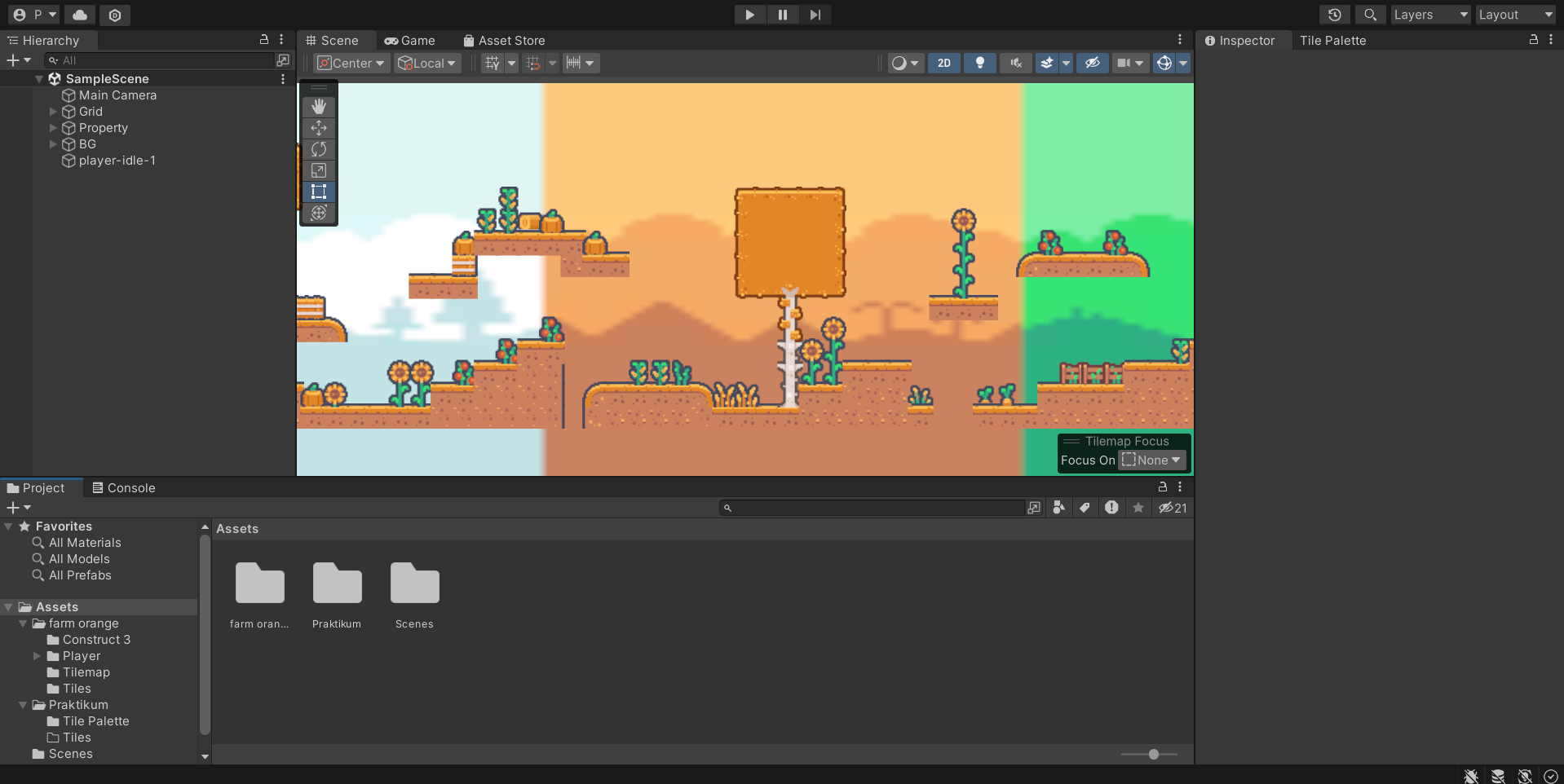
# 8 CAMERA & CHARACTER MOVEMENT

|  |  |  |
| --- | --- | --- |
| **NIM** | : | 2118126 |
| **Nama** | : | Prita Patricia Lakzmi |
| **Kelas** | : | D |
| **Asisten Lab** | : | Wisando Berlian P. (2218095) |

## Tugas 8 : Membuat Character Movement, Detect Ground, Jumping & Camera Movement

Membuat Character Movement, Detect Ground, Jumping, & Camera Movement Tidak Termasuk Animasi Character.

1. **Pergerakan Player**
2. Buka *Project Unity* pada 2118126\_BAB7 sebelumnya yang telah dibuat sebelumnya.

****

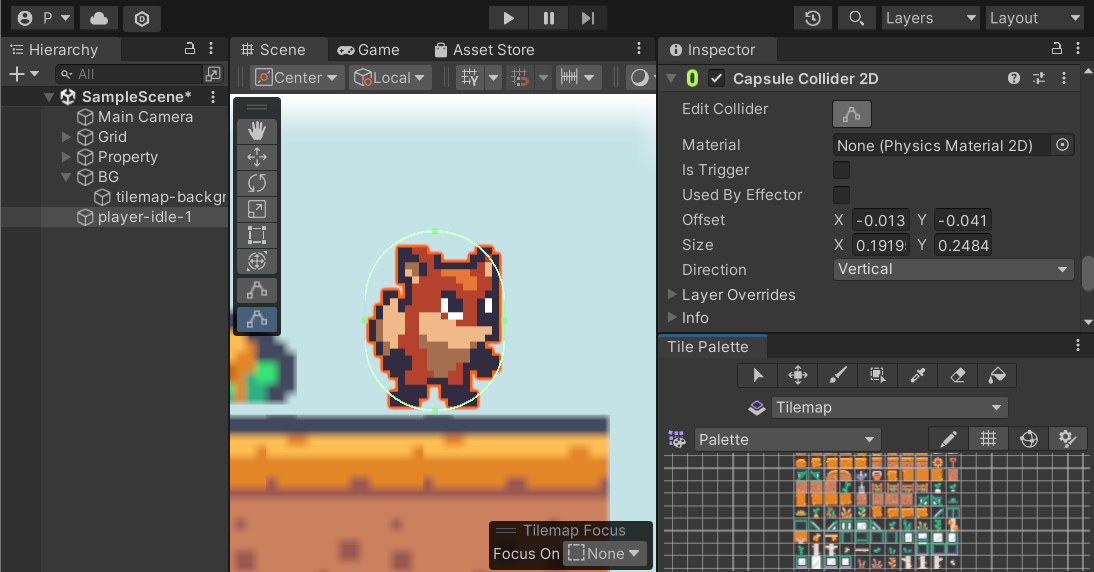
### Buka *Project* 2118126\_BAB7

1. Pada *inspector* dari *Hierarchy* *player-id*, *Add Component* dengan nama *Rigibody* *2D.*

****

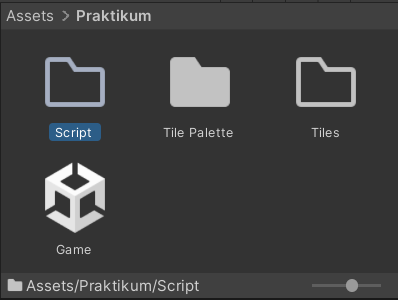
### *Add Component Rigibody 2D*

1. Dan tekan lagi *Add Component* dengan nama *Capsule Collider 2D*, kemudian tekan *icon* disebelah *Edit Collider*. Lalu rapikan garis lingkaran hingga menempel pada badan *player-id*.



### *Add Component Capsule Collider 2D*

1. Buat 1 buah *folder* baru pada *folder* praktikum dengan nama “*Script*”.



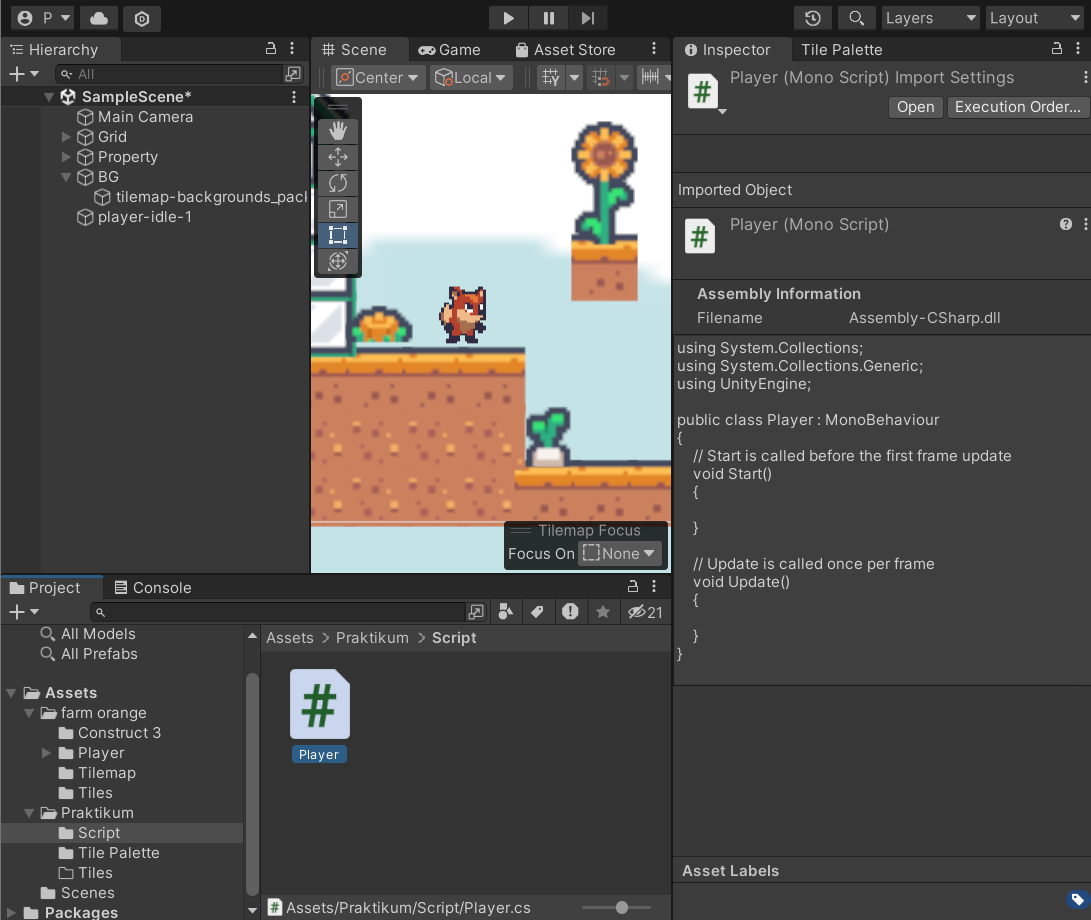
### Membuat 1 Buah *Folder* Dengan Nama Script

1. Di dalam *folder* praktikum buat 1 buah C# *Script* dengan nama “*Player*”.



### Membuat 1 Buah *Folder* Dengan Nama *Player*

1. *Drag and drop script Player* ke dalam *Hierarchy* *player-id* dan klik 2x pada *script Player*, maka akan muncul *code* di sebelah kanan dan masuk ke dalam *text editor* untuk memasukkan *code* dari *player*.

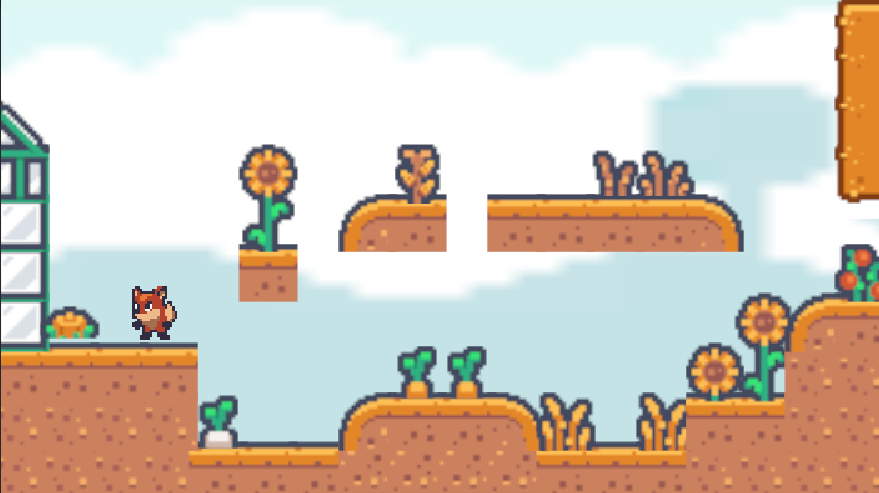


### *Drag And Drop Script Player*

1. Masukkan *source code* dibawah ini untuk *Player*, jika sudah lakukan Ctrl+S untuk *save code*.

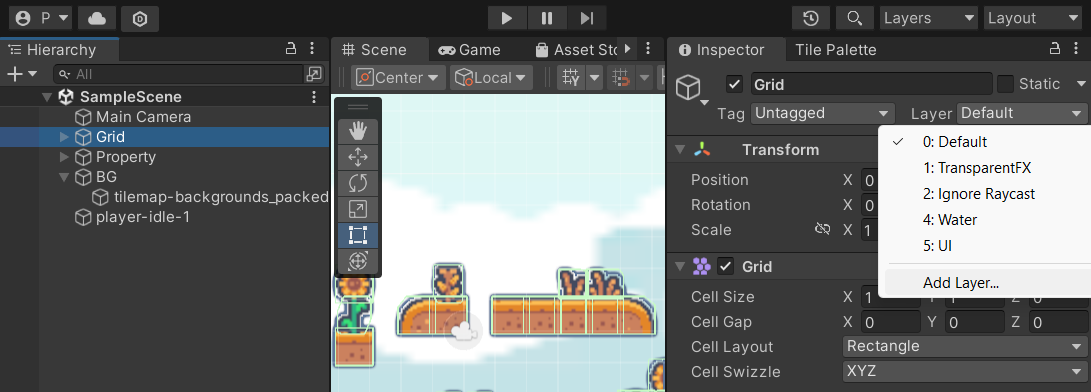
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Player : MonoBehaviour  {  Rigidbody2D rb;  [SerializeField] float speed = 1;  float horizontalValue;  bool facingRight;  private void Awake()  {  rb = GetComponent<Rigidbody2D>();  }  void Update ()  {  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(-5, 5, 5);  facingRight = false;  }  else if (!facingRight && dir > 0)  {  // ukuran player  transform.localScale = new Vector3(5, 5, 5);  facingRight = true;  }  #endregion  }  } |

1. Jalankan dengan cara menekan *icon play*, untuk mencoba apakah *source* *code* berhasil, tekan *keyboard left arrow* atau *right arrow*.



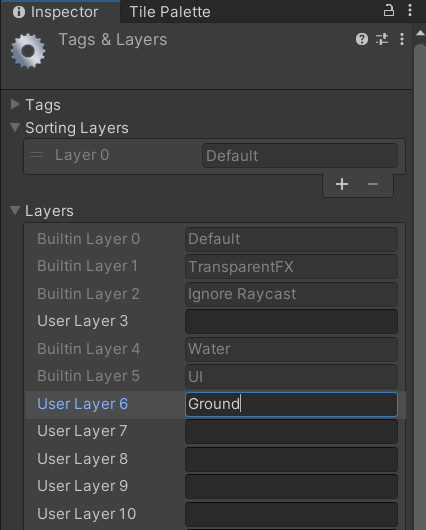
### Tampilan Saat Di *Play*

1. Ubah layer pada *Hierarchy Grid* dengan cara klik *Add Layer*. Ini adalah langkah awal untuk membuat player dapat melompat jika ditekan *keyboard* spasi.



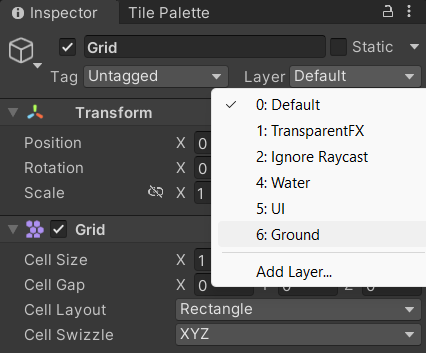
### *Add Layer* Pada *Inspector Grid*

1. Isikan “*Ground*” pada *User Layer* 6.



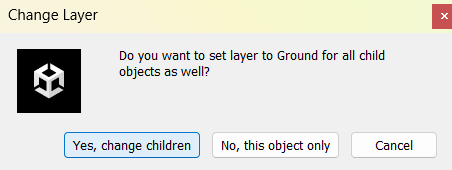
### *User Layer* 6 Menjadi “*Ground*”

1. Ubah layer menjadi *Ground*. Seperti pada gambar dibawah ini.



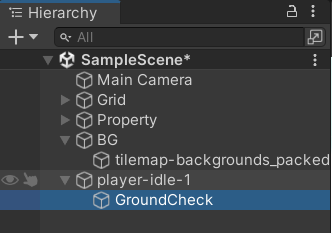
### Ubah Layer Menjadi *Ground*

1. Tekan ‘*Yes, change* *childern’*, jika muncul *pop up* *‘Change Layer’*.



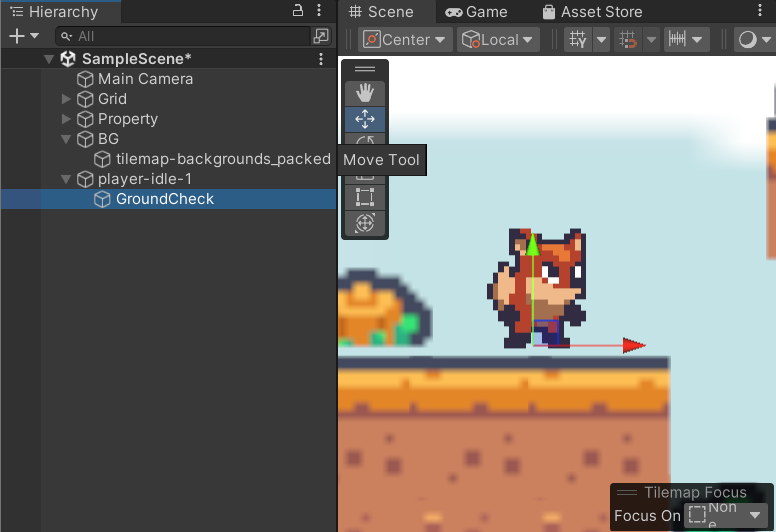
### *Pop Up Change Layer*

1. Klik kanan pada *Hierarchy* dan *Create Empty*, kemudian beri nama “*GroundCheck*”.



### *Create Empty* Dengan Nama *GroundCheck*

1. Tekan *Hierarchy* *GroundCheck*, kemudian gunakan *Move Tools* untuk memindahkan tanda panah sedikit kebawah, seperti pada gambar dibawah ini.

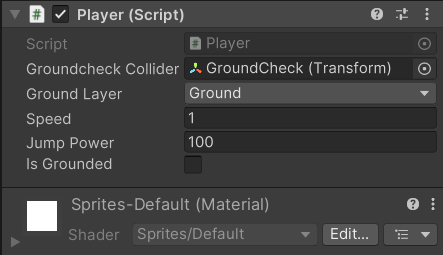


### *Move Tool GroundCheck*

1. Kembali ke *script Player* dan ubah source code untuk *Player* seperti *source* *code* dibawah ini.

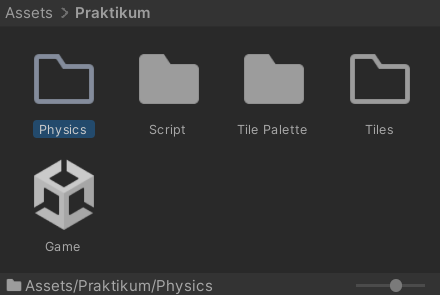
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Player : MonoBehaviour  {  Rigidbody2D rb;  [SerializeField] Transform groundcheckCollider;  [SerializeField] LayerMask groundLayer;  const float groundCheckRadius = 0.2f; // +  [SerializeField] 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(-5, 5, 5);  facingRight = false;  }  else if (!facingRight && dir > 0)  {  // ukuran player  transform.localScale = new Vector3(5, 5, 5);  facingRight = true;  }  #endregion  }  } |

1. Tekan *player-id* pada *Hierarchy* ubah *Ground* Layer pada *inspector* menjadi *Ground* dan untuk *GroundCheck* *Collider* menjadi *GroundCheck*.



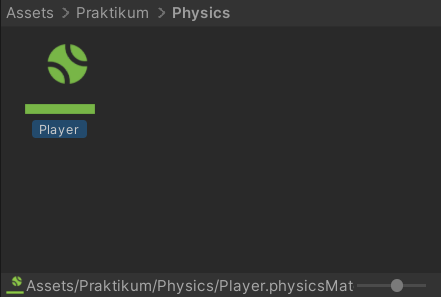
### Atur *Component Player (Script)*

1. Buat 1 buah *folder* baru di dalam *folder* Praktikum dengan nama “*Physics*”.



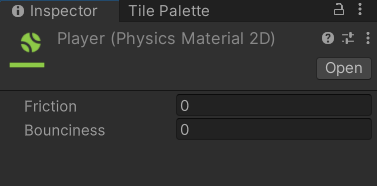
### Membuat 1 Buah Folder Dengan Nama *Physics*

1. Di dalam *folder Physics* klik kanan dan *Create > 2D > Physics Material 2D.* dan beri nama “*Player*”.



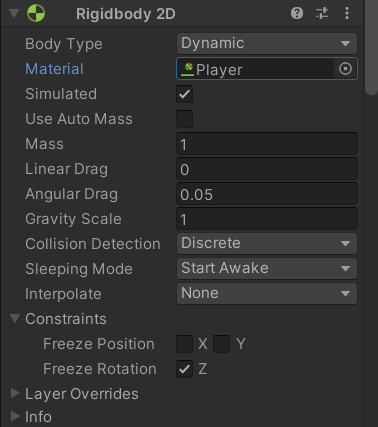
### Membuat 1 Buah *Physics Material 2D*

1. Klik *Player* dan ubah nilai pada *Friction* di bagian *inspector* menjadi 0.



### *Inspector Player (Physics Material 2D)*

1. Pada *component Rigibody 2D* dibagian *inspector* pada *Hierarchy* *player-id*, ubah Material nya menjadi *Player*.



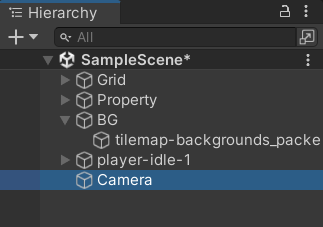
### *Inspector Rigibody 2D* Pada *Folder* *Player*

1. Jalankan dengan cara menekan *icon play*. Tekan spasi pada *keyboard* untuk melihat apakah *player* dapat melompat dan *code* yang dimasukkan benar.



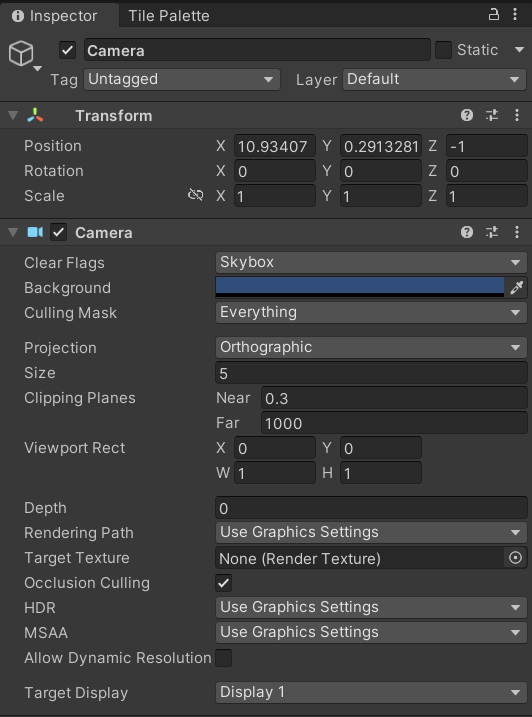
### Tampilan Saat Di *Play*

1. **Camera Movement**
2. Tambahkan 1 buah *Create Empty* pada *Hierarchy* dan ubah namanya menjadi “*Camera*”.



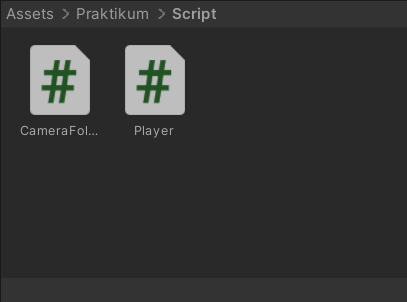
### Menambahkan *Camera*

1. Pada *inspector* di bagian camera *setting* *position* z dengan nilai -1.



### *Setting Inspector* Pada *Camera*

1. Buat file *script* baru di dalam *folder* *Script* dengan nama “*CameraFollow*”.



### Membuat *Folder* *Script* Dengan Nama *CameraFollow*

1. Tekan 2 kali pada file CameraFollow dan berikan code seperti dibawah ini.

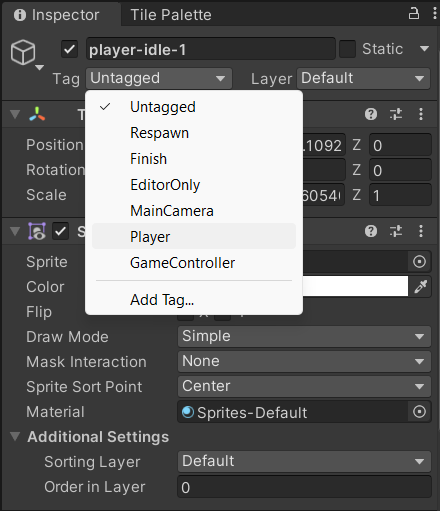
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class CameraFollow : MonoBehaviour  {  public float xMargin = 0.5f;  public float yMargin = 0.5f;  public float xSmooth = 4f;  public float ySmooth = 4f;  public Vector2 maxXAndY;  public Vector2 minXAndY;  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);  }  } |

1. Atur Max X And Y pada *inspector* *Camera Follow (Script).*



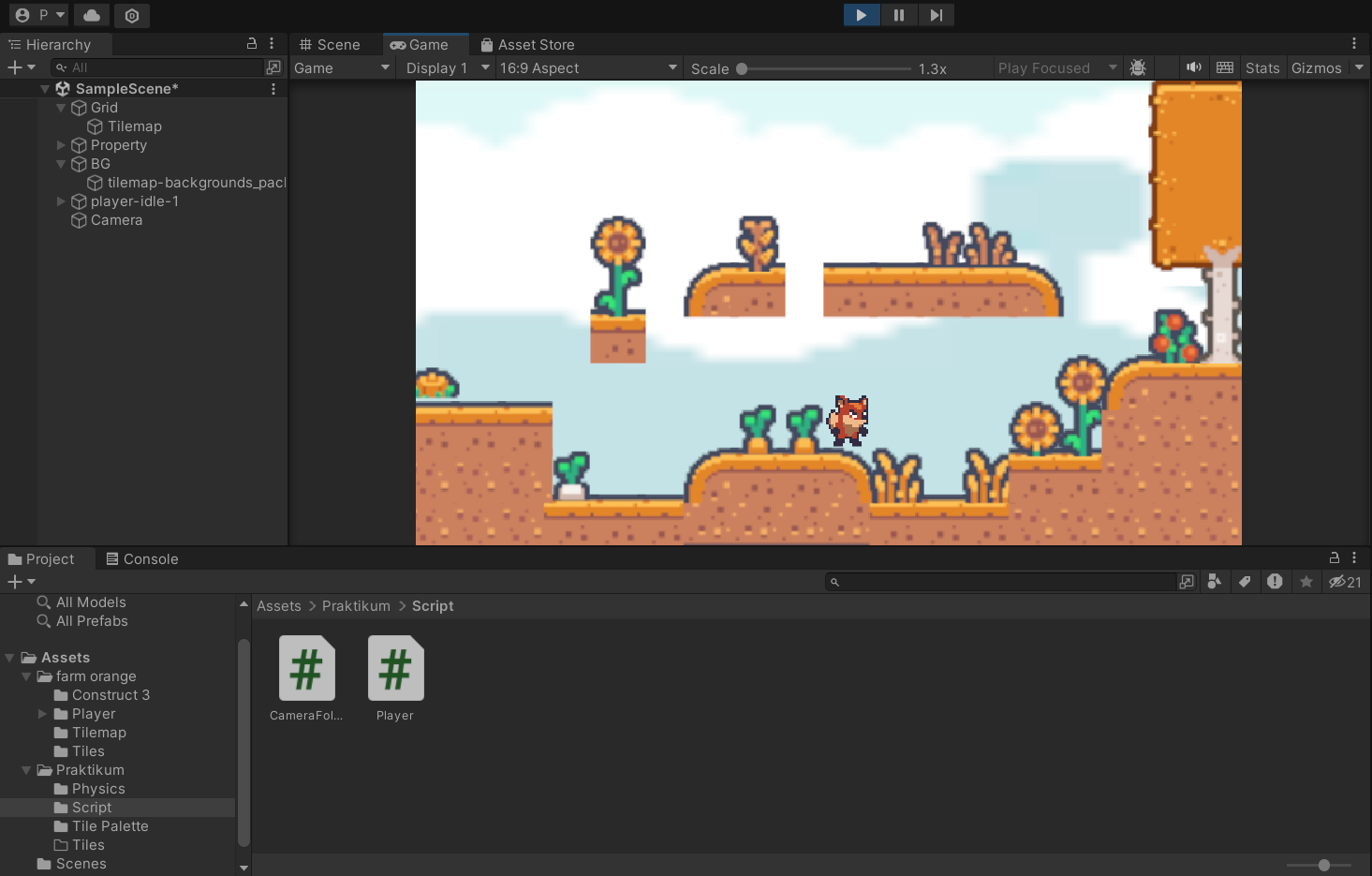
### Atur *Inspector Component Camera Follow (Script)*

1. Ubah tag pada *inspector player-id* yang sebelumnya *Untagged* menjadi *Player*.



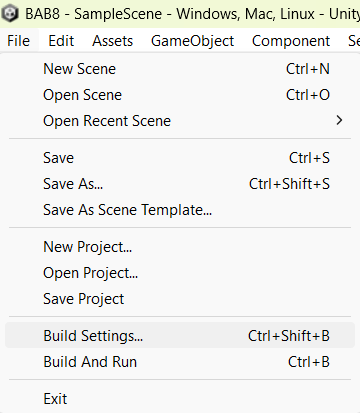
### Ubah Tag Menjadi *Player*

1. Tekan *icon play* untuk menjalankan, akurasi keberhasilan dapat dilihat ketika *player* digerakkan maka camera akan mengikuti pergerakan dari *player*.



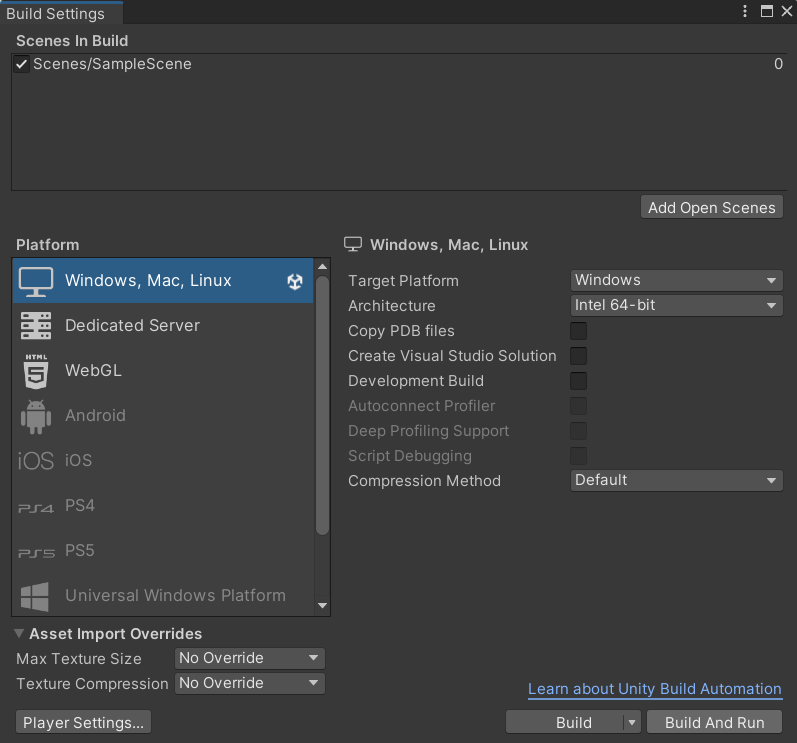
### Tampilan Saat Di *Play*

1. **RENDER**
2. Klik *Build Settings* pada menu *File*.

****

### Menu File Build Settings

1. Selanjutnya simpan project dengan cara klik *button Build And Run*. Dan hasil render akan tersimpan pada *folder* yang telah ditentukan.

****

### *Build And Run*

1. **KUIS CameraFollow**

|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class CameraFollow : MonoBehaviour  {  [SerializeFiled] private Transform player;  void Update(){  transform.position = new Vector3 (player. Position.x,transform.position.y, transform.position.z);  }  } |

Penjelasan :

*Source code* diatas adalah *code* untuk membuat *camera follow* atau *camera* dapat mengikuti karakter (*player*) dalam *game*. Dengan menggunakan *library* yang dibutuhkan. Variabel *player* dengan tipe *Transform* digunakan untuk merujuk ke objek karakter (*player*). Void *update* digunakan untuk membuat camera bergerak dan terpanggil ketika karakter (*player*) berpindah tempat.