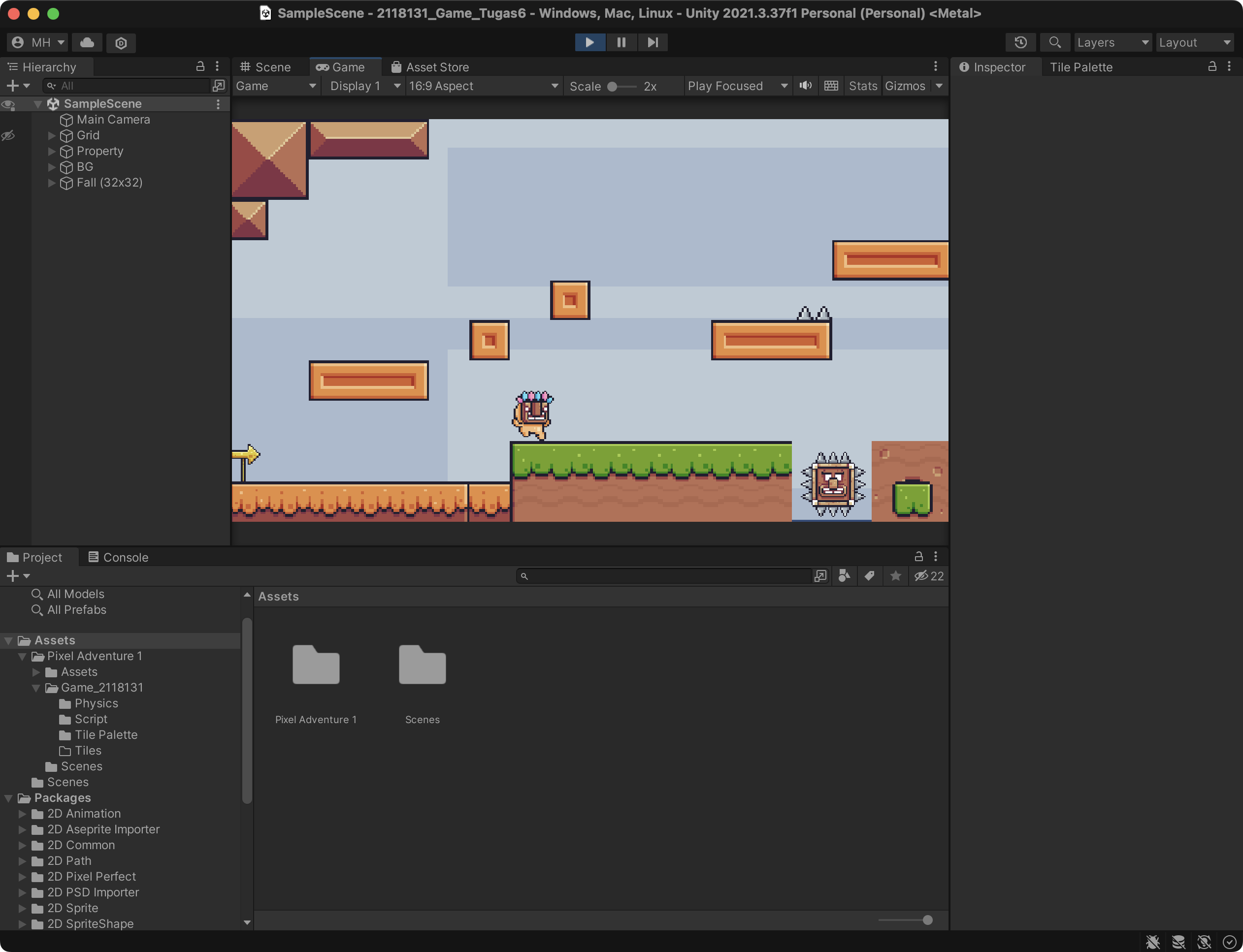
# 9 Game Animation

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

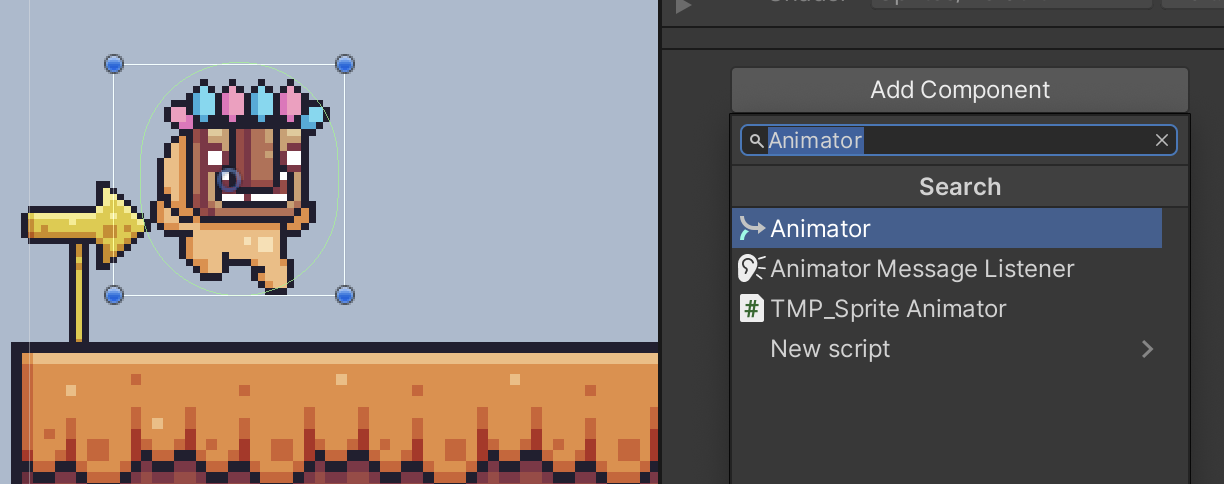
## Tugas 1 : Character Animation

1. **Character Animation**
2. Buka file projek Unity sebelumnya pada bab 8 untuk digunakan kembali



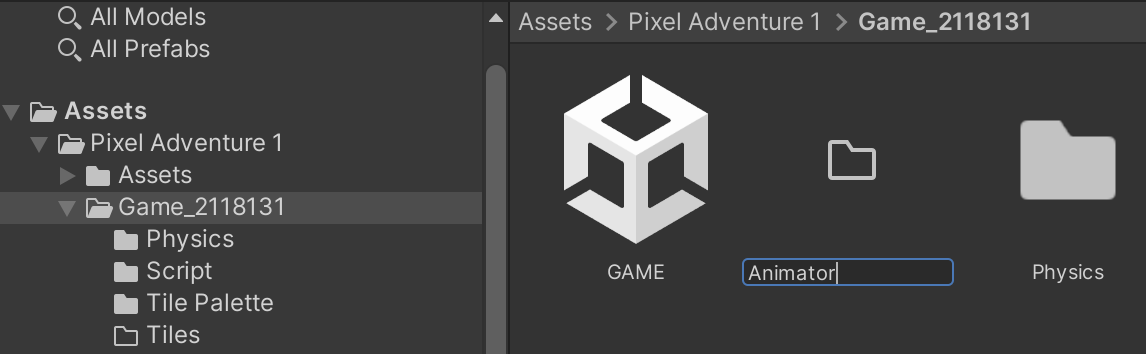
Gambar 9.1 Hasil Tampilan project Sebelumnya

1. Pada karakter klik inspector kemudian pilih Add Component Animator



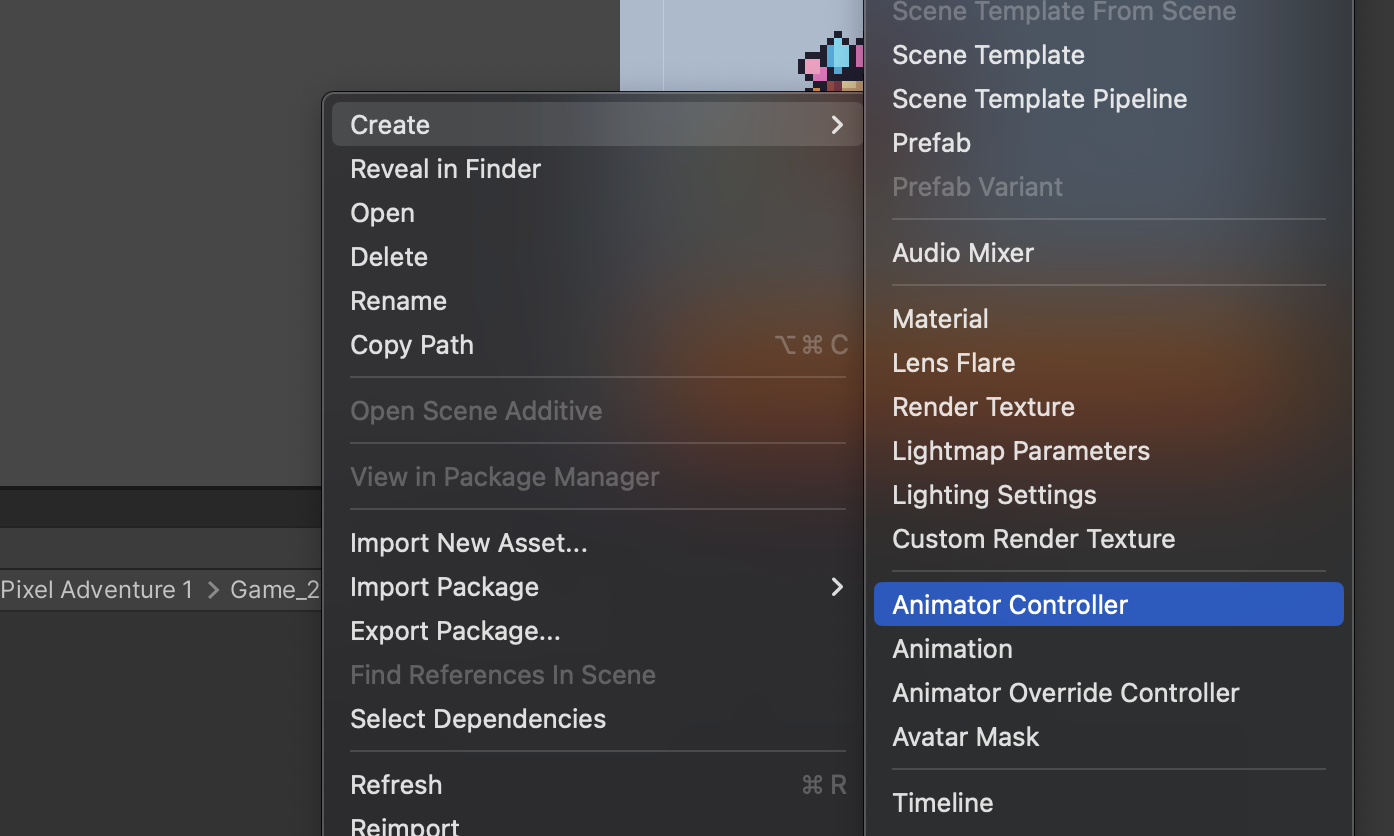
Gambar 9.2 Hasil Tampilan Creat Component

1. Pada folder Game\_2118131 Buat Folder baru “Animator”



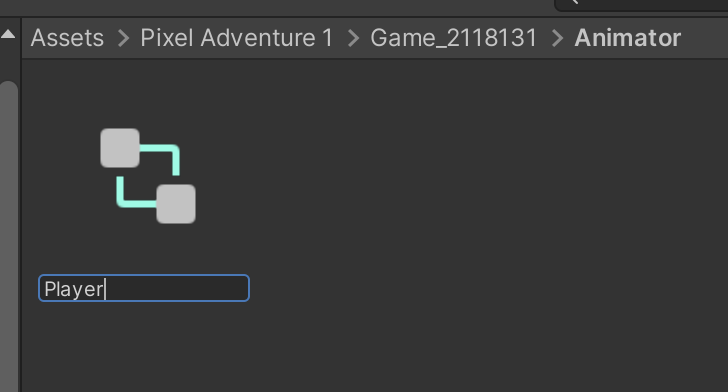
Gambar 9.3 Hasil Tampilan Create Folder

1. Buat File Animator Controller pada folder Animator



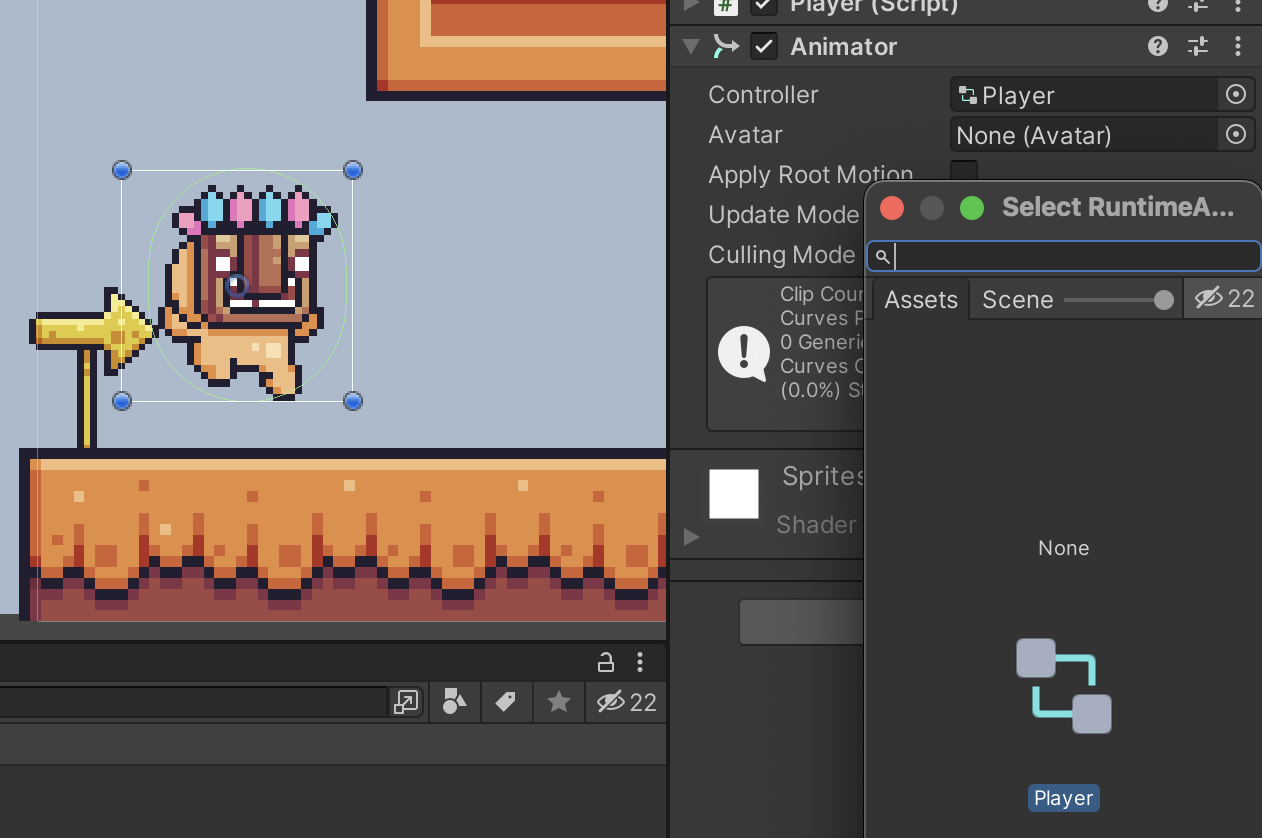
Gambar 9.4 Hasil Tampilan Creat Animator Controller

1. Selanjutnya ubah namanya menjadi Player



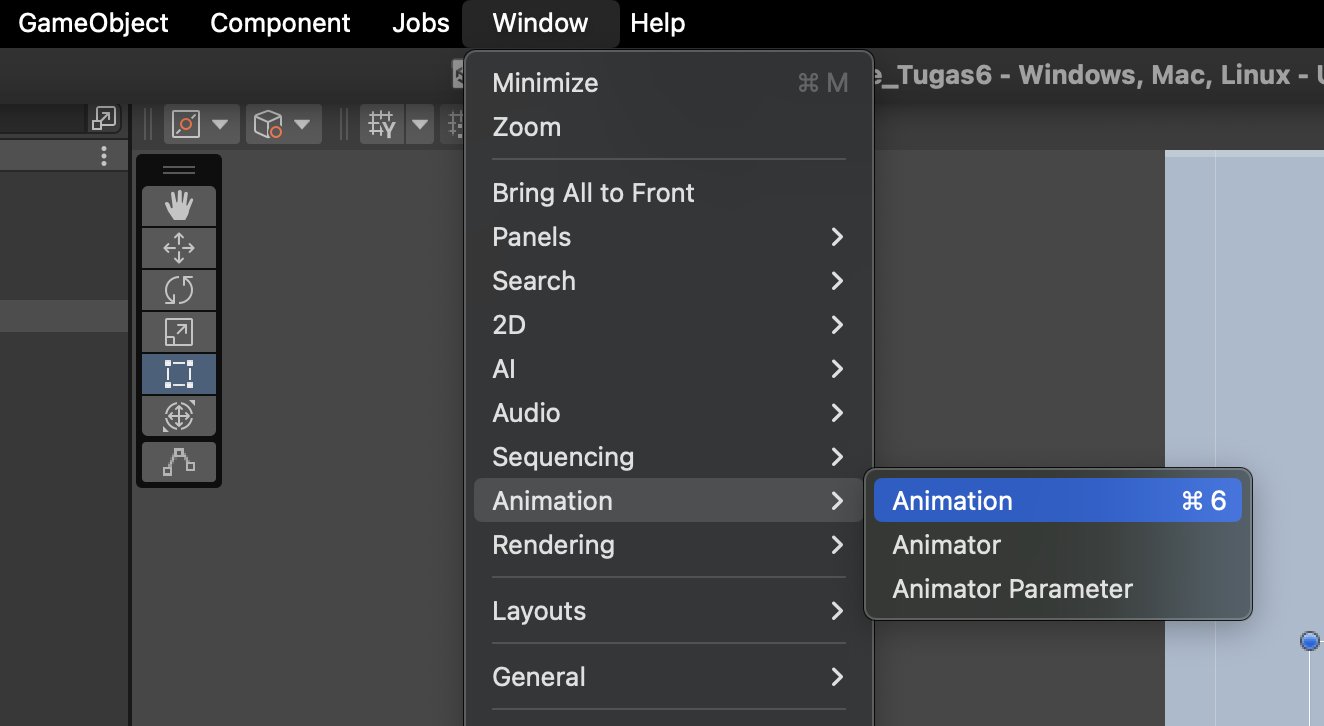
Gambar 9.5 Hasil Tampilan Name Animator Controller

1. Klik player pada Hierarchy, kemudian cari Component Animator, pada setting Controller ubah menjadi Player



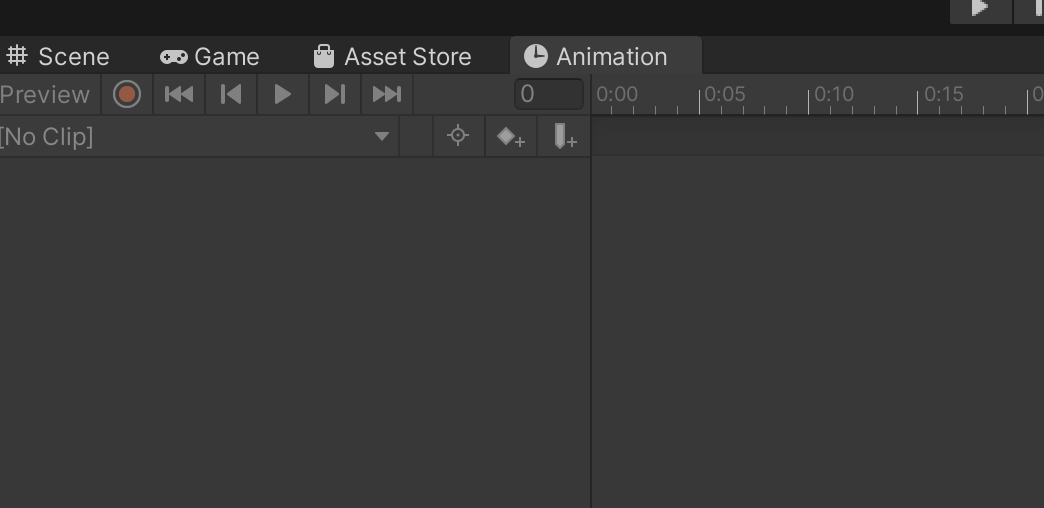
Gambar 9.6 Hasil Tampilan Select Controller

1. Tambahkan menu panel Animation di menu Window, pilih Animation > Animation atau tekan Shorcut CTRL(Comand) + 6



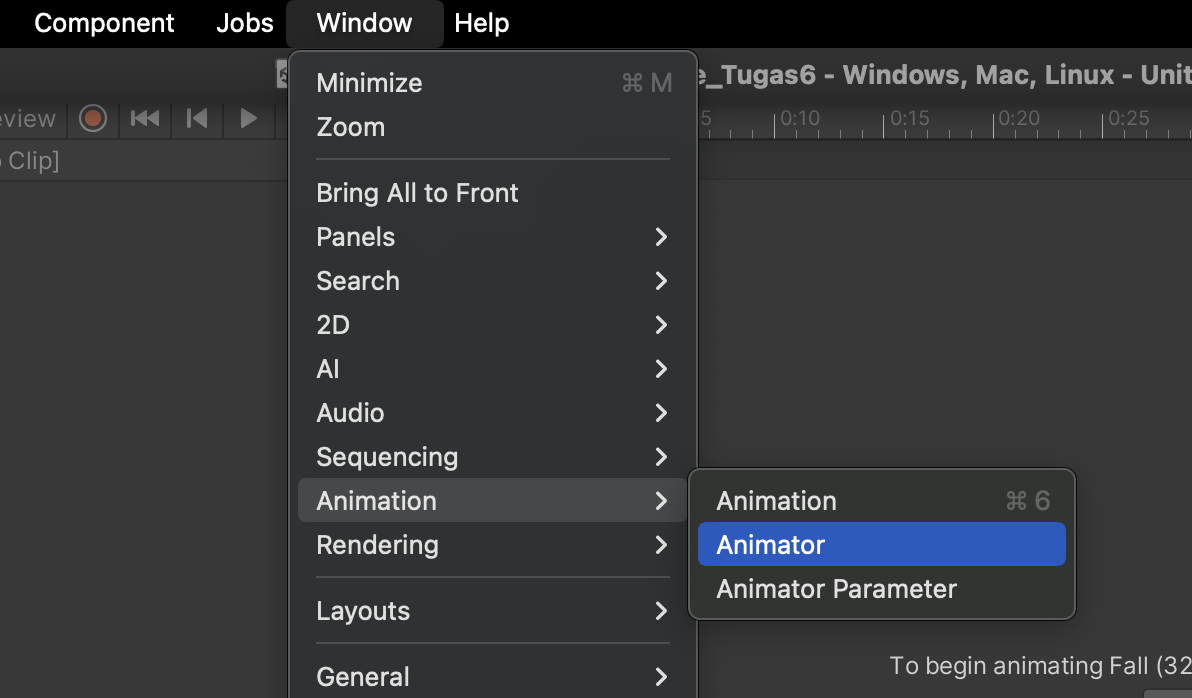
Gambar 9.7 Hasil Tampilan Menambahkan Window Animation

1. Akan muncul menu panel baru, geser panel tersebut sesuai dengan kenyamana masing-masing.



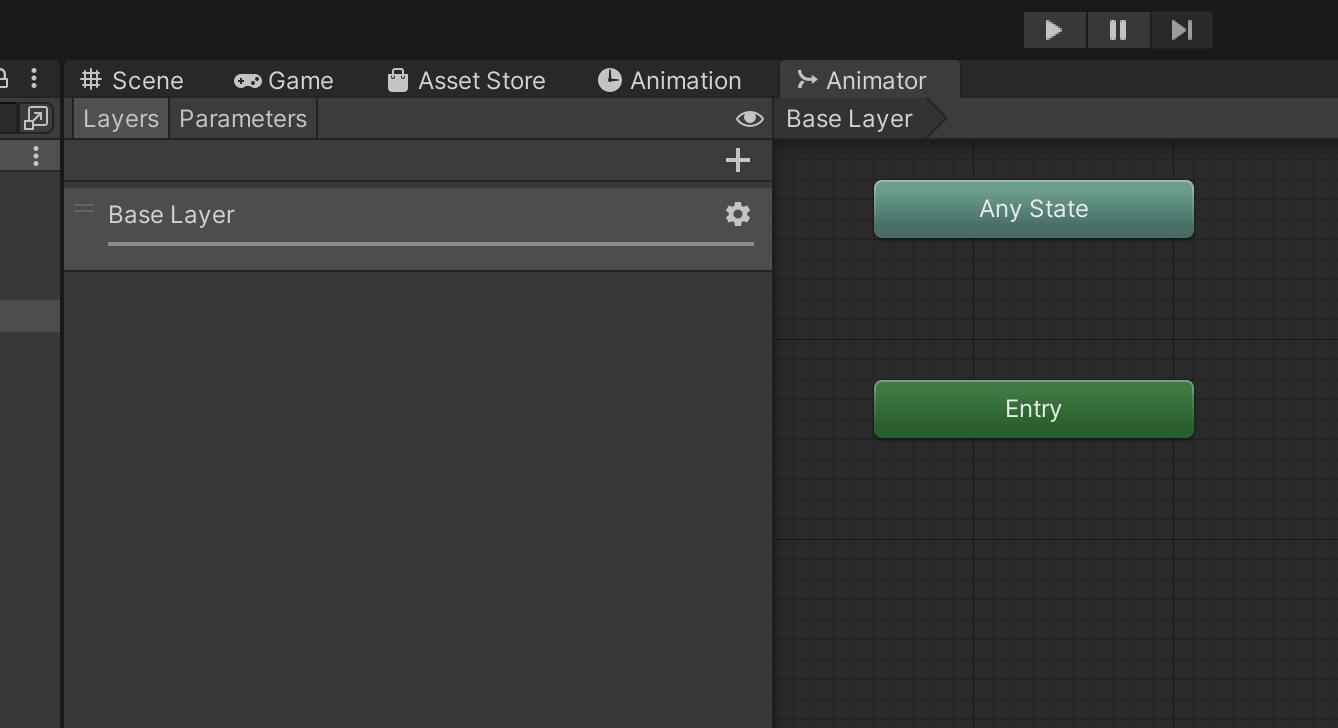
Gambar 9.8 Hasil Tampilan Windown Animation

1. Tambahkan menu panel Animator



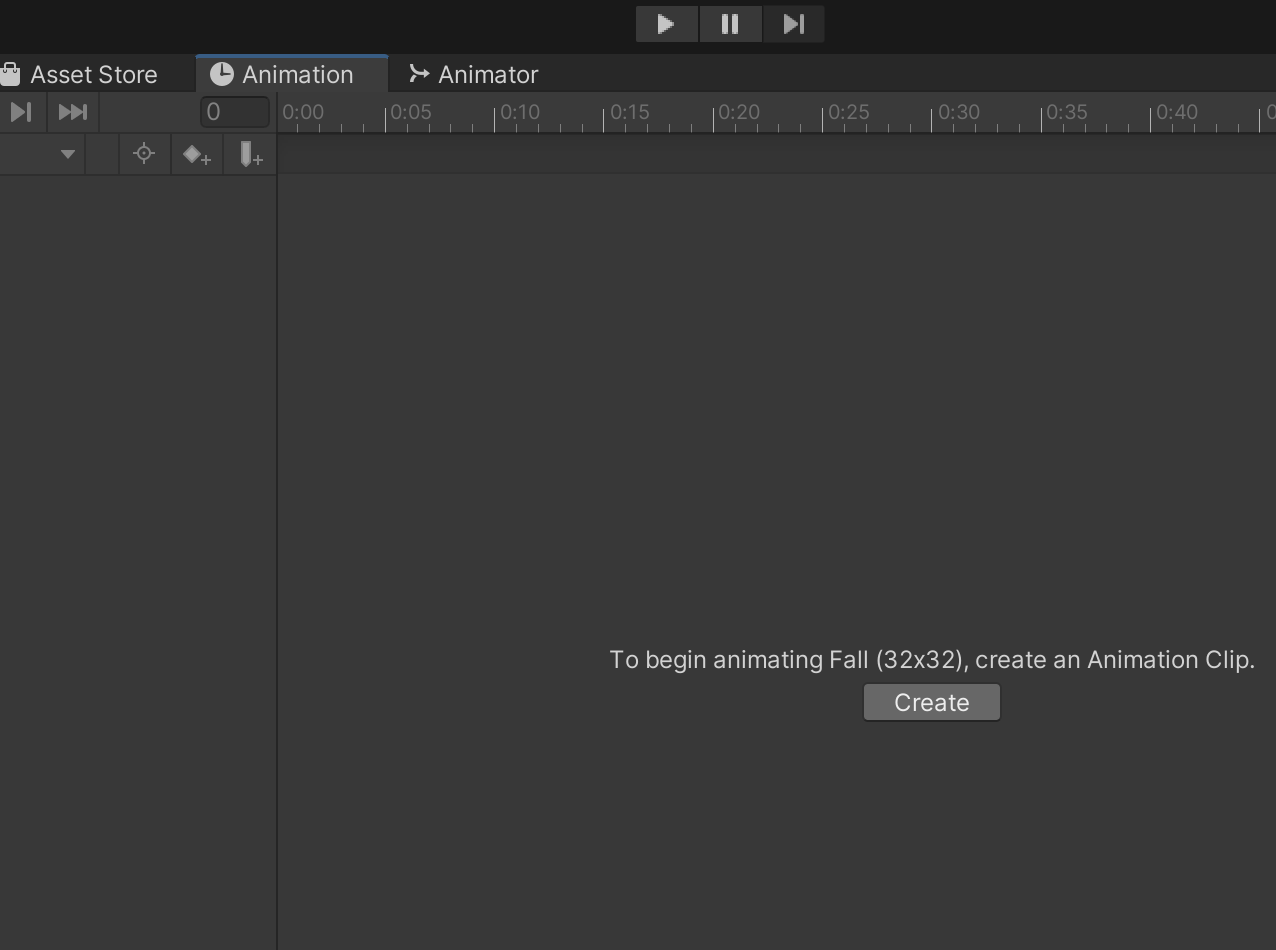
Gambar 9.9 Hasil Tampilan Create menu Animator

1. Geser Panel tersebut sesuai dengan gambar



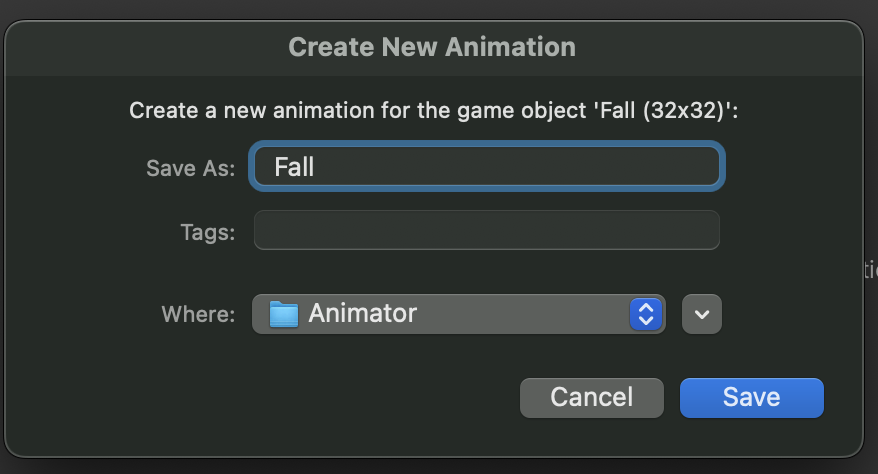
Gambar 9.10 Hasil Tampilan Window Animator

1. Untuk membuat animasi klik Fall pada Hierarchy, kemudian ke menu panel Animation, pilih Create



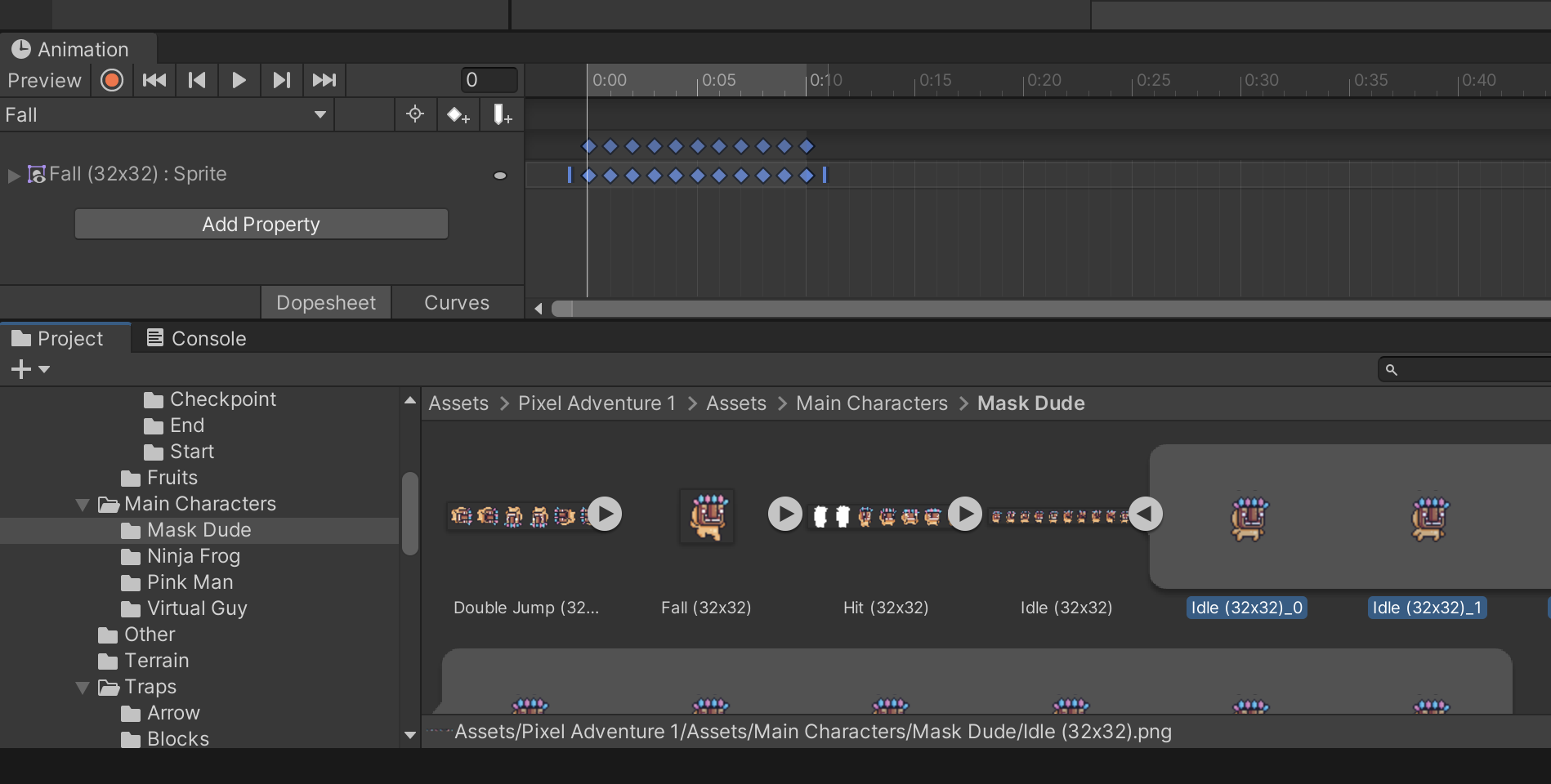
Gambar 9.11 Hasil Tampilan Create Animation Fall

1. Simpan pada folder Animator dan beri nama “Fall”



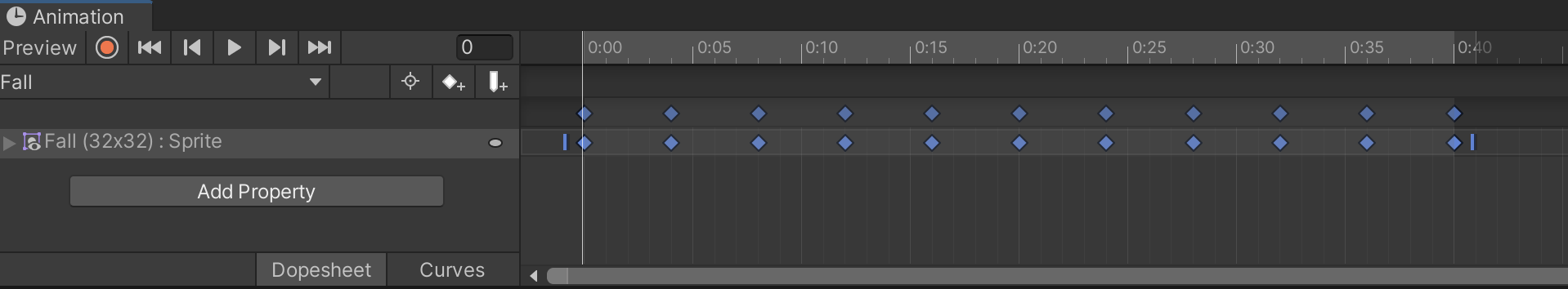
Gambar 9.12 Hasil Tampilan Simpan Animation

1. Pada menu Project buka folder main Character lalu pilih Maks dude dan pilih gambar idle(32x32), kemudian drag ke tab Animation



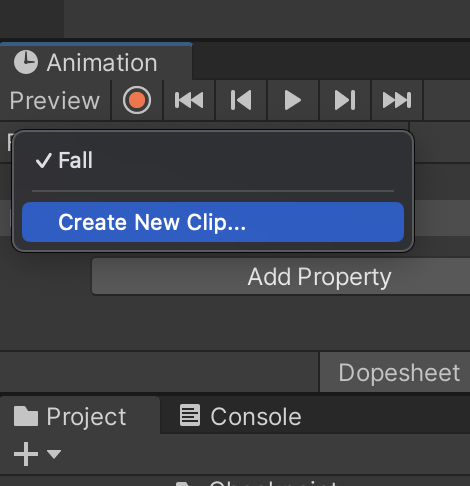
Gambar 9.13 Hasil Tampilan Creat Gambar Animation

1. Tekan CTRL (command) + A pada menu panel Animation geser kotak kecil pada timeline sampai frame 0:40 agar animasinya tidak terlalu cepat



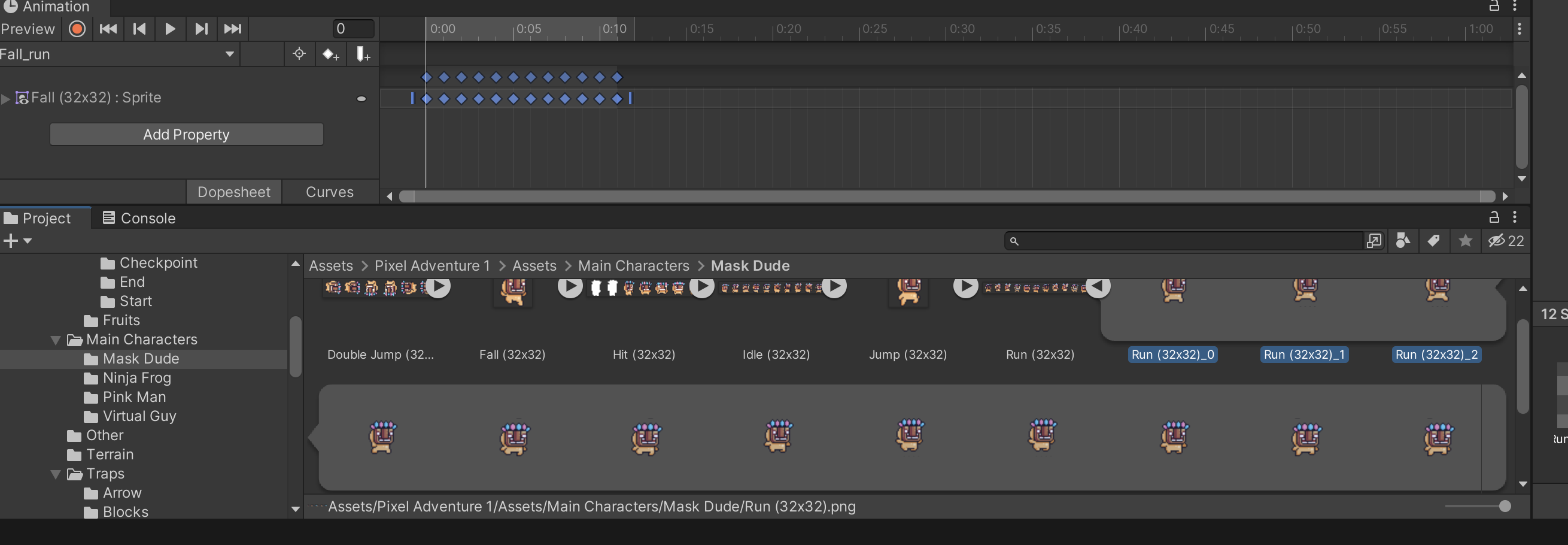
Gambar 9.14 Hasil Tampilan Setting Frame Animation

1. Buat animasi baru, Klik pada “Fall” kemudian pilih Create New Clip, dan beri nama “Fall\_run”, Simpan pada Folder Animator



Gambar 9.15 Hasil Tampilan Create Animation Baru

1. Buka menu Project kemudian cari folder Main character > maks dude, Pilih Run 1 hingga 11.



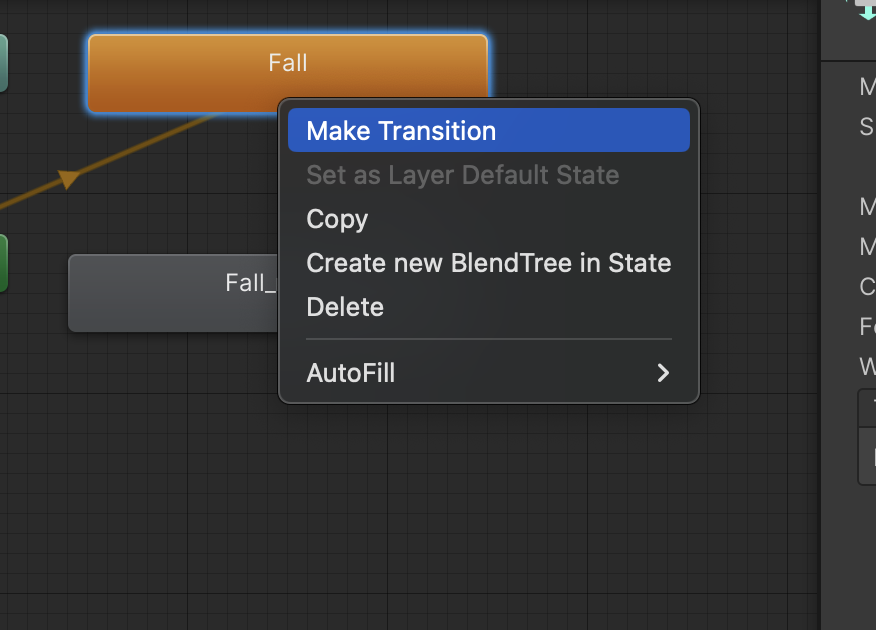
Gambar 9.16 Hasil Tampilan Create Gambar Run Fall

1. pada panel timeline tekan Ctrl (Command)+A di keyboard, klik bagian kotak kecil disamping keyframe terakhir dan geser sampai waktu 0:45



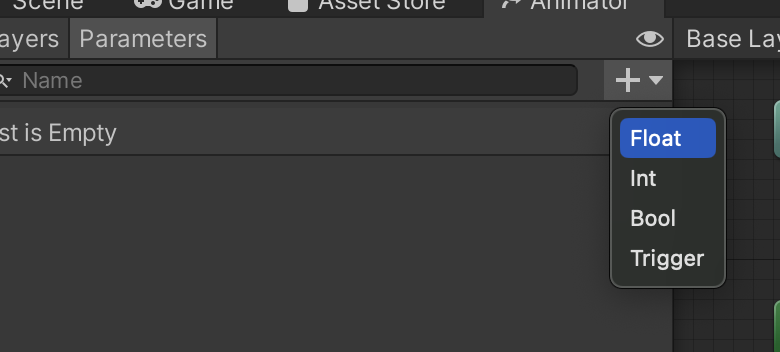
Gambar 9.17 Hasil Tampilan Rename Object

1. Kemudian buat transisi antara Fall dan Fall\_run dengan cara klik kanan pada Fall dan pilih Make Transitiondan tarik ke Fall\_run



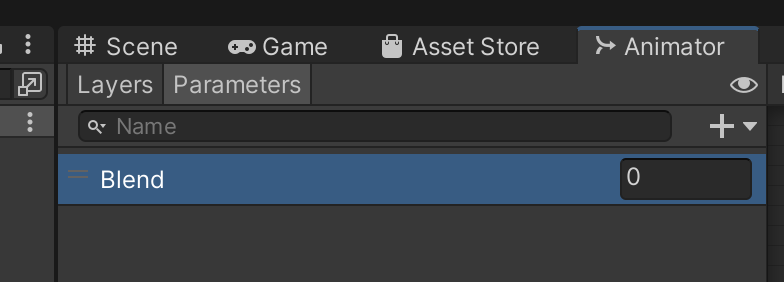
Gambar 9.18 Hasil Tampilan Create Make Transition

1. Masuk ke tab parameter, tambahkan tipe data bdengan cara tekan icon tambah dan ubah namanya menjadi “Blend”



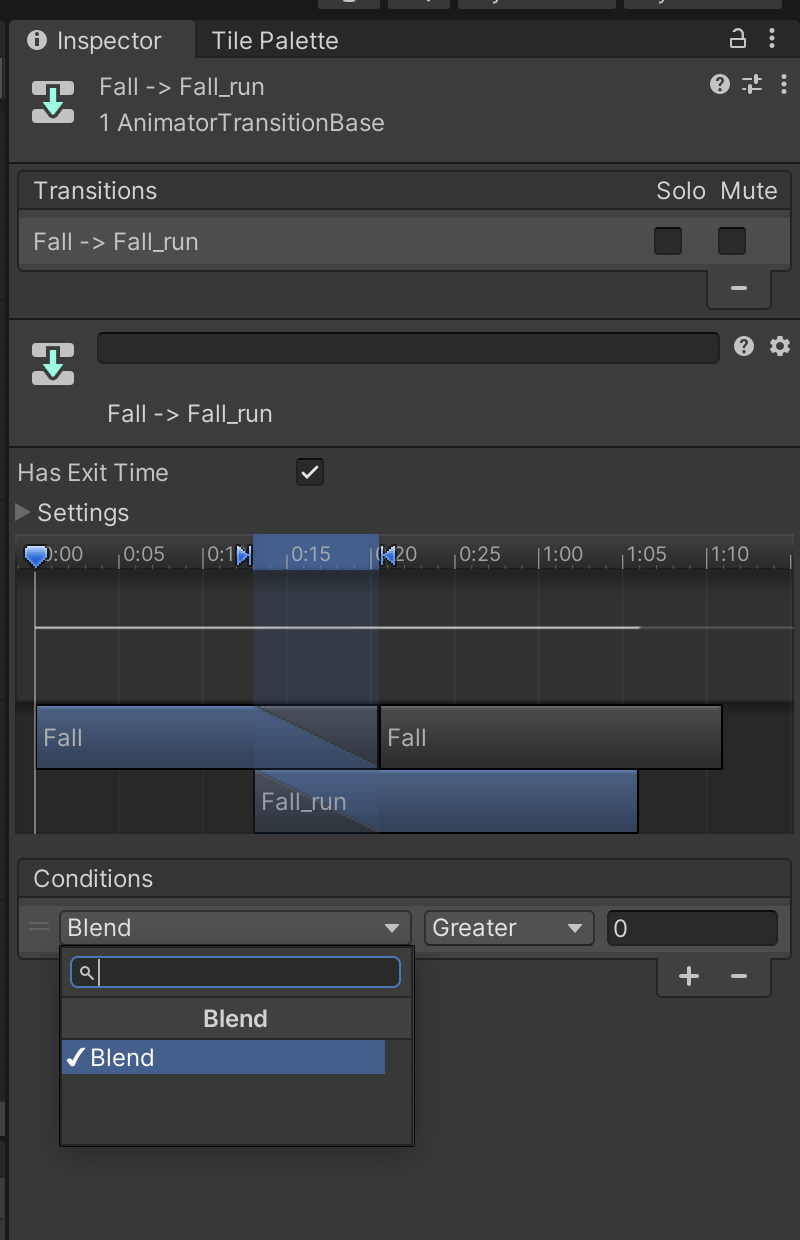
Gambar 9.19 Hasil Tampilan Create Paramaeters

1. Hasilnya akan seperti pada gambar



Gambar 9.20 Hasil Tampilan Parameters

1. Klik panah putih tersebut, pada bagian conditions klik icon tambah kemudian atur menjadi “Blend”.



Gambar 9.21 Hasil Tampilan Setting conditions

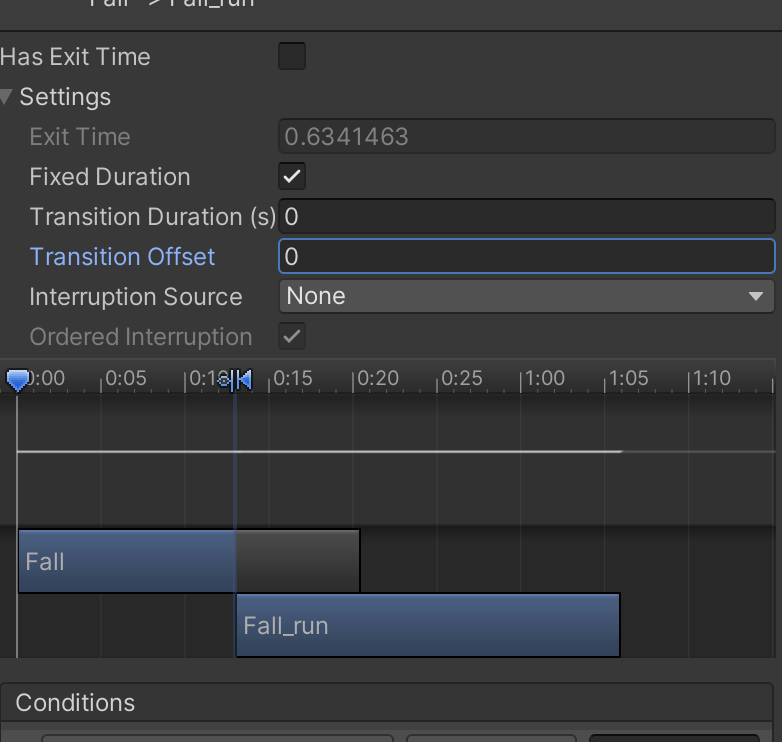
1. Atur nilai conditions blend tersebut menjadi 0.01

A screenshot of a computer

Description automatically generated

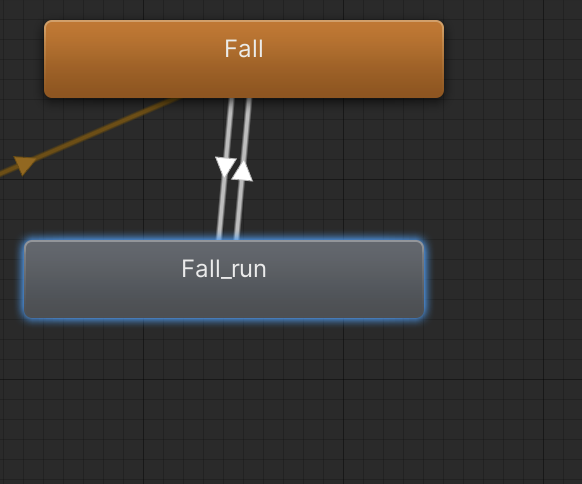
Gambar 9.22 Hasil Tampilan Value conditions

1. Pada bagian Settings, hilangkan centang pada Has Exit Time dan atur nilai Transition Duration menjadi 0



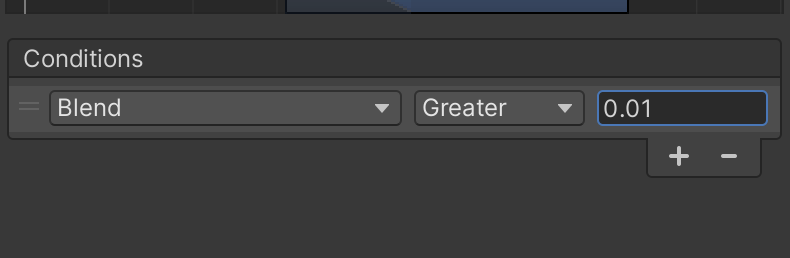
Gambar 9.23 Hasil Tampilan Setting Inspector

1. Buat transisi juga dari Fall\_run ke Fall dengan cara klik kanan pada Fall\_run dan pilih Make Transition.



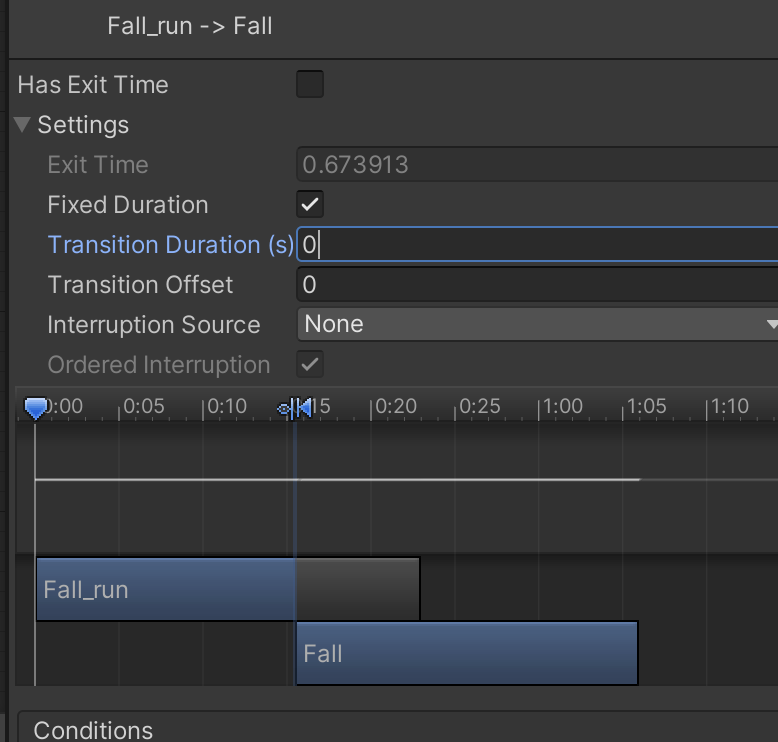
Gambar 9.24 Hasil Tampilan Make transition

1. Tambahkan parameter transisi dengan tipe data Float. Klik ikon tambah dan rename menjadi “Blend”. Setelah itu, ubah operator dari Greater menjadi Less dan atur nilainya menjadi 0.01.



Gambar 9.25 Hasil Tampilan Conditons

1. Pada bagian Settings, hilangkan centang pada Has Exit Time dan atur nilai Transition Duration menjadi 0



Gambar 9.26 Hasil Tampilan Setting Inspector

1. Agar animasi dapat sesuai ketika berjalan, buka script Player dan tambahkan source code berikut pada class Player.

|  |
| --- |
| public Animator animator; |

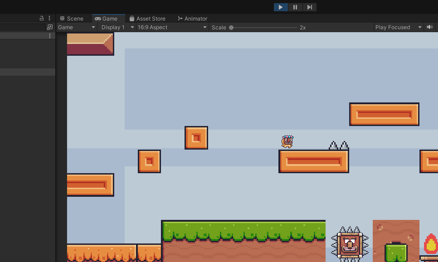
1. Tambahkan Script Komponen Animator di dalam method Awake()

|  |
| --- |
| animator = GetComponent<Animator>(); |

1. Dan pada fungsi FixedUpdate tambahkan source code berikut

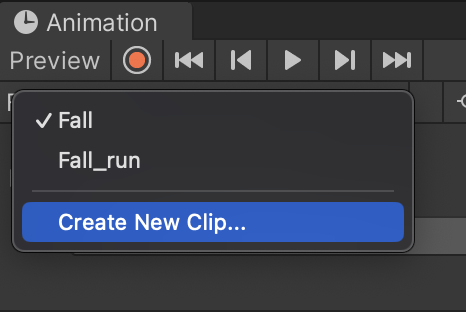
|  |
| --- |
| animator.SetFloat("Blend", Mathf.Abs(rb.velocity.x)); |

1. Jika dijalankan maka player dapat memiliki animasi ketika berhenti ataupun ketika berjalan



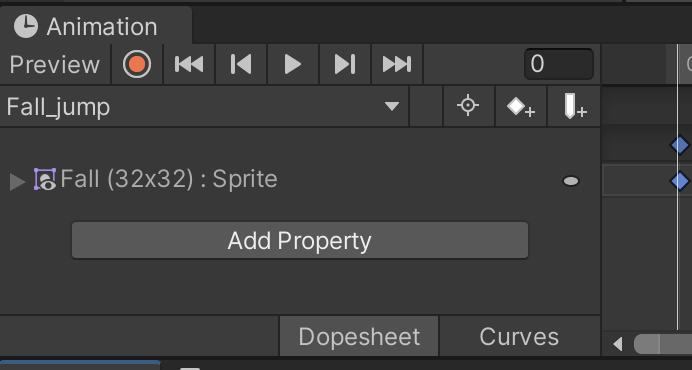
Gambar 9.27 Hasil Tampilan Menambahkan Animation

1. Kemudian buat animasi baru tekan tulisan “Fall\_run” kemudian pilih Create New Clip, dan beri nama “Fall\_jump”



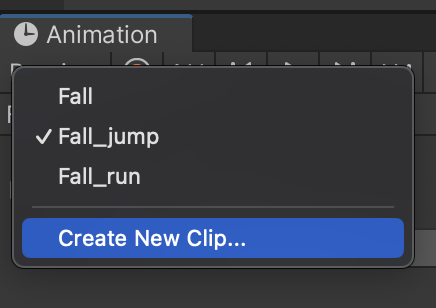
Gambar 9.28 Hasil Tampilan Create New Animation

1. Pada folder Main Character buka maks dude lalu pilih gambar wall jump(32x32), kemudian drag ke tab Animation.



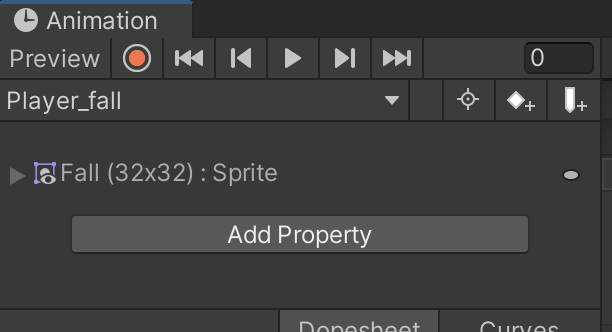
Gambar 9.29 Hasil Tampilan Drag Gambar

1. Buat animasi baru dengan cara tekan tulisan “Fall\_jump" kemudian pilihCreate New Clip, dan beri nama “Player\_fall”



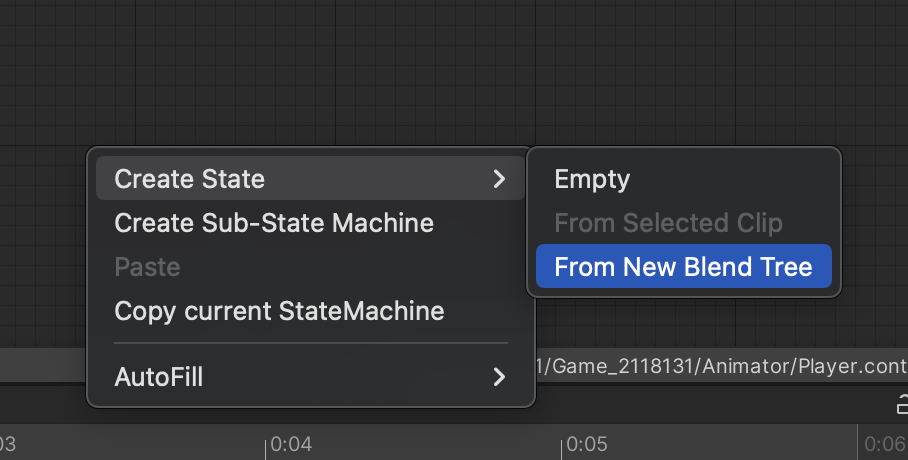
Gambar 9.30 Hasil Tampilan Creat New Animation

1. Pada tab Project buka folder karakter lalu pilih Idle dan pilih gambar Fall, kemudian drag ke tab Animation.



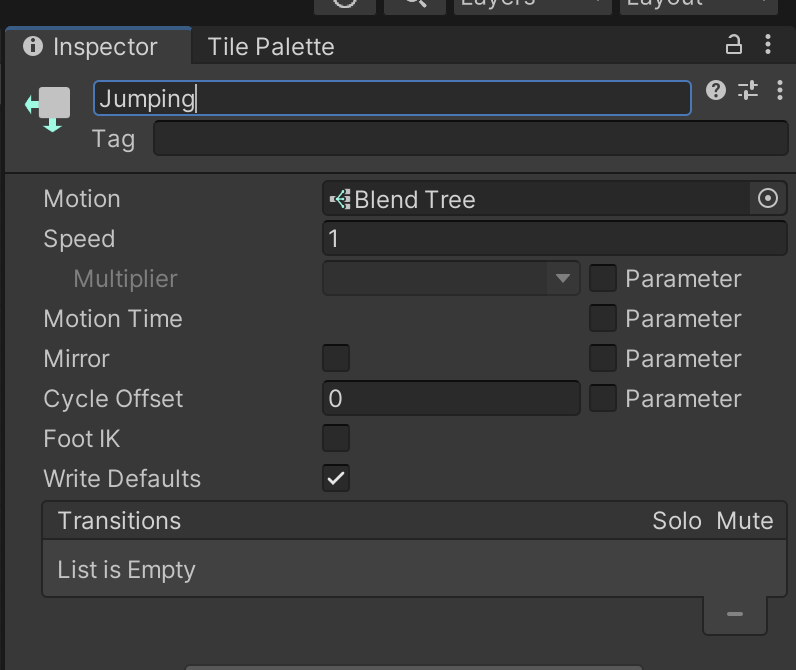
Gambar 9.31 Hasil Tampilan Drag Gambar

1. Kemudian untuk menambahkan animasi ketika melompat. Klik kanan pada menu Animator, di area kosong , pilih Create State>From New Blend Tree



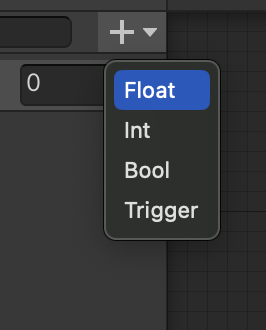
Gambar 9.31 Hasil Tampilan Create From New Blend Tree

1. Pada Animator klik Blend Tree, di menu Inspector, ubah namanya menjadi Jumpig



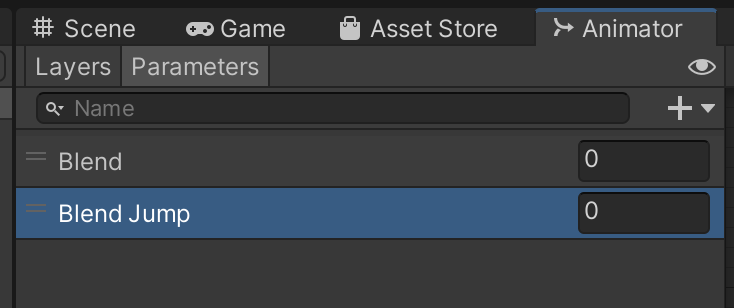
Gambar 9.31 Hasil Tampilan Inspector

1. Pada menu Parameteres tambahkan parameter tipe data Float tekan icon + dan ubah namanya menjadi “Blend Jump”



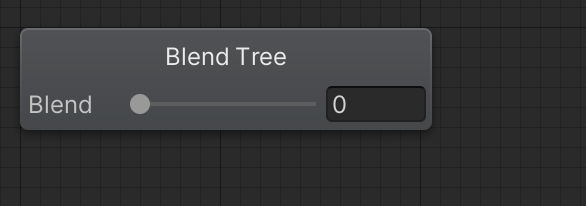
Gambar 9.31 Hasil Tampilan Create Parameters

1. Hasilnya akan seperti pada gambar di bawah



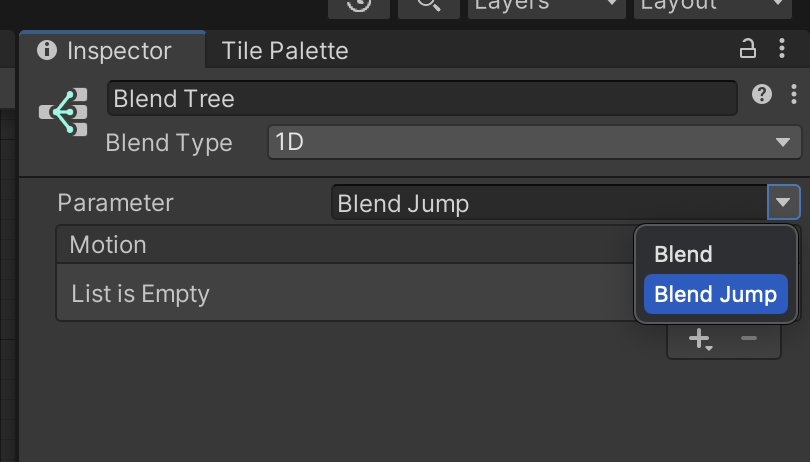
Gambar 9.31 Hasil Tampilan Parameters

1. Pada menu Animator, Klik dua kali pada Blend Tree “Jumping”, Tekan pada Blend Tree



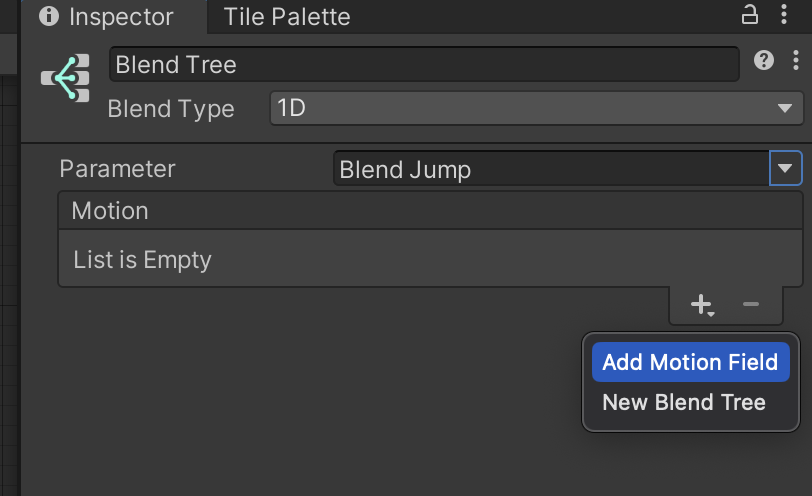
Gambar 9.31 Hasil Tampilan Blend Tree

1. Klik 2X Blend Tree “Jumping”, pada inspector ubah parameter menjadi “Blend Jump”



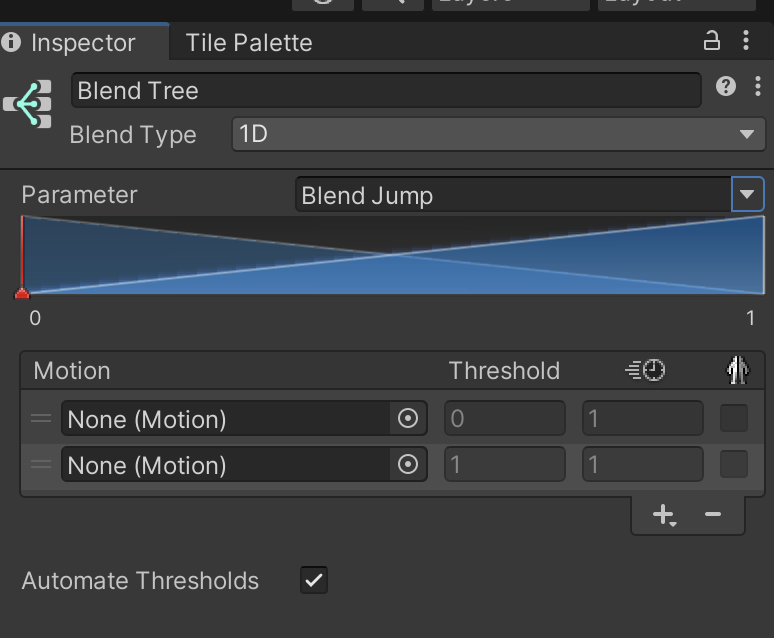
Gambar 9.31 Hasil Tampilan parametes Blend Tree

1. Buka menu Inspector, tekan icon + dan pilih Add Motion Field. Tambahkan dua Motion Field



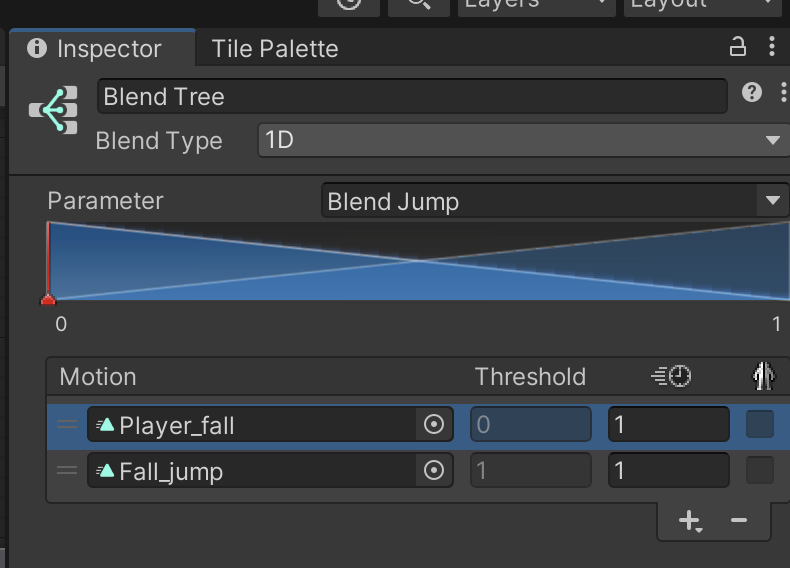
Gambar 9.31 Hasil Tampilan Creat Motion

1. Hasil yang sudah tambahkan



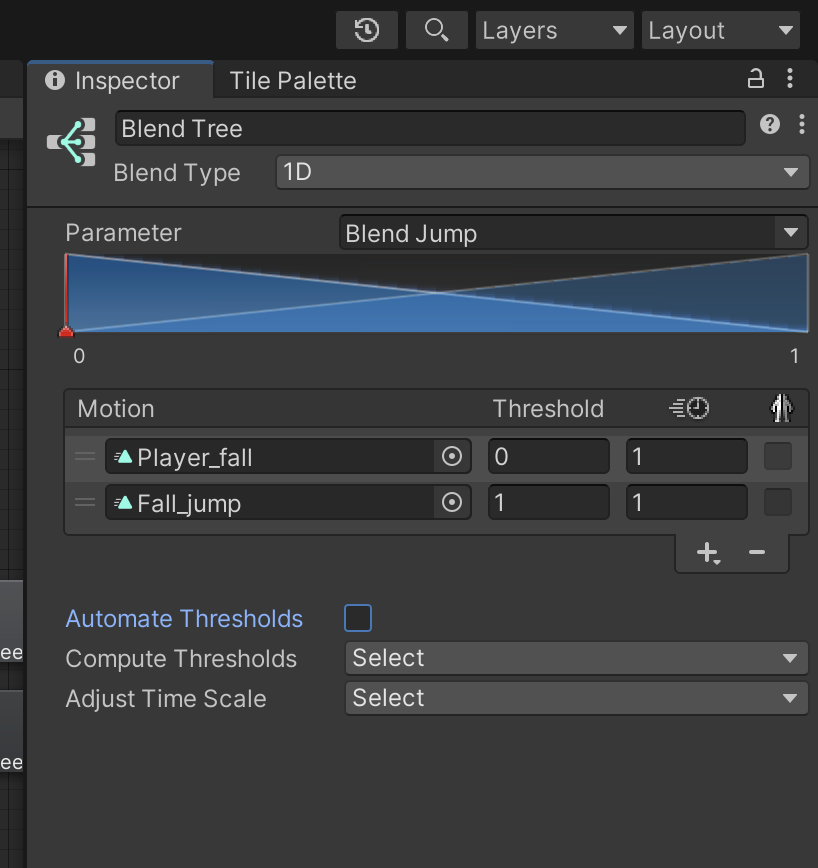
Gambar 9.32 Hasil Tampilan Motion

1. Klik bagian icon None (Motion), maka akan muncul Windows Motion, Tambahkan Sesuai dengan urutan



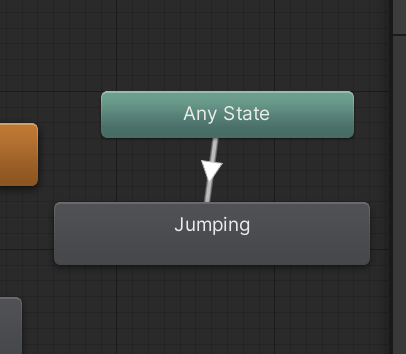
Gambar 9.33 Hasil Tampilan Setting Motion

1. Hilangkan centang “Automate Thresholds” dan atur nilai Threshold seperti berikut



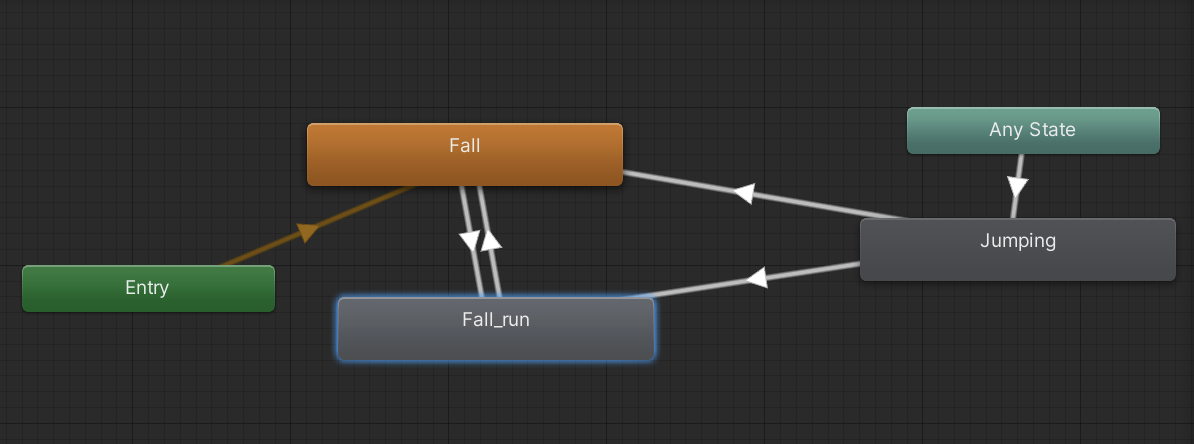
Gambar 9.34 Hasil Tampilan Unchek Automate Thresholds

1. Kembali ke Base Layer, klik kanan Any State, pilih Make Transition dan arahkan panahnya ke Jumping



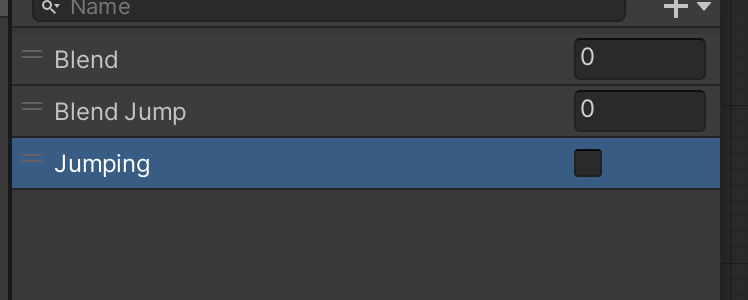
Gambar 9.35 Hasil Tampilan Make Transition

1. Klik kanan Jumping, pilih Make Transition dan arahkan panahnya ke Fall dan Fall\_run



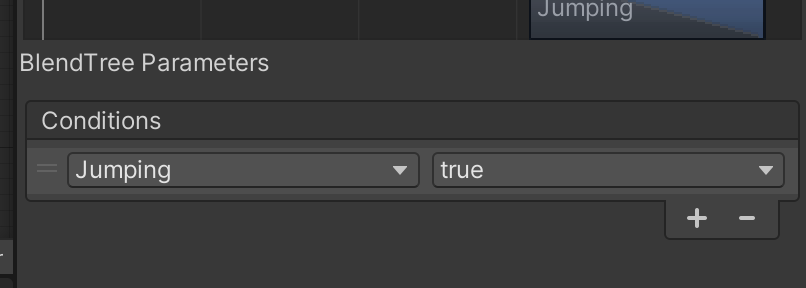
Gambar 9.36 Hasil Tampilan Arah Make Transition

1. Tambahkan parameter transisi dengan tipe data Bool tekan icon + dan ubah namanya menjadi “Jumping”



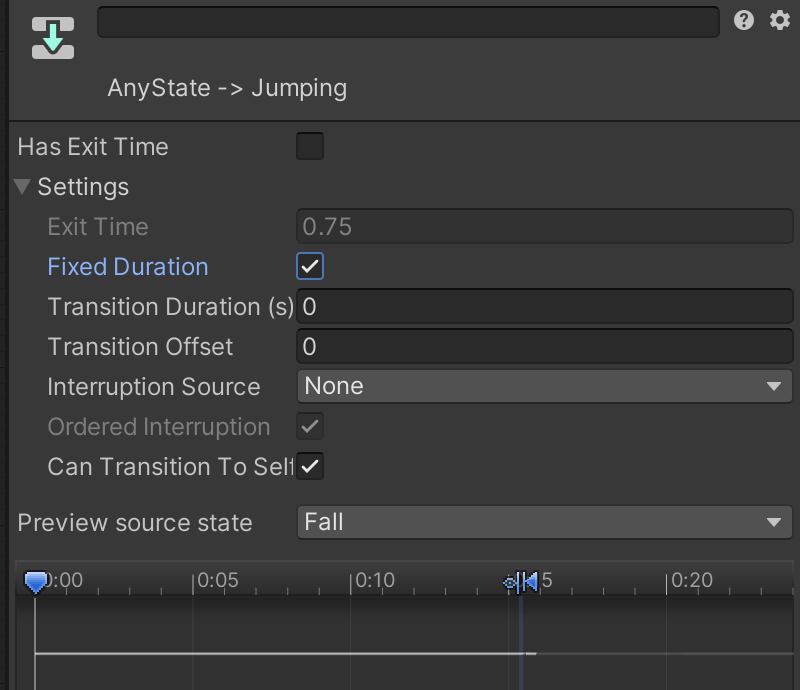
Gambar 9.37 Hasil Tampilan Creat Parameters

1. Klik panah yang mengarah ke Jumping, pada inspector tambahkan condition, pilih condition Jumping dan ubah nilainya menjadi true



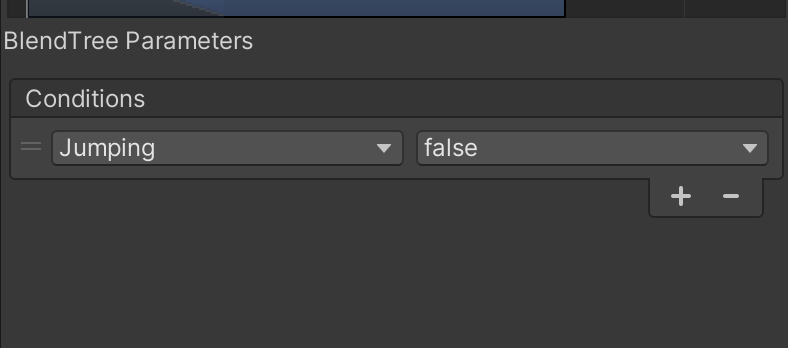
Gambar 9.38 Hasil Tampilan Vlaue Conditions Jumping

1. Klik Settings dan ubah nilai Transition Duration menjadi 0 dan hilangkan centang Has Exit Time



Gambar 9.39 Hasil Tampilan Setting Inspector

1. Klik panah yang mengarah ke Fall dan Fall\_run, pada inspector tambahkan condition, pilih condition Jumping, pada arah panah ke Fall ubah menjadi false, pada arah panah ke Fall\_run ubah menjadi true

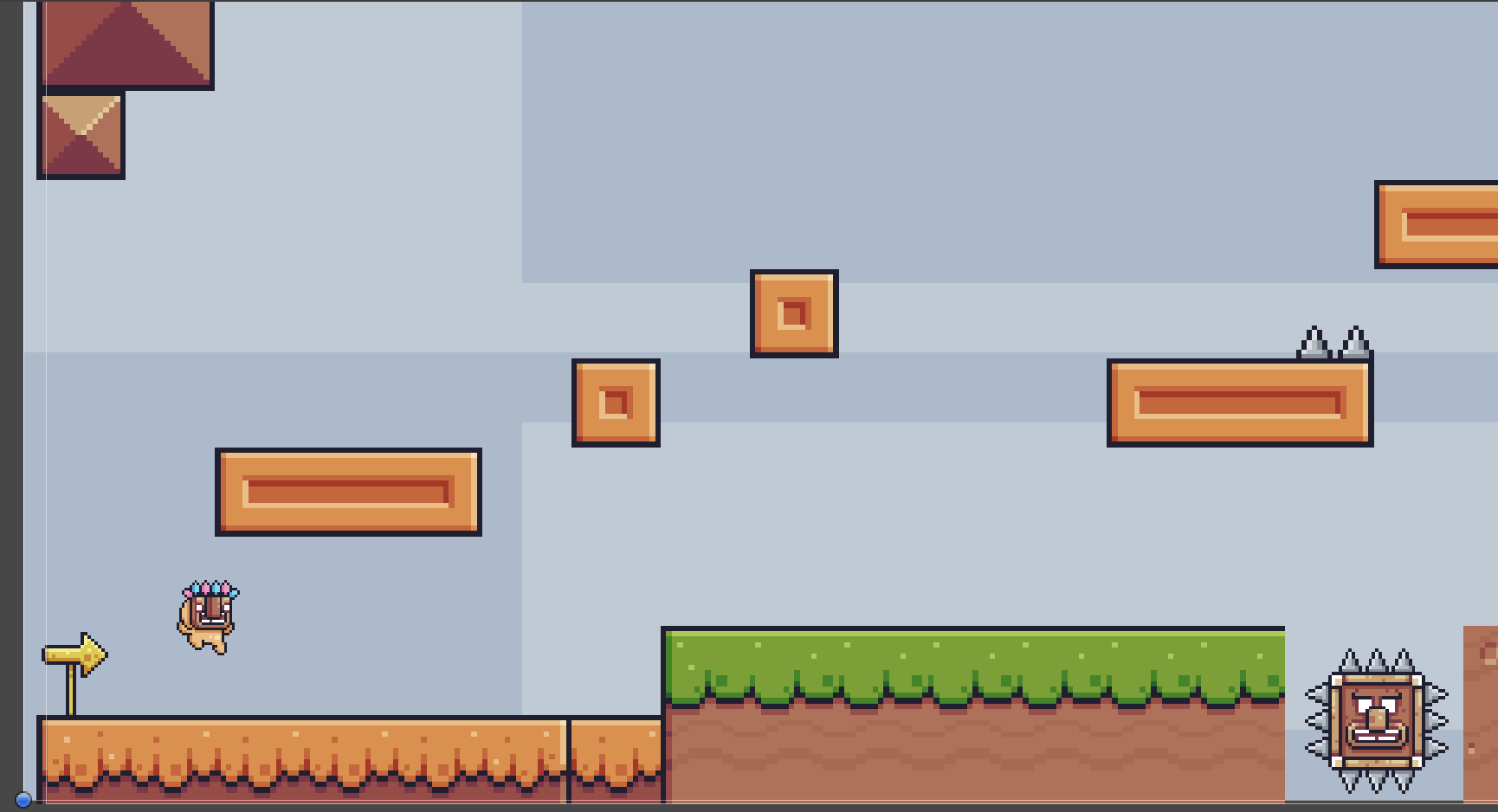


Gambar 9.40 Hasil Tampilan Conditions

1. Ubah Semua Source code pada player

|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Player : MonoBehaviour  {  Rigidbody2D rb;  public Animator animator;  [SerializeField] Transform groundcheckCollider;  [SerializeField] LayerMask groundLayer;  const float groundCheckRadius = 0.2f; // +  [SerializeField] float speed = 1;  [SerializeField] float jumpPower = 70;  bool jump;  float horizontalValue;  [SerializeField] bool isGrounded; // +  bool facingRight;  private void Awake()  {  rb = GetComponent<Rigidbody2D>();  animator = GetComponent<Animator>();  }  void Update ()  {  horizontalValue = Input.GetAxisRaw("Horizontal");  if (Input.GetButtonDown("Jump")){  jump = true;  animator.SetBool("Jumping", true);  }  else if (Input.GetButtonUp("Jump"))  jump = false;  }  void FixedUpdate()  {  GroundCheck();  Move(horizontalValue, jump);  animator.SetFloat("Blend", Mathf.Abs(rb.velocity.x));  animator.SetFloat("Blend Jump", rb.velocity.y);  }  void GroundCheck()  {  isGrounded = false;  Collider2D[] colliders = Physics2D.OverlapCircleAll(groundcheckCollider.position, groundCheckRadius, groundLayer);  if (colliders.Length > 0){  isGrounded = true;  }  animator.SetBool("Jumping", !isGrounded);  }  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(-3, 3, 3);  facingRight = false;  }  else if (!facingRight && dir > 0)  {  // ukuran player  transform.localScale = new Vector3(3, 3, 3);  facingRight = true;  }  #endregion  }  } |

1. Hasil yang sudah tambahkan



Gambar 9.32 Hasil Tampilan Akhir

1. **Link Ghithub**

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

## Tugas 2 : Lengkapi Source code dibawah ini

|  |
| --- |
| void HandleJumpInput()  {  if (Input.GetKeyDown(KeyCode.Space))  {  animator.SetBool("isJumping", true);  rb.AddForce(Vector2.up \* jumpForce, ForceMode2D.Impulse);  }  else if (Input.GetKey(KeyCode.Space))  {  animator.SetBool("isJumping", true);  }  else  {  animator.SetBool("isJumping", false);  }  }  void HandleMovementInput()  {  float move = Input.GetAxis("Horizontal");  if (move == 0)  {  animator.SetBool("isIdle", true);  animator.SetBool("isWalking", false);  }  else  {  animator.SetBool("isIdle", false);  animator.SetBool("isWalking", true);  transform.Translate(Vector3.right \* move \* Time.deltaTime);  }  if (move < 0)  {  transform.localScale = new Vector3(-1, 1, 1);  }  else if (move > 0)  {  transform.localScale = new Vector3(1, 1, 1);  }  } |

Penjelasan :

Pada source code atas yaitu melakukan. Perubahan dan penambahan pada metode HandleJumpInput memastikan animasi lompat (isJumping) diaktifkan saat tombol Space ditekan atau ditahan dan dinonaktifkan saat tombol dilepas, sehingga animasi tidak tetap aktif setelah tombol dilepas. Pada metode HandleMovementInput, kondisi untuk idle diperbaiki dengan menggunakan move == 0, animasi berjalan diaktifkan saat ada pergerakan, dan arah pergerakan diubah dari Vector3.left ke Vector3.right agar karakter bergerak ke kanan saat menerima input positif. Selain itu, skala objek diatur ulang agar karakter menghadap ke arah yang benar berdasarkan arah gerakan.