 2x untuk mengedit *script*.



Gambar 1.16 Menambahkan Scirpt Untuk Kamera

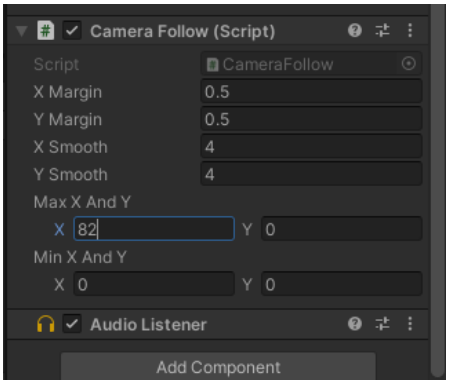
3. Lalu klik 2x pada *cript* "*CameraFollow*". Jika sudah tambahkan *source code* berikut. Jika sudah tinggal di *drag and drop* pada hirarki di bagian kamera.

   Source code:

```
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;
```

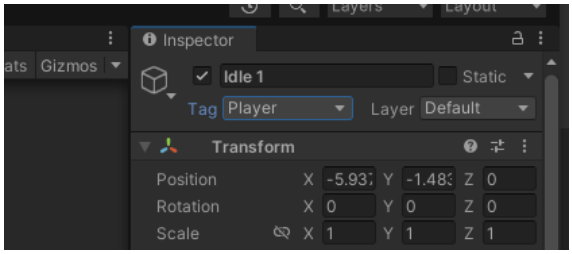
```
    void Awake()
    {
        player                                          =
GameObject.FindGameObjectWithTag("Player").transform;
    }

    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. Setelah itu klik kamera pada hirarki. Lihat pada bagian *inspector* dan cari bagian *camera follow*. Jika sudah ketemu atur seperti pada gambar dibawah ini.



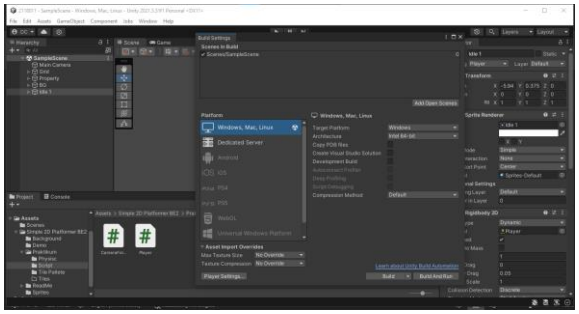
Gambar 1.17 Mengatur Script Kamera

5. Selanjutnya kembali ke bagian karakter. Lalu lihat ke bagian *inspector*. Jika sudah pada bagian *tag*, rubah menjadi *player*. Jika sudah maka tekan play untuk menguji coba.



Gambar 1.18 Ubah Tag Pada Bagian Karakter

6. Selanjutnya tinggal *render* saja. Pergi ke *file*, lalu pilih *build setting*. Jika sudahatur pada *build setting*, dan pilih tempat menyimpan. Jika sudah maka prosesnya hingga selesai.



Gambar 1.19 Merender Game

E. 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 Source code

Program dilakukan untuk membuat kamera mengikuti player atau pemain secara horizontal. Terdapat void update yang terus menerus memperbaharui posisi kameranya berdasarkan lokasi pemain. Sehingga kamera bisa bergerak ke kiri dan ke kanan.