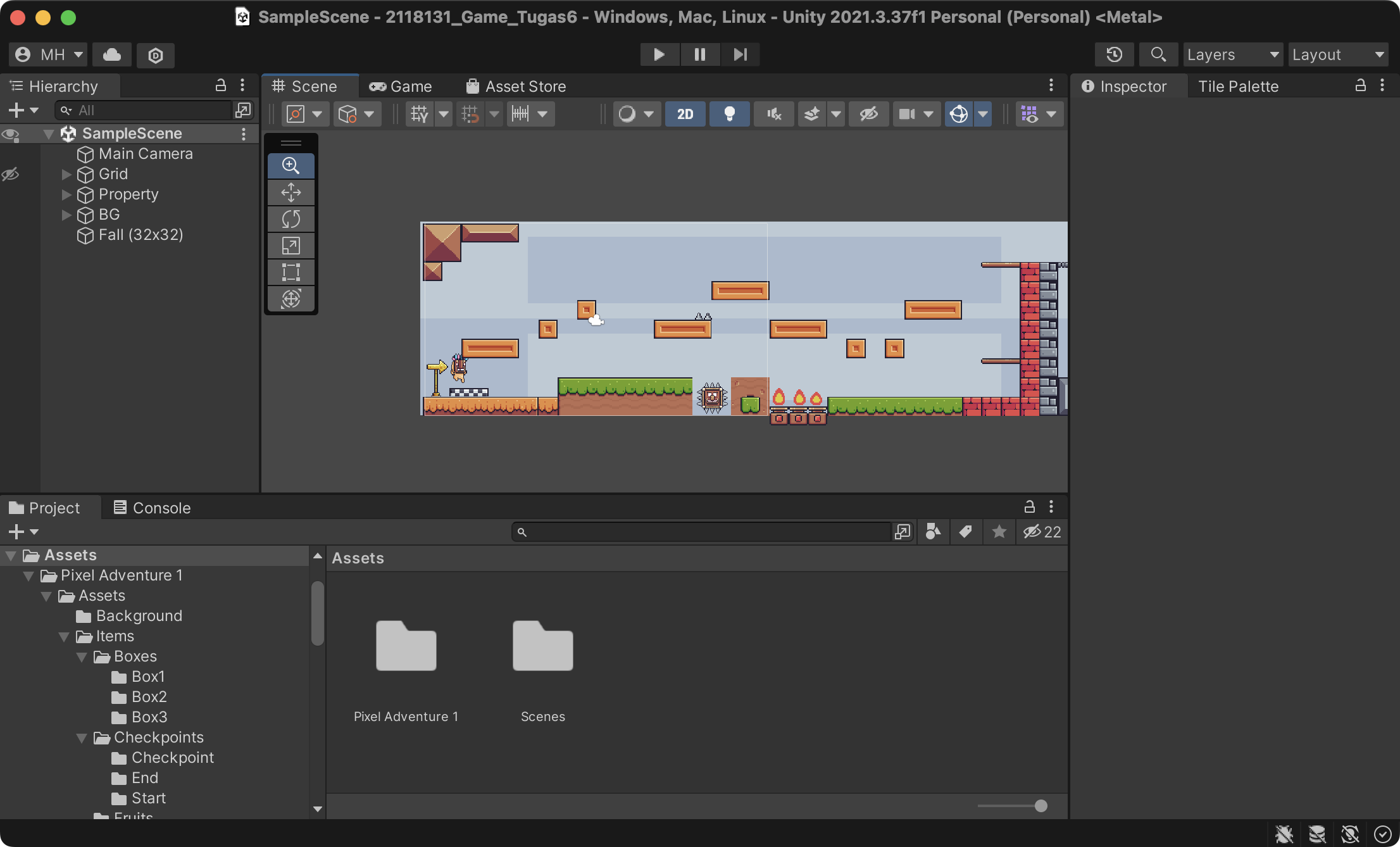
# 8 Camera & Character Movement

|  |  |  |
| --- | --- | --- |
| **NIM** | : | 2118131 |
| **Nama** | : | Mohammad Harifin |
| **Kelas** | : | D |
| **Asisten Lab** | : | Wisando Berlian Pandensolang (2218095) |

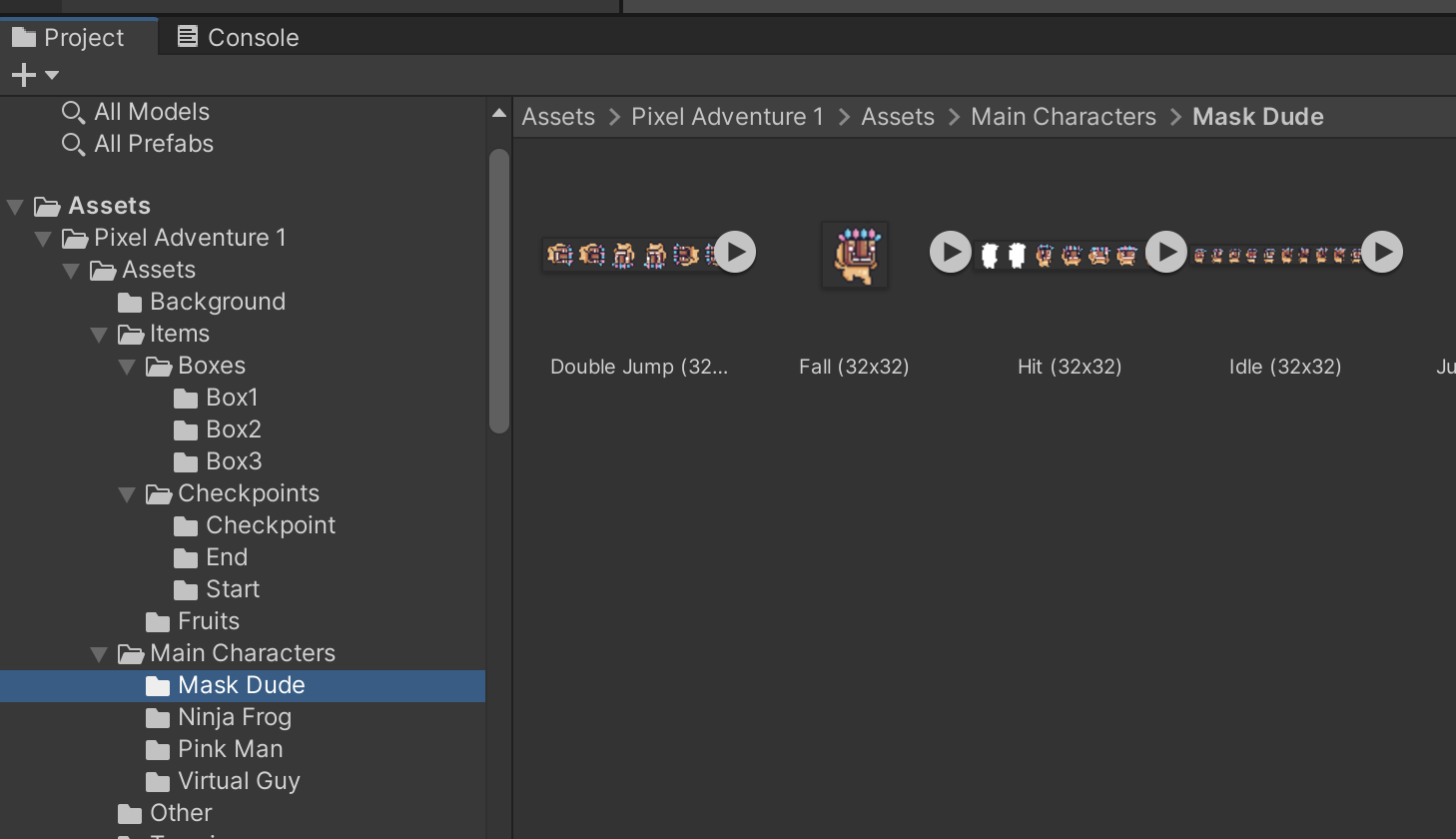
## Tugas 1 : Membuat Character Movement, Detect Ground, Jumping, & Camera Movement Tidak Termasuk Animasi Karakter

1. **Membuat Pergerakan Player**
2. Buka file projek Unity sebelumnya pada bab 7 untuk digunakan kembali



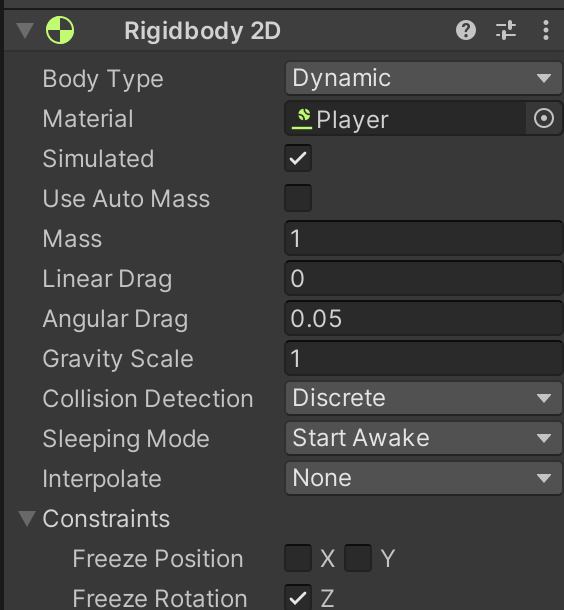
Gambar 8.1 Hasil Tampilan project Sebelumnya

1. Tambahkan player bernama Fall, pilih yang Mask Dude, Import kedalam Hirarki.



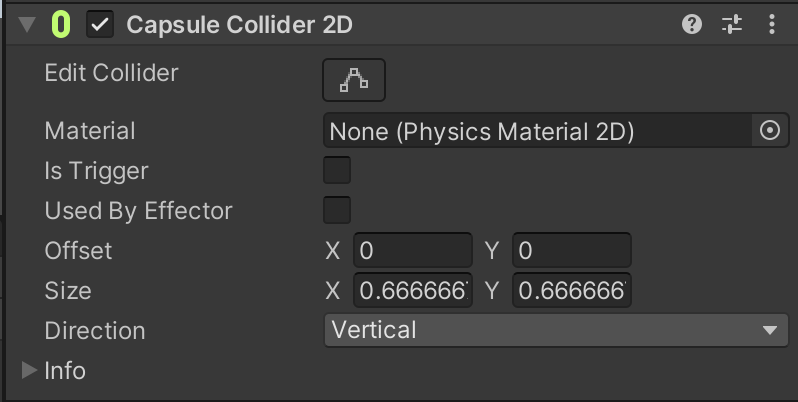
Gambar 8.2 Hasil Tampilan Creat Player Character

1. Klik Fall (32x32), tambahkan Component Rigidbody 2D, sesuaikan settingannya seperti gambar berikut, Centang pada Freeze Rotation Z



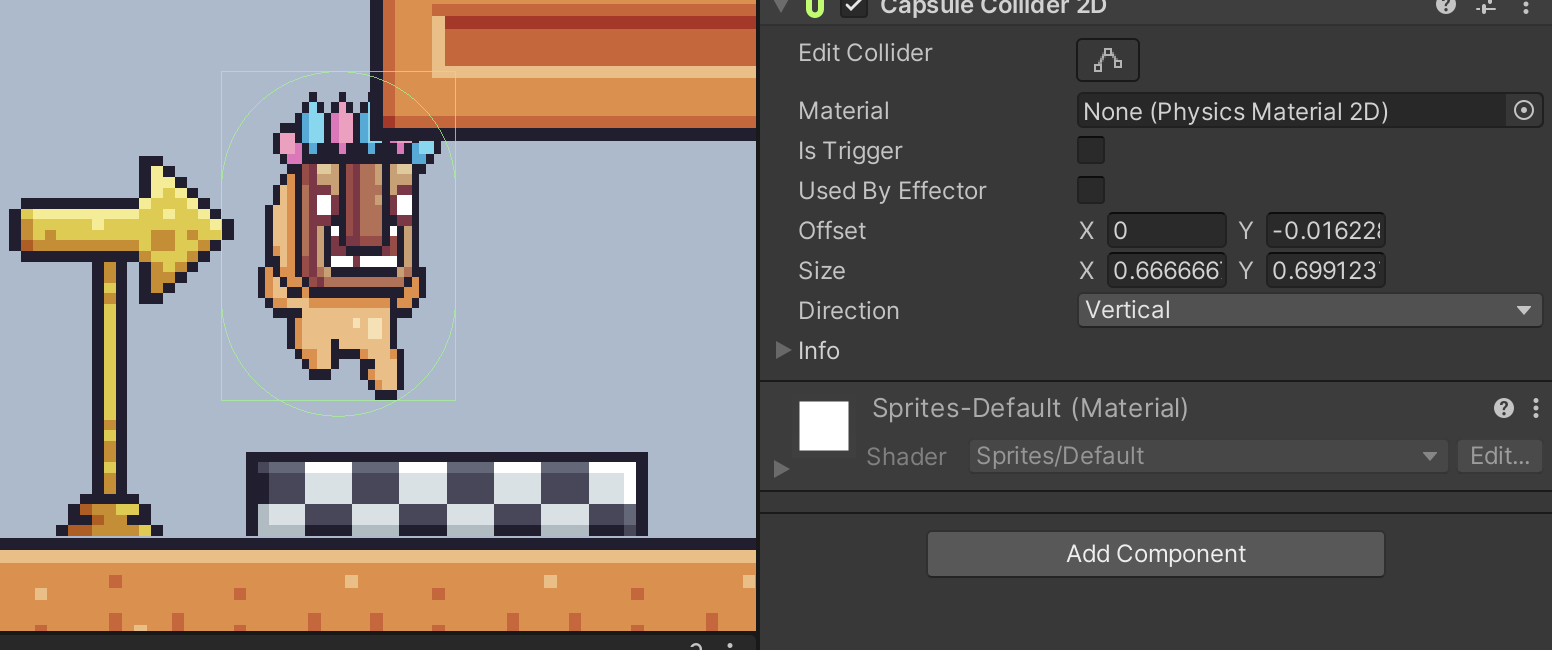
Gambar 8.3 Hasil Tampilan Centang pada Freeze Rotation Z

1. Lalu tambahkan komponen Capsule Colider di Fall(32x32), lalu klik icon sebelah kanan edit colider



Gambar 8.4 Hasil Tampilan Creat Capsule Colider

1. Lalu cocokkan garis oval degan karakternya atau bisa di inputkan Offset X, Y dan juga Size X, Y nya



Gambar 8.5 Hasil Tampilan garis oval

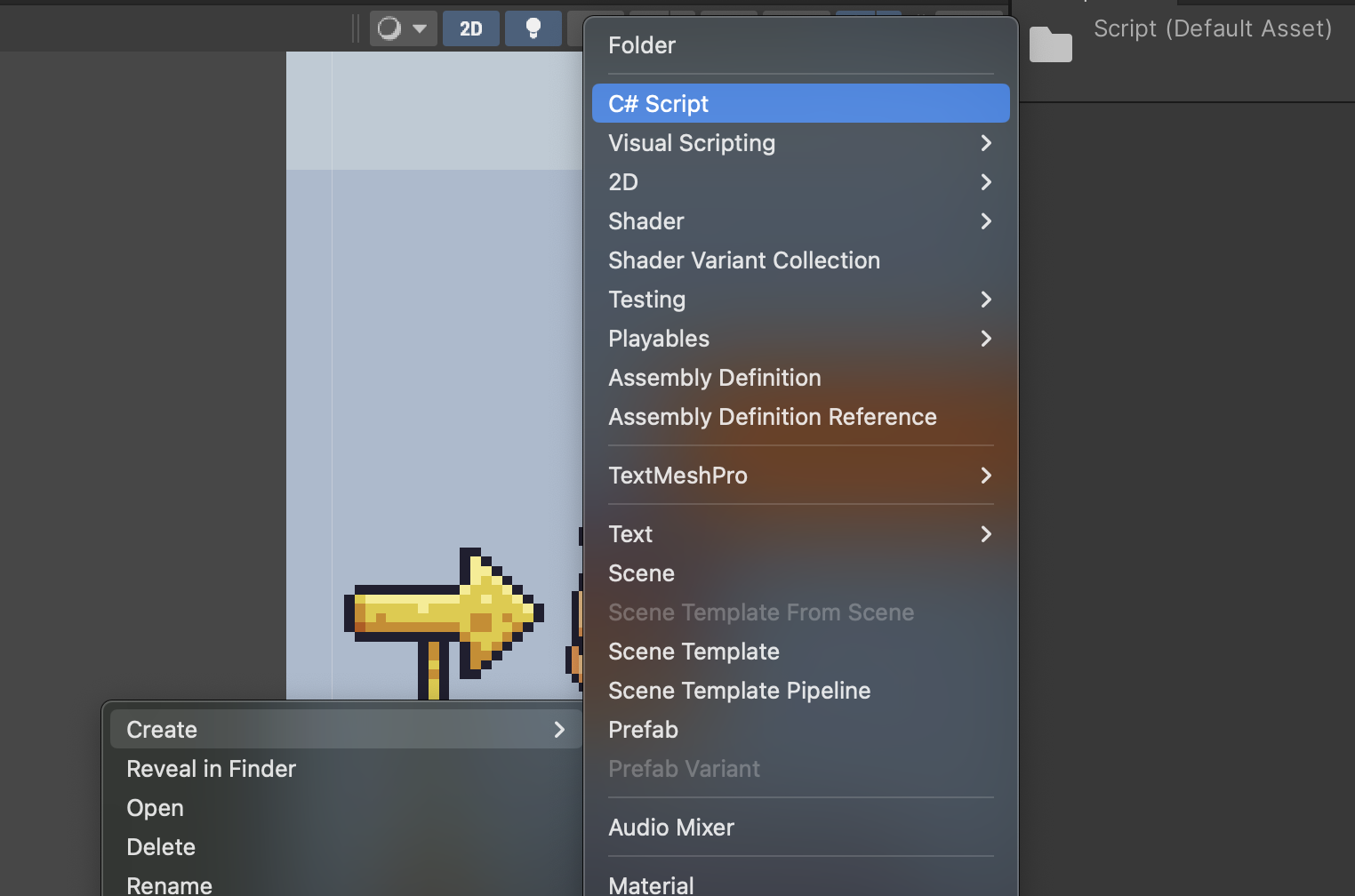
1. Buka Folder Game\_2118131, lalu bikin folder baru bernama Script

A screenshot of a computer

Description automatically generated

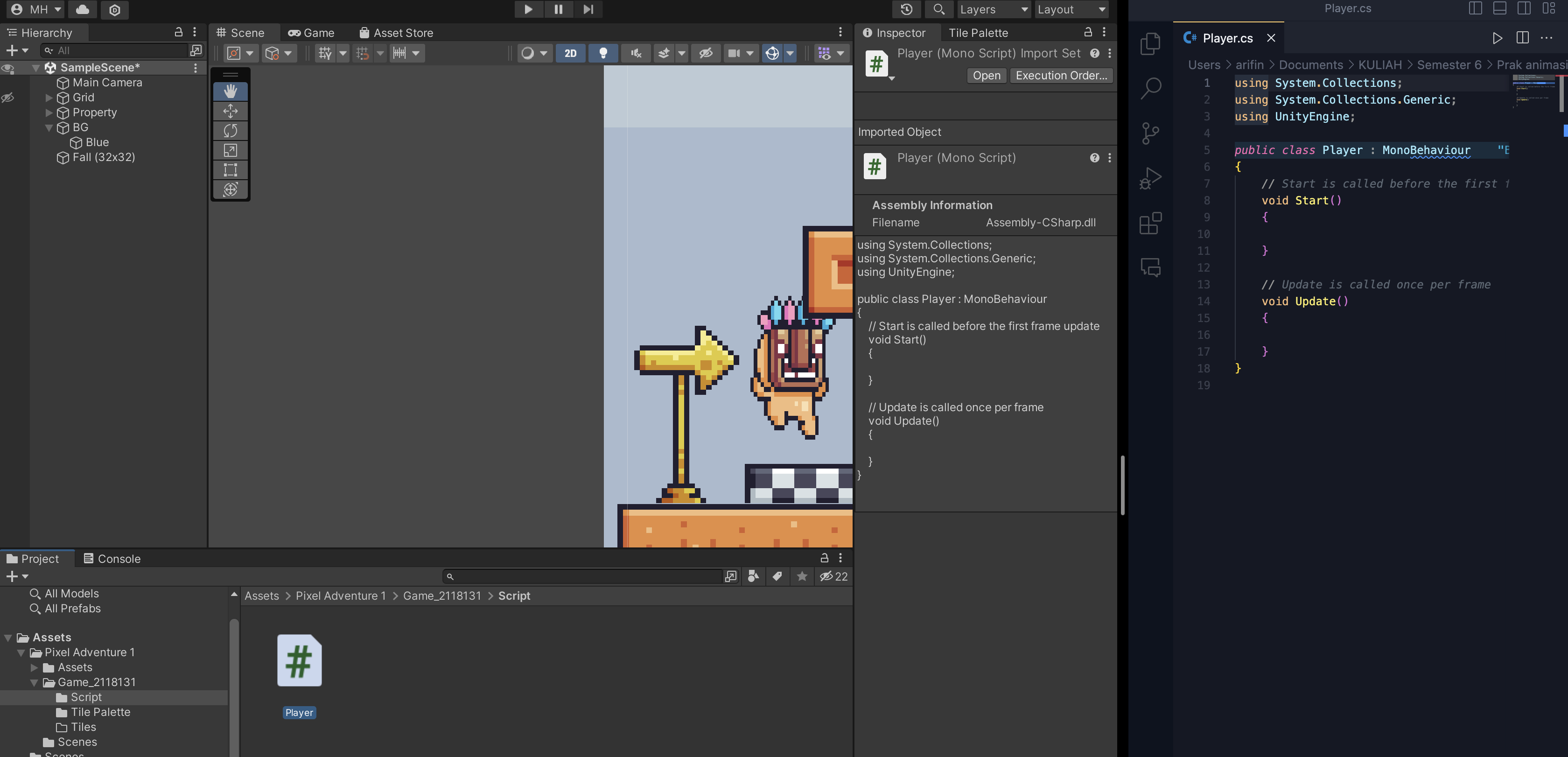
Gambar 8.6 Hasil Tampilan Creat Folder Script

1. Masuk kedalam folder Script, lalu buat C# Script, beri nama Player



Gambar 8.7 Hasil Tampilan Create C#

1. Drag & drop script player kedalam Hirarki Fall (32x32), lalu klik 2x pada script player maka akan masuk kedalam text editor seperti ini

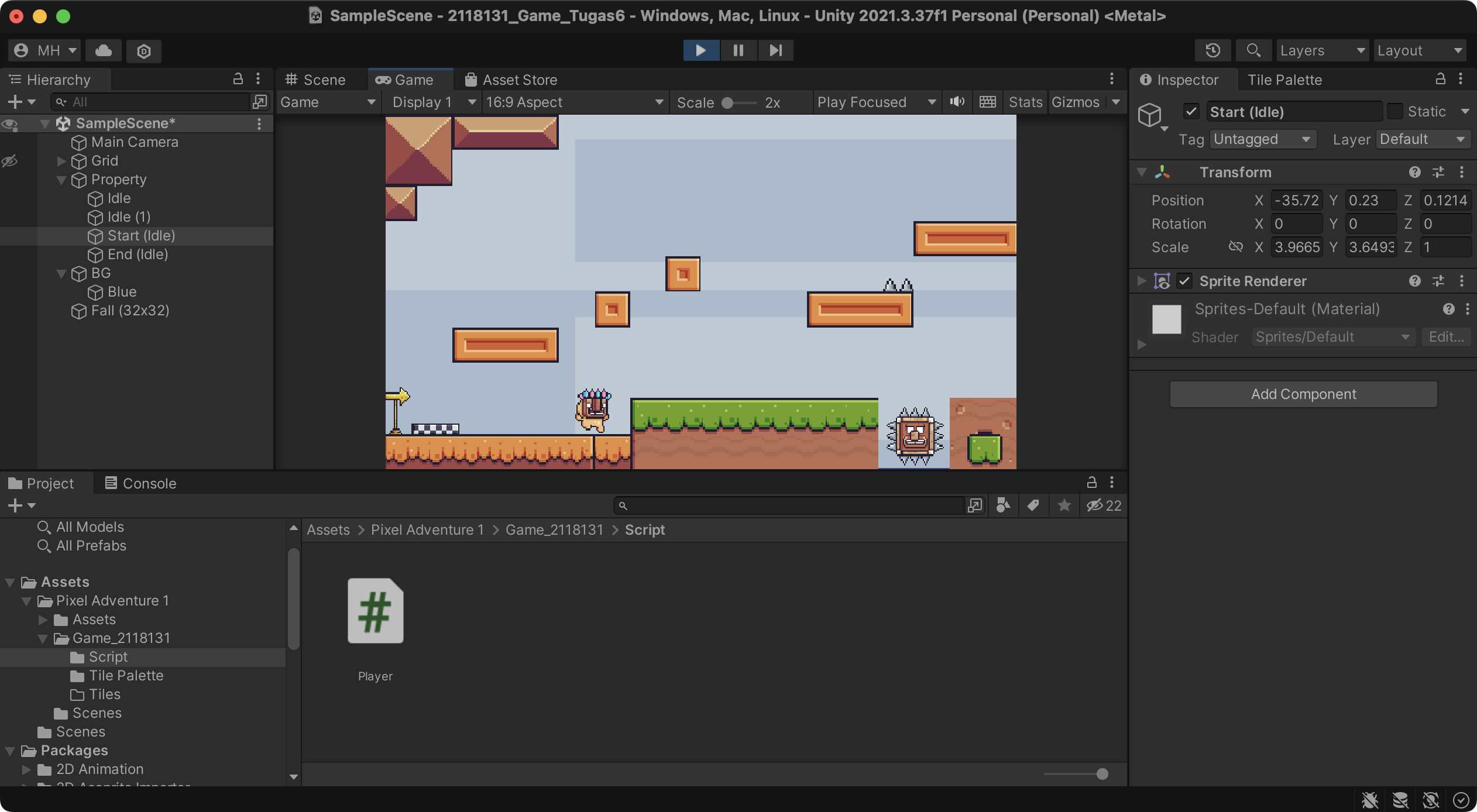


Gambar 8.8 Hasil Tampilan Drag & drop script

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

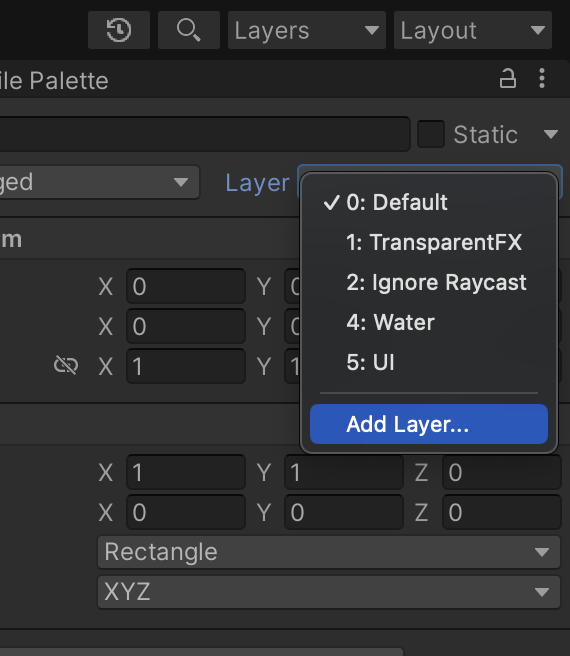
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Player : MonoBehaviour  {  Rigidbody2D rb;  [SerializeField] float speed = 1;  float horizontalValue;  bool facingRight;  private void Awake()  {  rb = GetComponent<Rigidbody2D>();  }  void Update ()  {  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 mencoba Source code diatas berhasil, Tekan dikeyboard “a” atau “left arrow” untuk ke arah kiri, tekan “d” atau “right arrow” untuk ke arah kanan



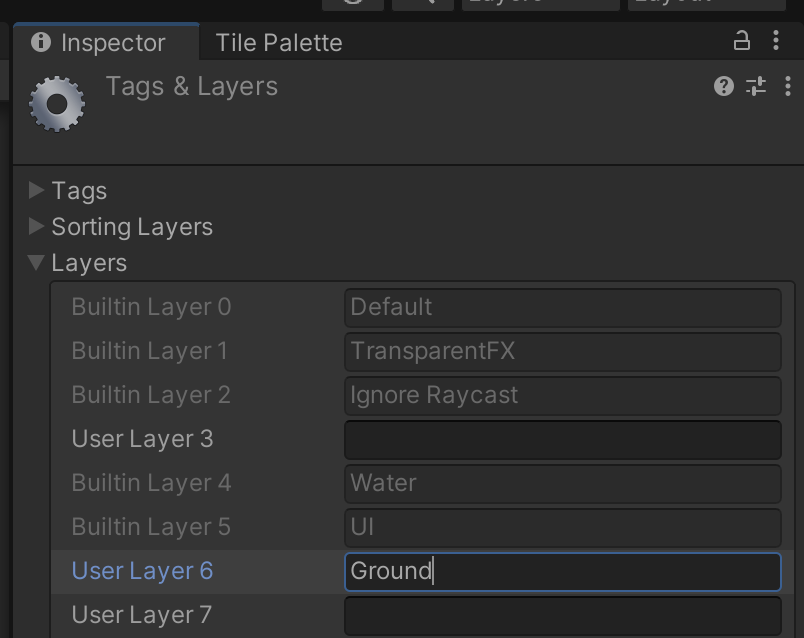
Gambar 8.9 Hasil Tampilan Hasil Mencoba Source code

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



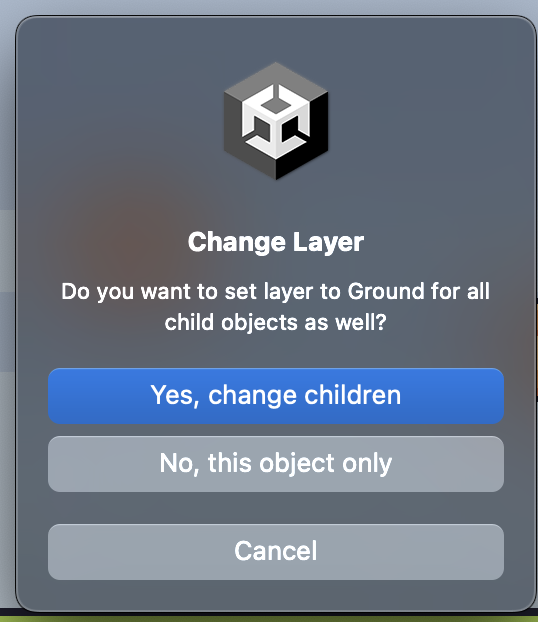
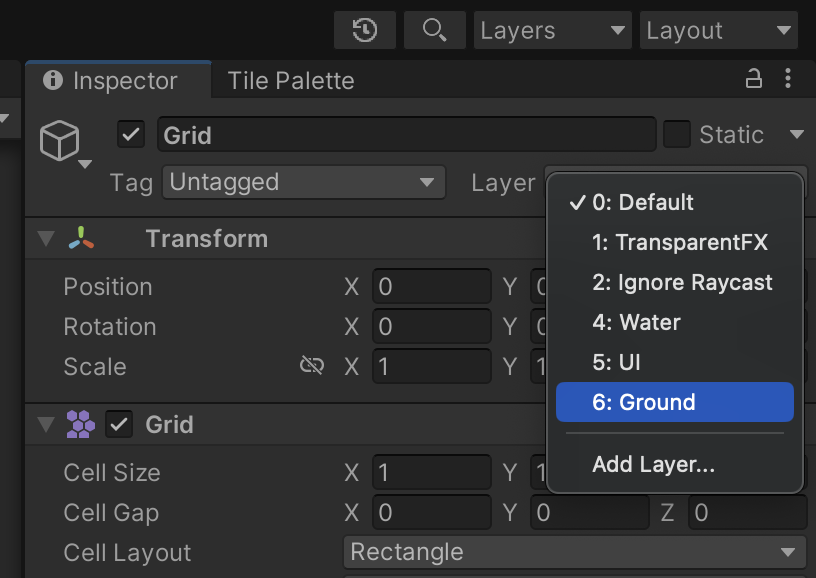
Gambar 8.10 Hasil Tampilan Creat Layer

1. Lalu isi “Ground” pada User Layer 6



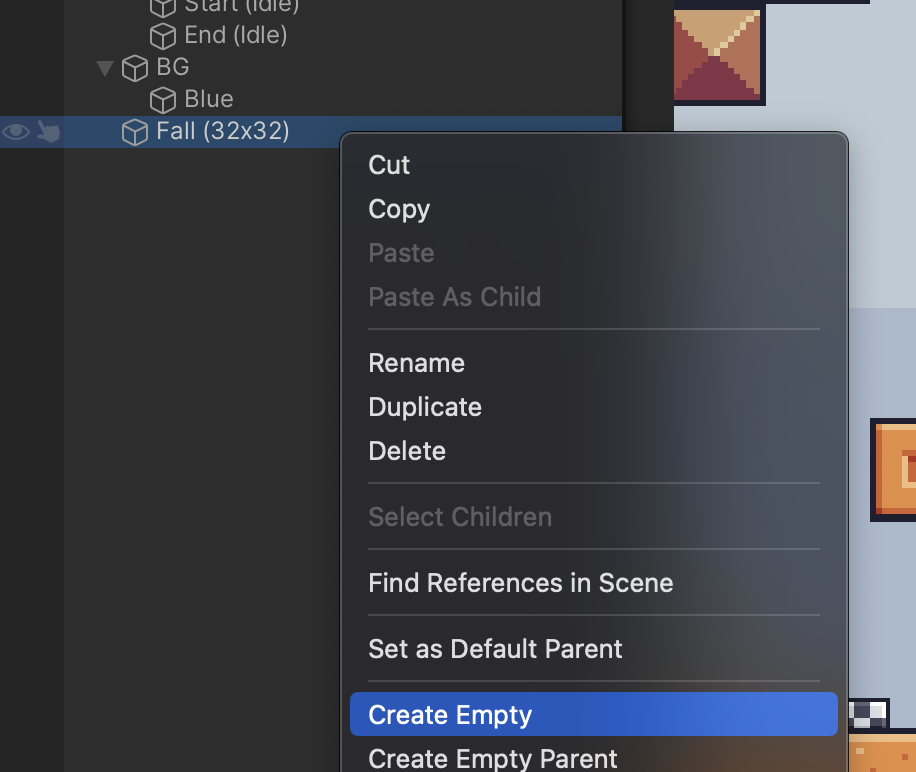
Gambar 8.11 Hasil Tampilan Name Layer

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



Gambar 8.12 Hasil Tampilan Menerpkan Layer

1. Klik kanan pada Fall(32x32), lalu Create empty, beri nama GorundCheck



Gambar 8.13 Hasil Tampilan Creat Empaty

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



Gambar 8.14 Hasil Tampilan Setting Hirarki GorundCheck

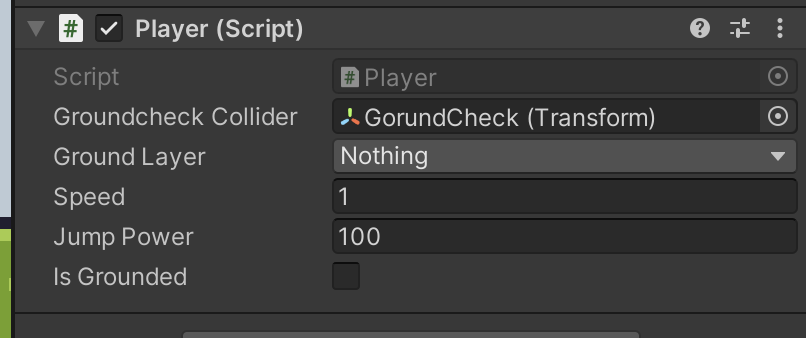
1. Kembali ke script Player tambahkan source code seperti ini

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

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

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

1. Klik fall(32x32), lalu ke inspector ke effect Player script di bagian “Goruncheck collider” tekan icon lalu pilih yang GorundCheck Transform, dan pada Ground Layer pilih Ground



Gambar 8.15 Hasil Tampilan Menempatkan GorundCheck

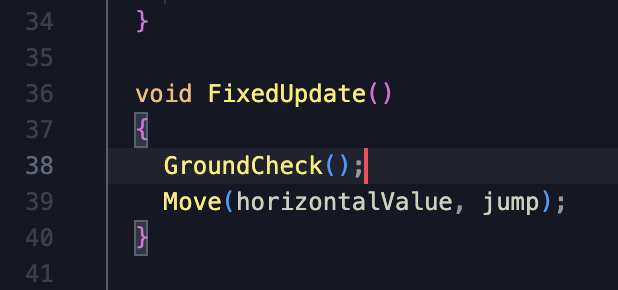
1. Lalu untuk membuat player melompat tambahkan script berikut

|  |
| --- |
| [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; |

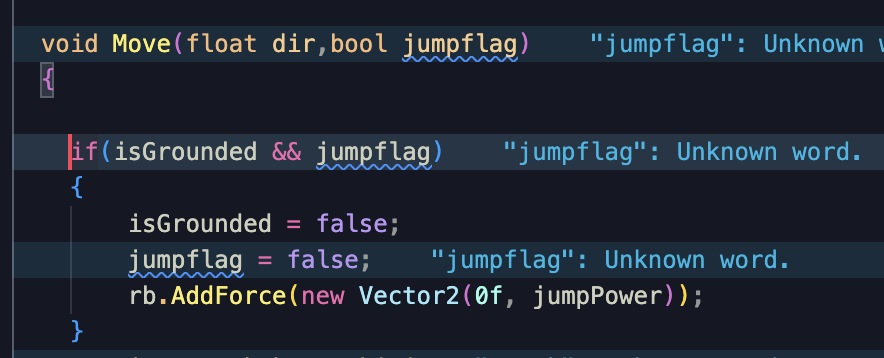
1. Tambahkan juga jump pada parameter Move



Gambar 8.16 Hasil Tampilan Create Parameter

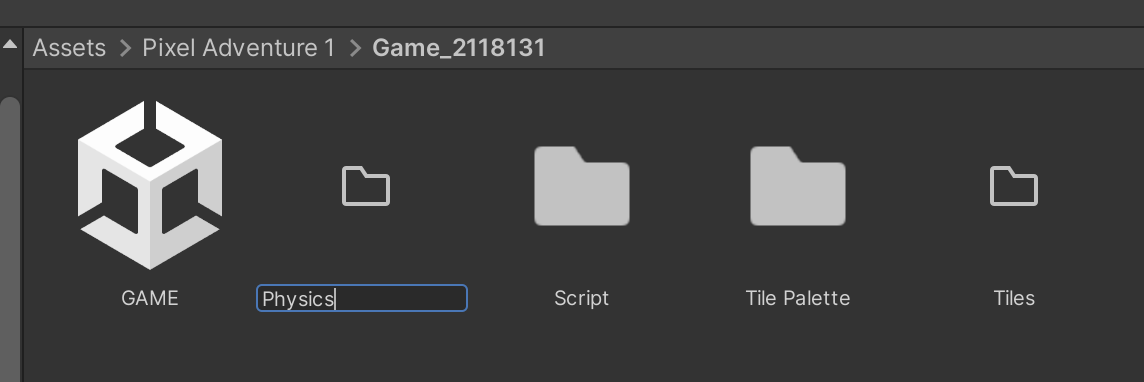
1. Tambahkan script berikut pada void Move

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



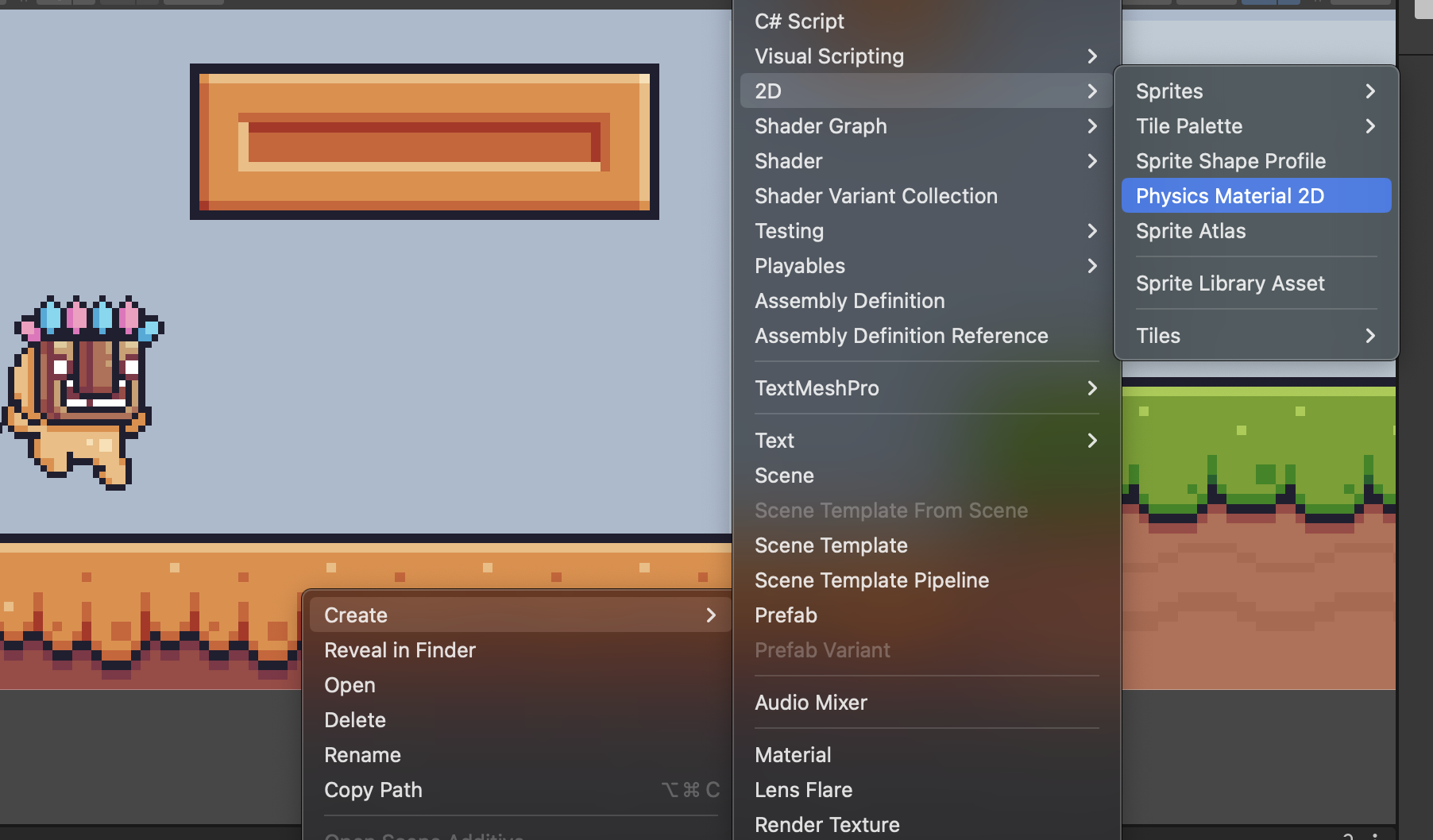
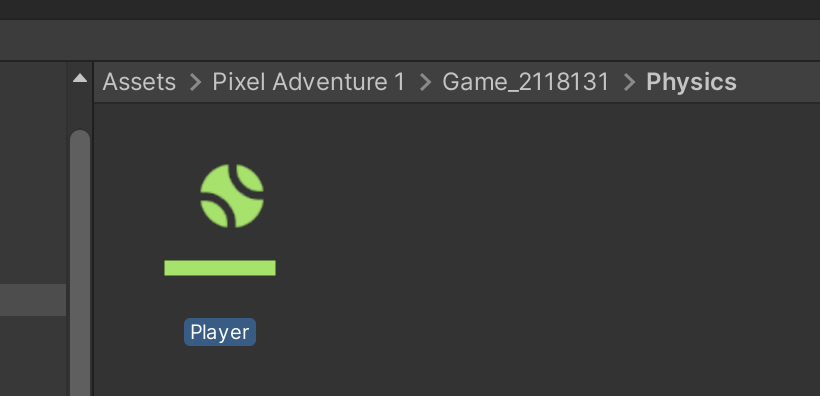
Gambar 8.17 Hasil Tampilan Rename Object

1. Buat folder baru di dalam Game\_2118131 bernama “Physics”



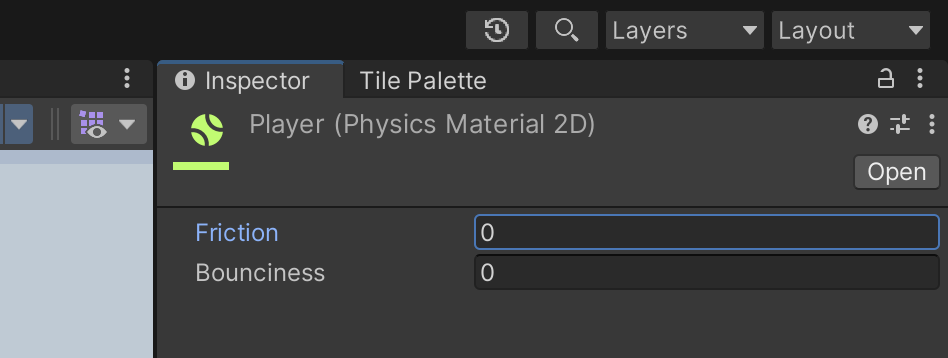
Gambar 8.18 Hasil Tampilan Create Folder

1. Didalam folder Pyshics create > 2d > physical material 2d , berinama “Player”

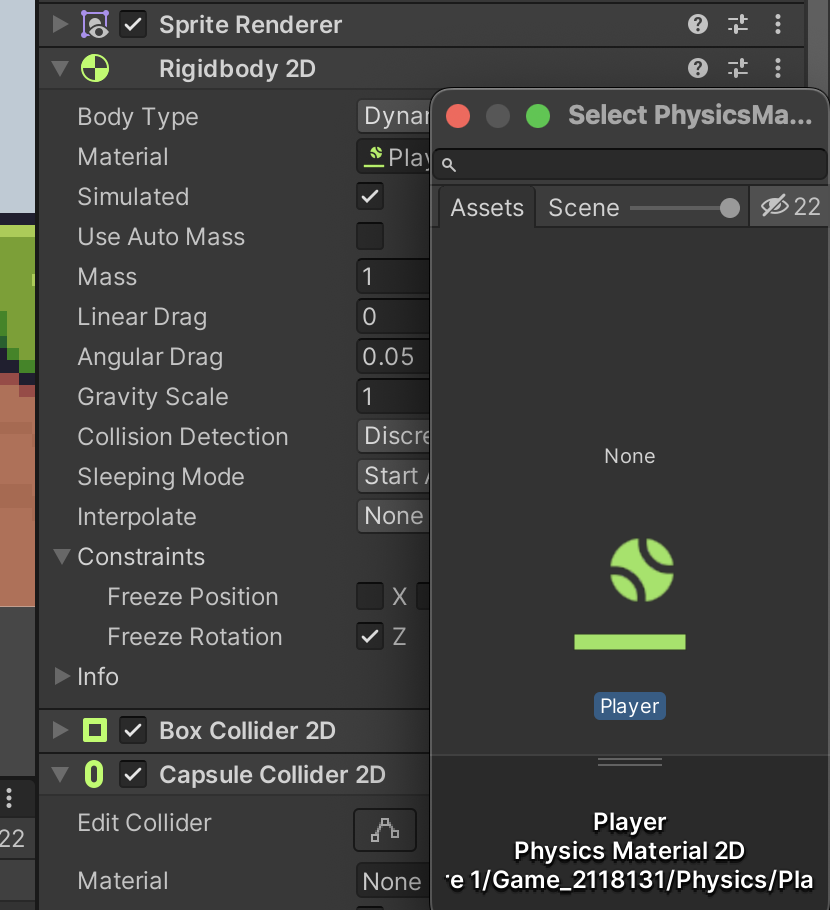
Gambar 8.19 Hasil Tampilan create 2d physical material 2d

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



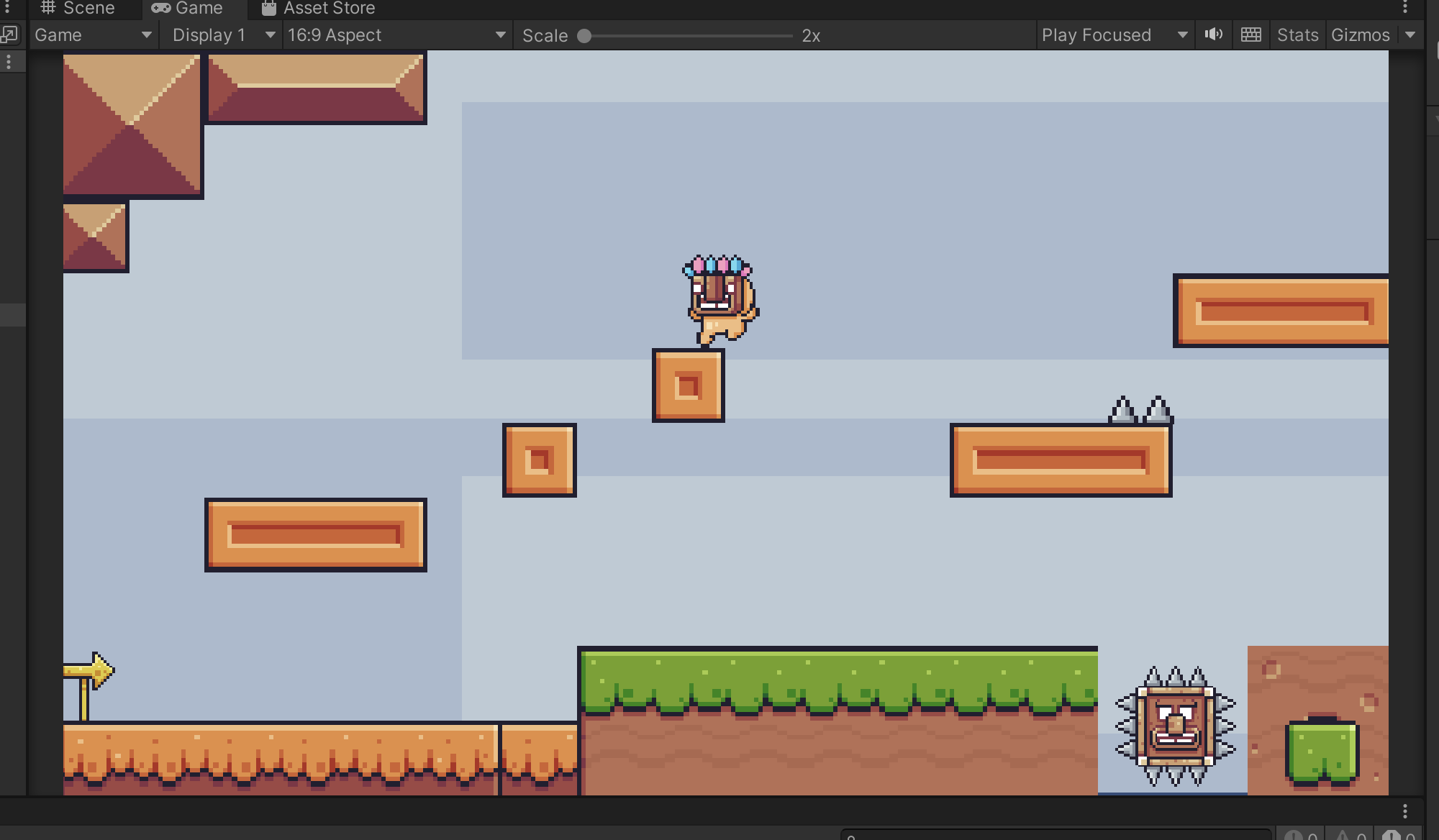
Gambar 8.20 Hasil Tampilan Physics Material 2D

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



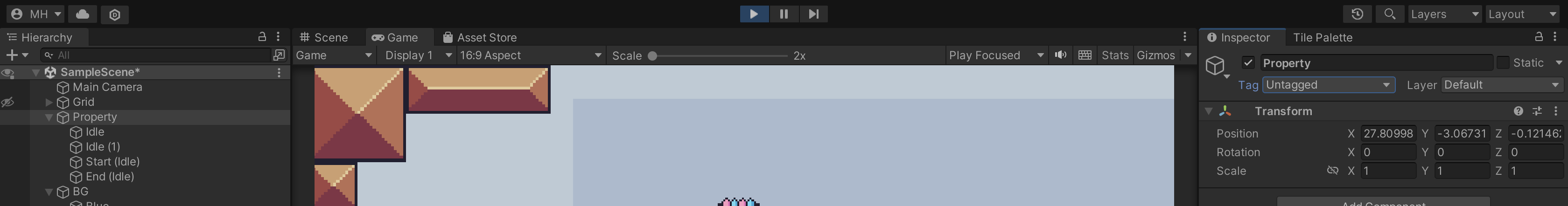
Gambar 8.21 Hasil Tampilan select physhics material 2d

1. Hasil maka akan serti di bawah ini



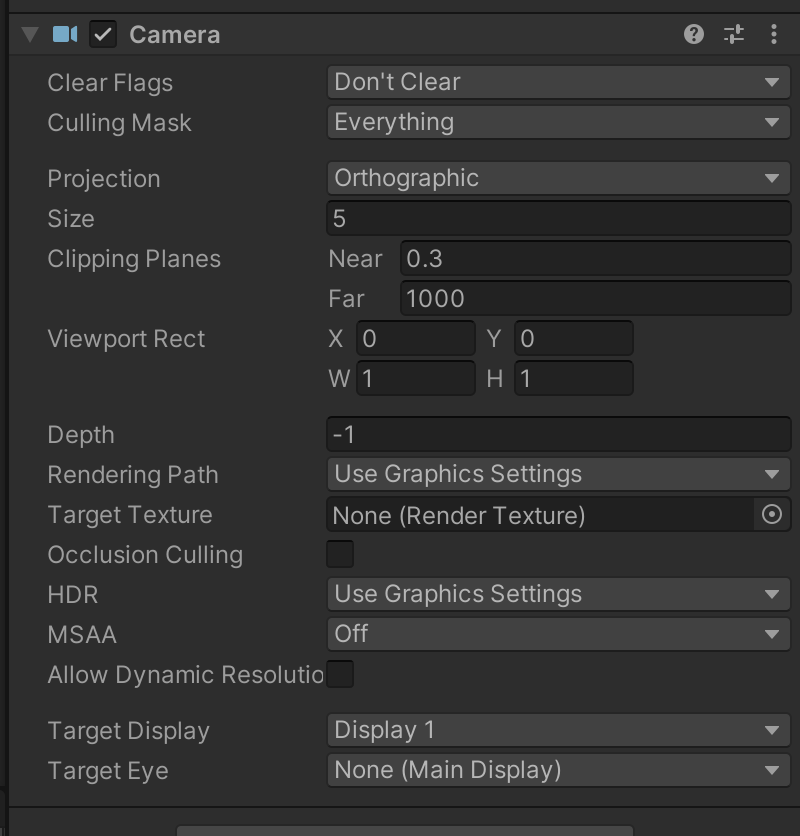
Gambar 8.22 Hasil Tampilan Jump

1. **Camera Movement**
2. Pada Hirarki Property Ubah Inspector pada tag Main camera Menjadi untagged



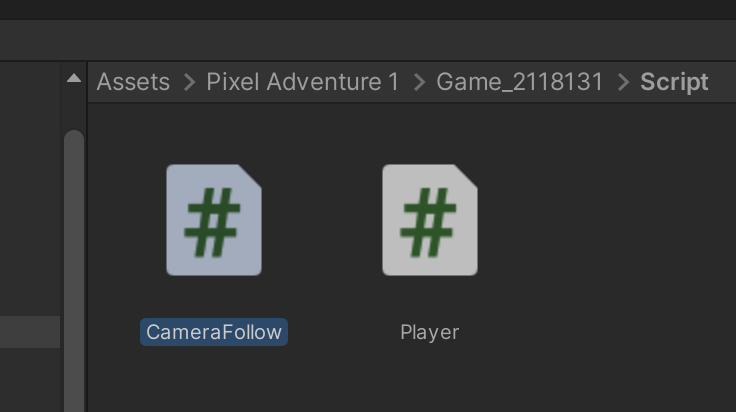
Gambar 8.23 Hasil Tampilan Crete Object

1. Sesuaikan Setting Layer Camera seperti gambar dibawah ini



Gambar 8.24 Hasil Tampilan Setting Layer Camera

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



Gambar 8.25 Hasil Tampilan Create C#

1. Lalu tuliskan script berikut ini

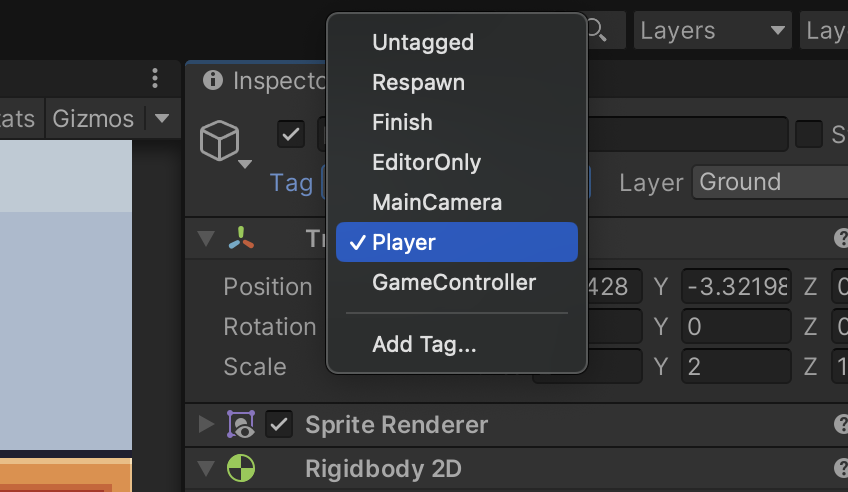
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class CameraFollow : MonoBehaviour  {  public float xMargin = 0.5f;  public float yMargin = 0.5f;  public float xSmooth = 4f;  public float ySmooth = 4f;  public Vector2 maxXAndY;  public Vector2 minXAndY;  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. Drag & drop script CameraFollow Kedalam Layer Main Camera, Lalu klik pada camera, buka inspector Pada bagian Camera Follow (Script) Ubah Bagian Max X dan Max Y nya



Gambar 8.26 Hasil Tampilan Drag & drop script

1. Ubah tag di Fall(32x32) Untagged menjadi ”Player”



Gambar 8.27 Hasil Tampilan Ubah Tag

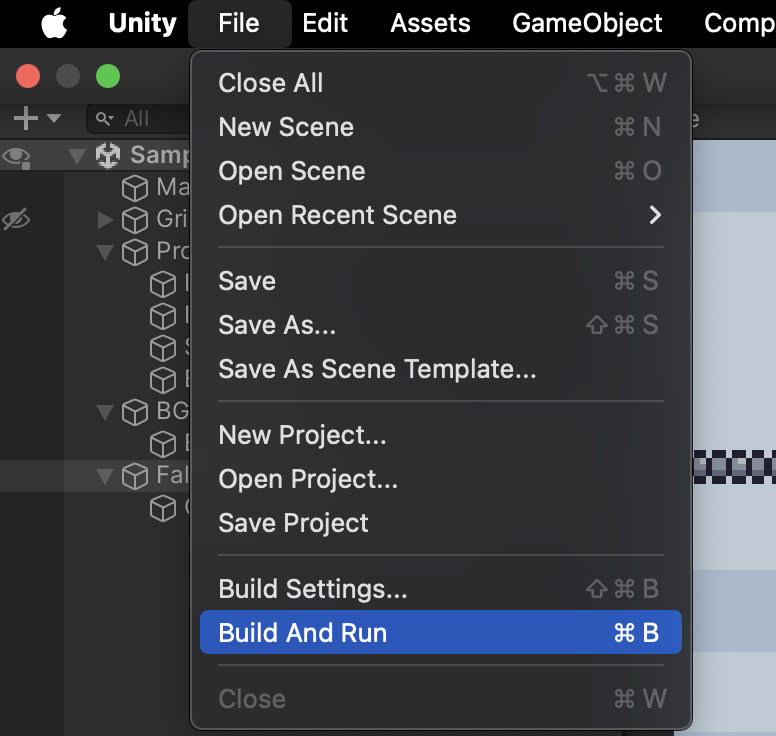
1. Tekan play untuk menjalankan, maka sekarang kamera akan mengikuti pergerakan karakter

A video game with a cartoon character

Description automatically generated

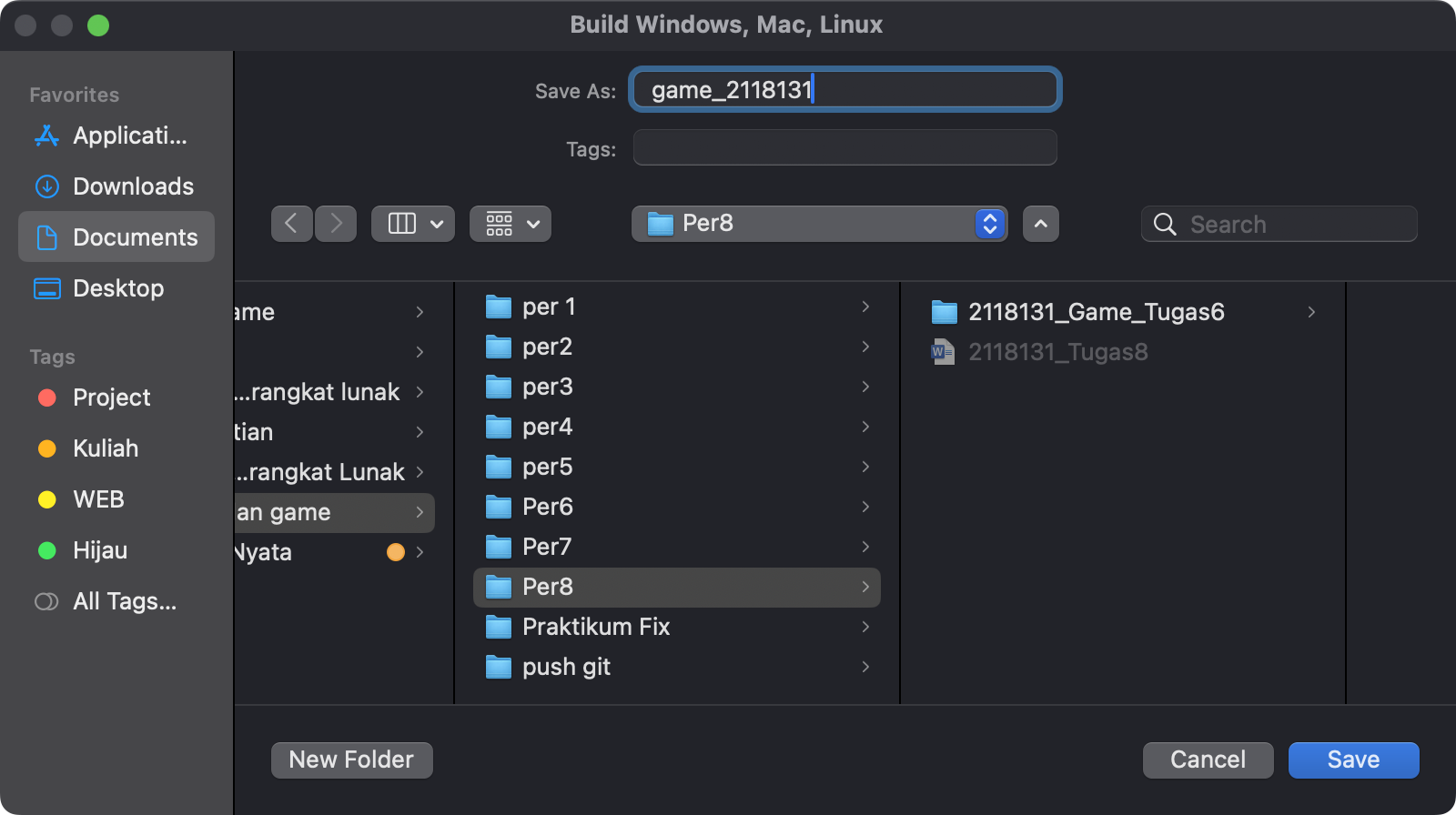
Gambar 8.28 Hasil Tampilan Play

1. **Render**
2. Pergi ke menu File kemudian pilih Build Setting (Ctrl(command) + Shift + B)



Gambar 8.29 Hasil Tampilan Pilih Build

1. Pergi ke menu File kemudian pilih Build Setting (Ctrl(command) + Shift + B)



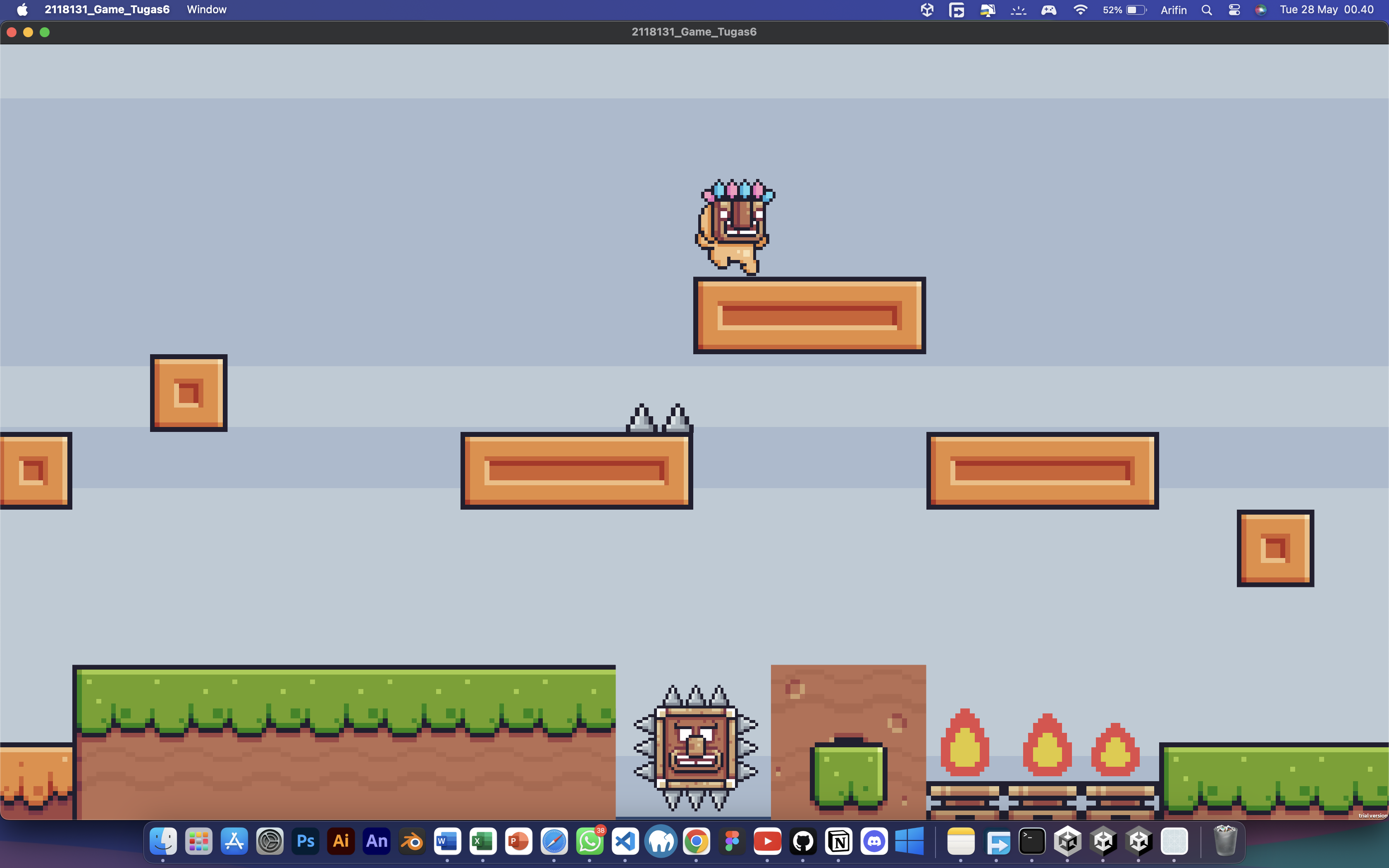
Gambar 8.30 Hasil Tampilan Almat Simpan

1. Tunggu



Gambar 8.31 Hasil Tampilan Prosses Render

1. Hasil



Gambar 8.32 Hasil Tampilan Render

1. **Link Ghithub**

<https://github.com/Arifin-MRX/2118131_Animasi_dan_Game.git>

## Tugas 2 : Menjelaskan Source Code Dibawah ini

|  |
| --- |
| 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 :

Pada source code di atas menggunakan script Unity C# yang membuat kamera hanya melacak posisi pemain pada sumbu x. Skrip ini mengimpor beberapa namespace yaitu System.Collections, System.Collections.Generic dan UnityEngine. Kelas CameraFollow berasal dari MonoBehaviour, yang merupakan kelas dasar untuk semua skrip Unity. Variabel jenis konversi Pemain diberi label [SerializeField], yang memungkinkan Anda menetapkan nilai dari editor Unity. Dalam metode pembaruan, posisi x kamera selalu ditetapkan pada posisi x pemain, sedangkan posisi y dan z tidak berubah. Hasilnya, kamera mengikuti pemain secara horizontal..