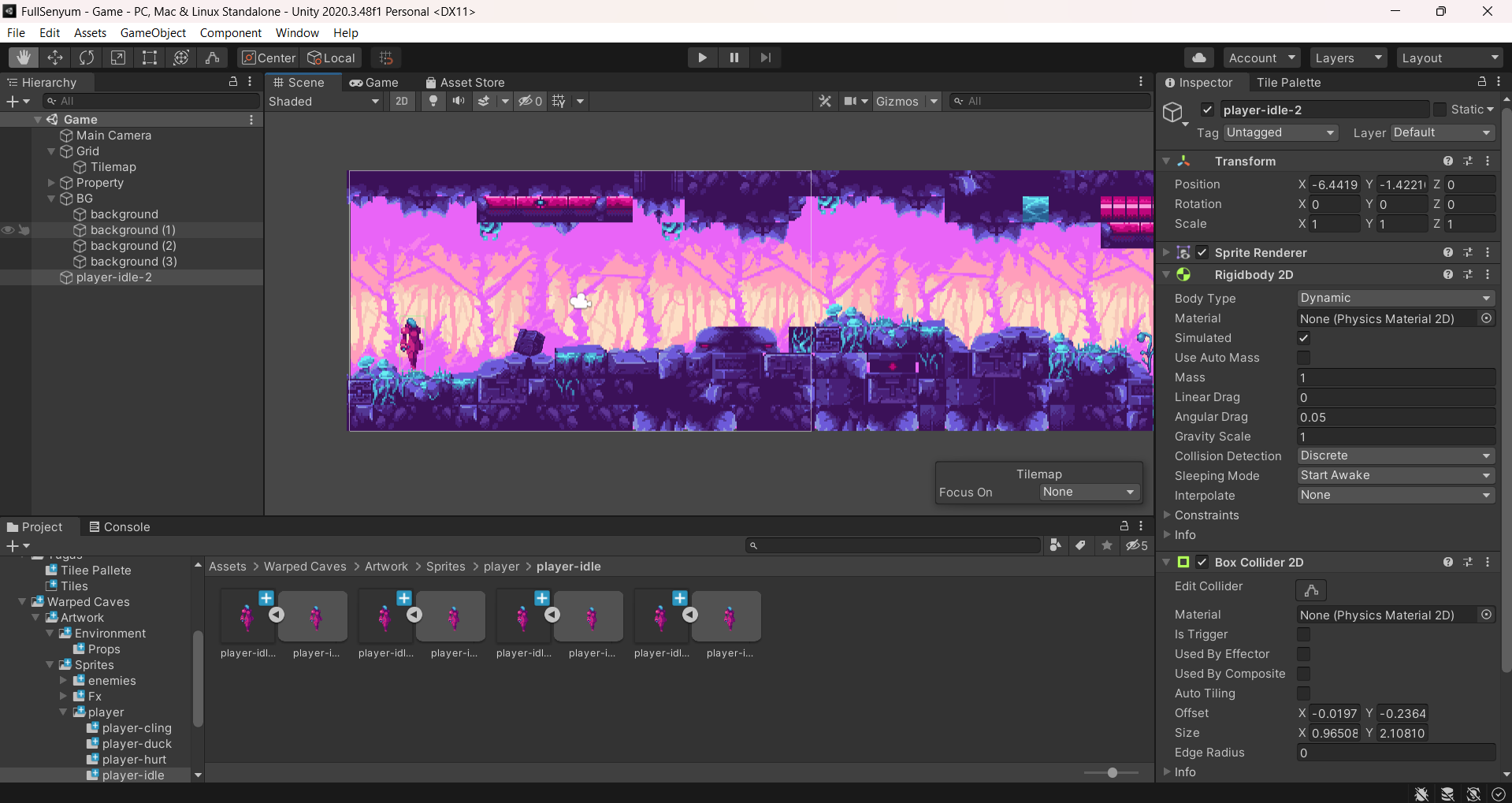
# 8 Camera & Character Movement

|  |  |  |
| --- | --- | --- |
| **NIM** | : | I Kadek Oka Sukrisna |
| **Nama** | : | 2118036 |
| **Kelas** | : | B |
| **Asisten Lab** | : | Rifal Rifqi Rhomadon (2218106) |

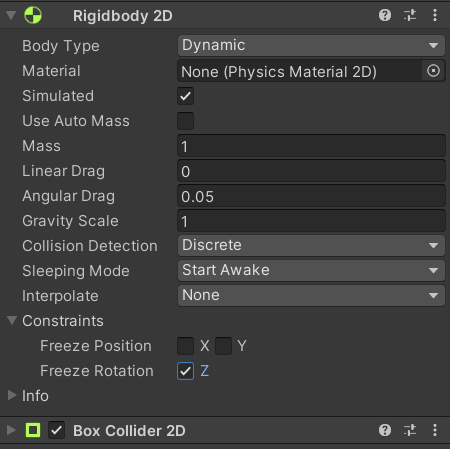
## Tugas 8 : Membuat Camera & Character Movement

1. Buka file projek Unity sebelumnya pada bab 7 untuk digunakan kembali Tambahkan player bernama hero-idle-1, import kedalam hirarki



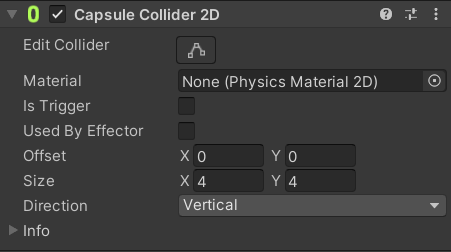
### Tampilan Project Sebelumnya

1. Klik hero-idle-1 tambahkan Component Rigidbody 2D, sesuaikan settingannya seperti gambar berikut, Centang pada Freeze Rotation Z



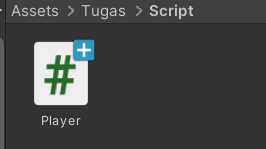
### Menambahkan component RigidBody 2D

1. Lalu tambahkan komponen Capsule Colider di hero-idle-1, lalu klik icon sebelah kanan edit collider. Lalu cockan garis oval degan karakternya atau bisa di inputkan Offset X, Y dan juga Size X, Y nya



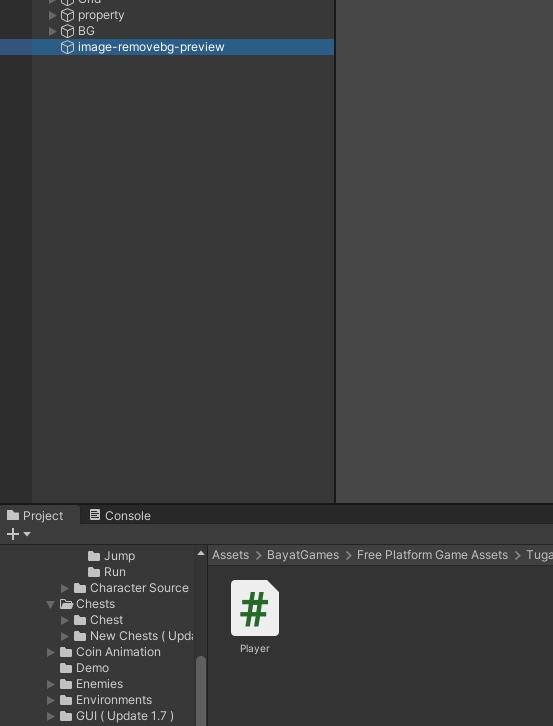
### Menambahkan component Capsule Collider 2D

1. Buka folder Tugas, lalu bikin folder baru bernama Script. Masuk kedalam folder Script, lalu buat C# Script, beri nama Player



### Membuat Script Player

1. Drag & drop script player kedalam hirarki hero-idle-1, lalu klik 2x pada script player maka akan masuk kedalam text editor seperti ini.

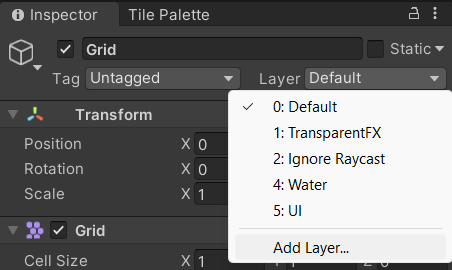


### Memasukkan script ke dalam player

1. Masukan source code dibawah ini, pastikan nama public class harus sama dengan nama file yang dibuat.

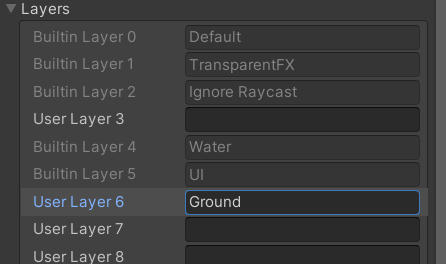
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Player : MonoBehaviour  {  Rigidbody2D rb;  [SerializeField] float speed = 1;  float horizontalValue;  bool facingRight;  private void Awake()  {  rb = GetComponent<Rigidbody2D>();  }  void Update ()  {  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  }  } |

1. Untuk membuat player loncat menggunakan spasi, kita perlu membuat GorundCheck dengan cara, klik Grid pada Hierarchy, pergi ke inspector, pilih Layer, Klik Add Layer[​](https://moduls-anigames.netlify.app/docs/modul-anigame/bab8/modul#11-untuk-membuat-player-loncat-menggunakan-spasi-kita-perlu-membuat-gorundcheck-dengan-cara-klik-grid-pada-hierarchy-pergi-ke-inspector-pilih-layer-klik-add-layer)



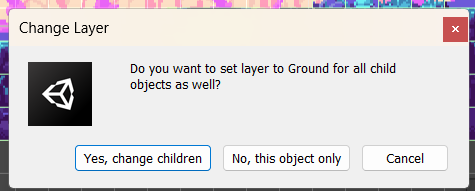
### Menambah layer pada grid

1. Lalu isi “Ground” pada User Layer 6



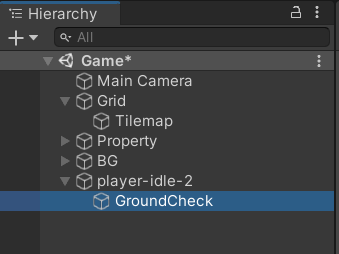
### Membuat layer 6 ground

1. Ubah Layer menjadi Ground, jika muncul pop up Change Layer, klik yes saja.



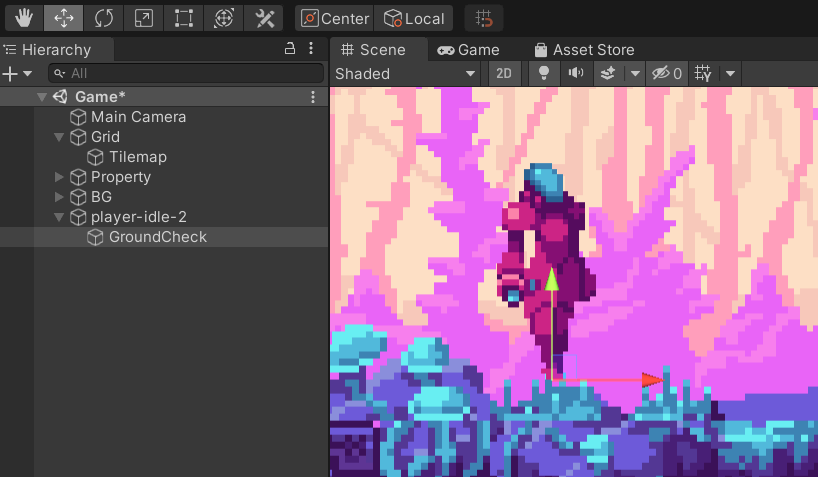
### Mengubah layer grid

1. Klik kanan pada hero-idle-1, lalu Create empty, beri nama GorundCheck.



### Membuat GroundCheck

1. Klik pada Hirarki GorundCheck, lalu gunakan “Move Tools” untuk memindahkan ke bagian bawah Player seperti gambar berikut.



### Mengubah posisi GroundCheck

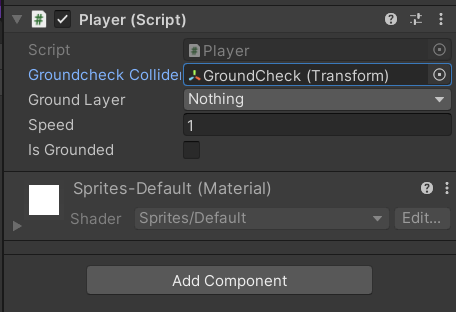
1. Kembali ke script Player tambahkan source code seperti ini. Ubah 3 baris kode yang berada dibawah Rigidbody2D rb;

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

1. Buat void ground check dibawah void fixedUpdate & tambahkan GroundCheck(); pada void fixedUpdate

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

1. Klik hero-idle-1, lalu ke inspector ke effect Player script di bagian “Groundcheck collider” tekan icon lalu pilih yang GorundCheck Transform, dan pada Ground Layer pilih Ground



### Mengubah properti script player

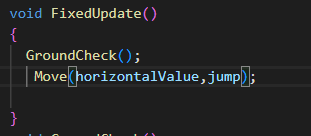
1. Lalu untuk membuat player melompat tambahkan script berikut

|  |
| --- |
| [SerializeField] float jumpPower = 100;  bool jump; |

1. Tambahkan juga script berikut di bagian void update.

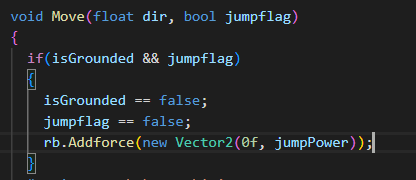
|  |
| --- |
| if (Input.GetButtonDown("Jump"))  jump = true;  else if (Input.GetButtonUp("Jump"))  jump = false; |

1. Tambahkan juga parameter jump pada fungsi Move.



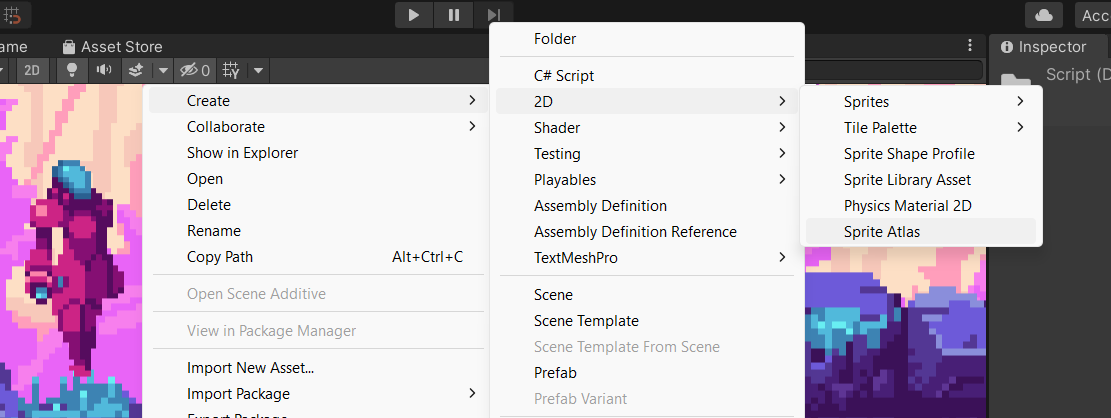
### Menambah parameter

1. Tambahkan script berikut pada void Move[​](https://moduls-anigames.netlify.app/docs/modul-anigame/bab8/modul#22-tambahkan-script-berikut-pada-void-move).



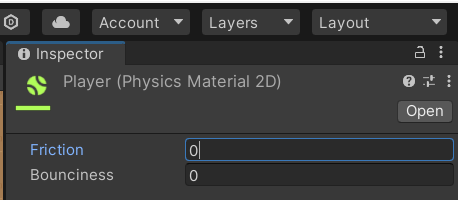
### Mengubah script Move()

1. Buat folder baru di TugasPrak dengan nama “Physics”. Didalam folder Pyshics create > 2d > physical material 2d , beri nama “Player”.



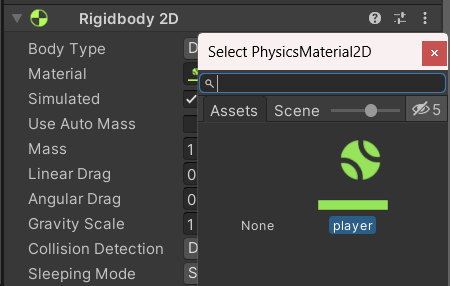
### Menambahkan Physical Material 2D

1. Klik Player (Physics Material 2D), dibagian menu inspector, friction & bounces ubah menjadi 0



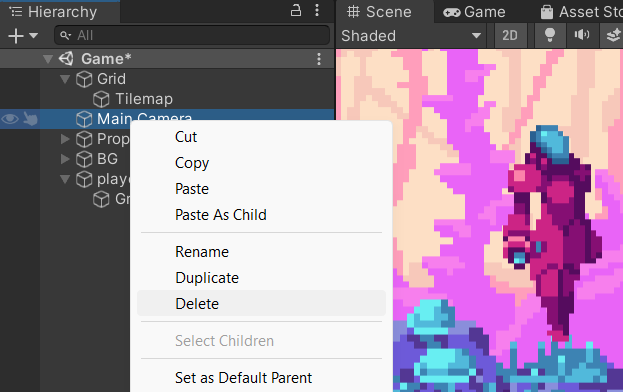
### Mengubah properti Physics Material 2D

1. Klik Hierarchy pilih layer hero-idle 1, pada Inspector Cari Rigidbody 2D lalu klik icon untuk membuka box select physhics material 2d , lalu pilih asset Player yang sudah kita buat tadi



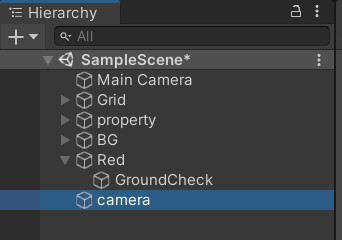
### Mengubah material RigidBody 2D

1. Hapus Main Camera dari hierarchy



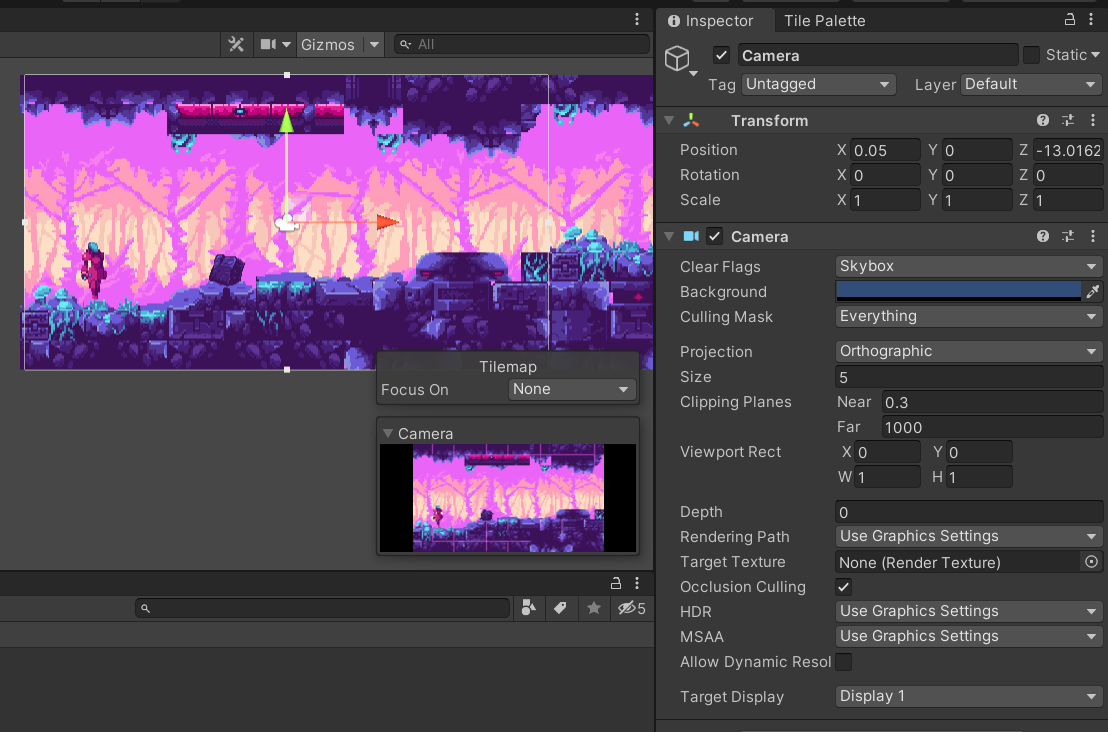
### Menghapus Main Camer

1. Create Empty pada Hirarki, dan Rename Menjadi Camera



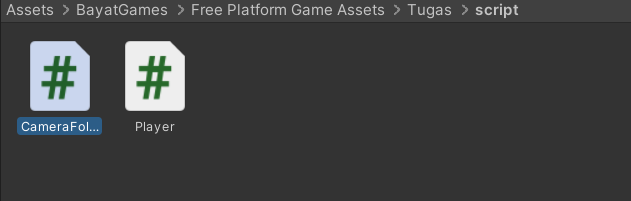
### Menambahkan Hierarcy (Camera)

1. Tambahkan component camera, lalu ubah setting component seperti berikut.



### Mengubah properti component Camera

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



### Membuat script CameraFollow

1. Lalu tuliskan script berikut ini, kemudian drag & drop script CameraFollow Kedalam Layer Camera

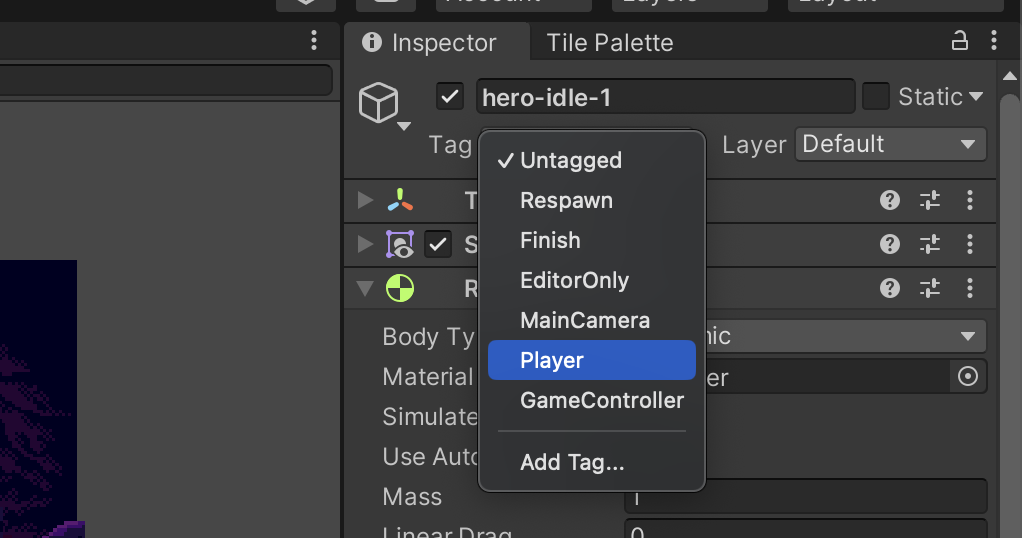
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class CameraFollow : MonoBehaviour  {  public float xMargin = 0.5f;  public float yMargin = 0.5f;  public float xSmooth = 4f;  public float ySmooth = 4f;  public Vector2 maxXAndY;  public Vector2 minXAndY;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

1. Lalu klik pada camera, buka inspector Pada bagian Camera Follow (Script) Ubah Bagian Max X dan Max Y nya



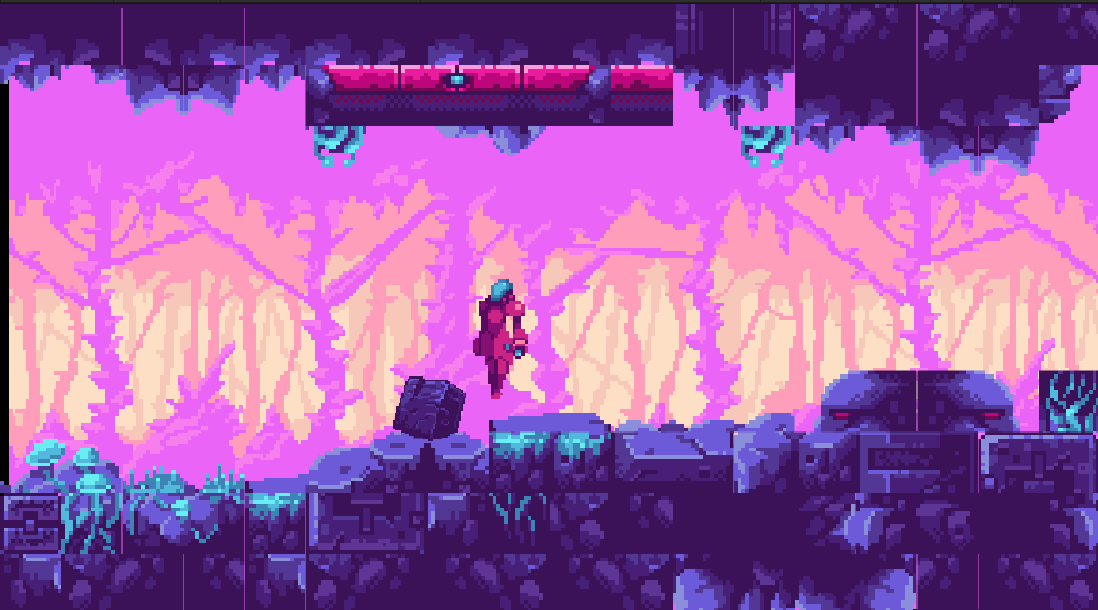
### Mengubah properti script Camera Follow

1. Ubah tag di hero-idle-1 Untagged menjadi ”Player”



### Mengubah tag dari hero

1. Tekan play untuk menjalankan, maka sekarang kamera akan mengikuti pergerakan karakter



### Hasil Camera & Character Movement

## Kuis Camera Follow

|  |
| --- |
| 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 :

Kode di atas mendefinisikan sebuah script bernama CameraFollow yang membuat kamera mengikuti pemain pada sumbu x. Dalam script ini, variabel player dari tipe Transform diatur sebagai serialized field agar bisa diatur melalui Inspector di Unity. Pada metode Update, yang dipanggil setiap frame, posisi kamera disesuaikan agar memiliki posisi x yang sama dengan pemain, sementara posisi y dan z dari kamera tetap tidak berubah. Dengan cara ini, kamera akan selalu mengikuti pergerakan pemain ke kiri atau ke kanan. Perlu diperhatikan bahwa kata kunci Void harus diubah menjadi void untuk memastikan script berfungsi dengan benar.