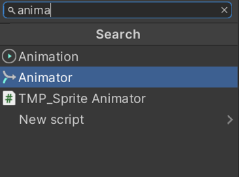
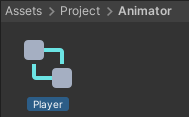
# 9 Game Animation

|  |  |  |
| --- | --- | --- |
| NIM | : | 2118113 |
| Nama | : | Vincnet Mikael |
| Kelas | : | D |
| Asisten Lab | : | Nayaka Apta Nayottama (2218102) |

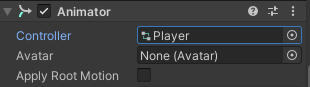
1. Membuat Character Animation
2. Pada karakter klik inspector kemudian pilih Add Component Animator



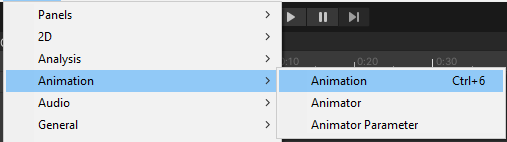
1. Pada folder Project  Buat Folder baru “Animator”. Buat File Animator Controller pada folder Animator , ubah namanya menjadi Player.



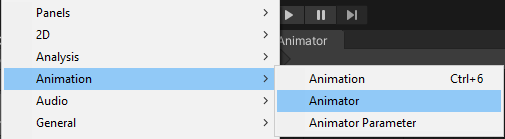
1. Klik player pada Hierarchy, kemudian cari Component Animator, pada setting Controller ubah menjadi Player



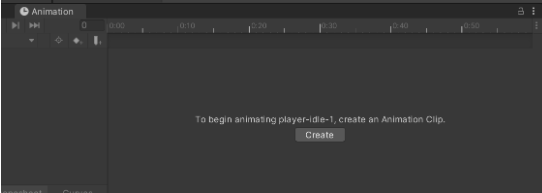
1. Tambahkan menu panel Animation di menu Window, pilih Animation > Animation.



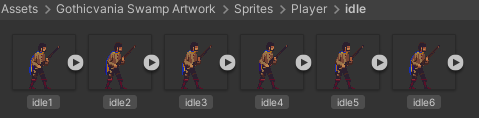
1. Tambahkan menu panel Animator di menu Window, pilih Animation > Animator.



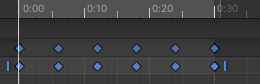
1. Untuk membuat animasi klik karakter pada Hierarchy, kemudian ke menu panel Animation, pilih Create. Simpan pada folder Animator dan beri nama “karakter”.



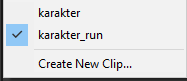
1. Ikuti pathway folder seperti gambar berikut, lalu pilih player idle 1-6, kemudian drag ke tab Animation.



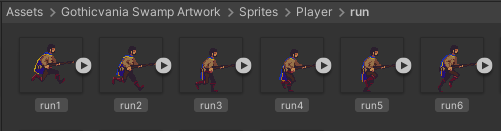
1. Tekan CTRL + A pada menu panel Animation geser kotak kecil pada timeline sampai frame 0:30 agar animasinya tidak terlalu cepat.



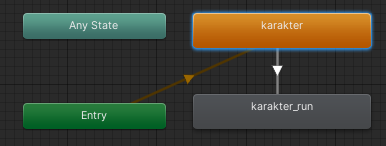
1. Buat animasi baru, Klik pada “karakter” kemudian pilih Create New Clip, dan beri nama “karakter\_run”, Simpan pada Folder Animator.



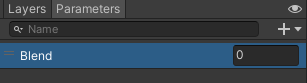
1. Ikuti pathway berikut, lalu copy player\_run, drag and drop pada menu Animation.
2. pada panel timeline tekan Ctrl+A di keyboard, klik bagian kotak kecil disamping keyframe terakhir dan geser sampai waktu 0:25.



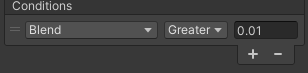
1. pilih ke menu Animator yang telah dibuka sebelumnya, Kemudian buat transisi antara karakter dan karakter\_run dengan cara klik kanan pada karakter dan pilih Make Transition dan tarik ke karakter\_run



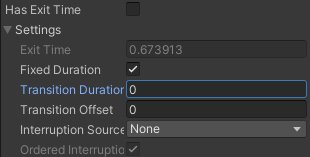
1. Masuk ke tab parameter, tambahkan tipe data bdengan cara tekan icon tambah dan ubah namanya menjadi “Blend”.



1. Klik panah putih tersebut, pada bagian conditions klik icon tambah kemudian atur menjadi “Blend”. Atur nilai conditions blend tersebut menjadi 0.01



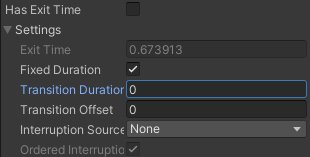
1. Pada bagian Settings, hilangkan centang pada Has Exit Time dan atur nilai Transition Duration menjadi 0.



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



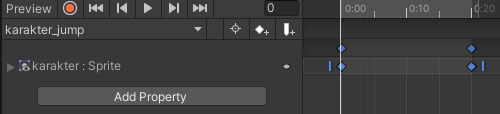
1. Pada bagian Settings, hilangkan centang pada Has Exit Time dan atur nilai Transition Duration menjadi 0.



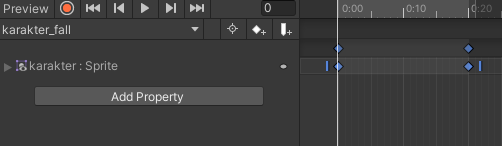
1. Ubah Source code pada Player menjadi seperti berikut.

|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class player : MonoBehaviour  {  public Animator animator;  Rigidbody2D rb;  [SerializeField] Transform groundcheckCollider;  [SerializeField] LayerMask groundLayer;  const float groundCheckRadius = 0.2f; // +  [SerializeField] float speed = 1;  [SerializeField] float jumpPower = 500;  float horizontalValue;  [SerializeField] bool isGrounded; // +  bool facingRight;  bool jump;  void FixedUpdate()  {  GroundCheck();  Move(horizontalValue, jump);  animator.SetFloat("Blend",Mathf.Abs(rb.velocity.x));  }  void GroundCheck()  {  isGrounded = false;  Collider2D[] colliders = Physics2D.OverlapCircleAll(groundcheckCollider.position, groundCheckRadius, groundLayer);  if (colliders.Length > 0)  isGrounded = true;  }  private void Awake()  {  rb = GetComponent<Rigidbody2D>();  animator = GetComponent<Animator>();  }  void Update ()  {  horizontalValue = Input.GetAxisRaw("Horizontal");  if (Input.GetButtonDown("Jump"))  jump = true;  else if (Input.GetButtonUp("Jump"))  jump = false;  }  void Move(float dir, bool jumpflag)  {  #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(4, 4, 1);  facingRight = false;  }  else if (!facingRight && dir > 0)  {  // ukuran player  transform.localScale = new Vector3(4, 4, 1);  facingRight = true;  }  if(isGrounded && jumpflag)  {  isGrounded = false;  jumpflag = false;  rb.AddForce(new Vector2(2f, jumpPower));  }  #endregion    }  } |

1. Kemudian buat animasi baru tekan tulisan “Player\_run” kemudian pilih Create New Clip, dan beri nama “Player\_jump”. Pada folder player buka jump lalu pilih gambar player-jump-1, kemudian drag ke tab Animation.

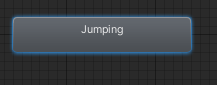


1. Buat animasi baru dengan cara tekan tulisan “Player\_jump" kemudian pilih Create New Clip, dan beri nama “Player\_fall”. Pada tab Project buka folder yang berisi file karakter\_fall, kemudian drag ke tab Animation

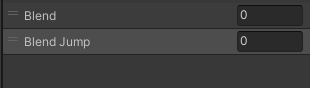


1. Kemudian untuk menambahkan animasi ketika melompat. Klik kanan pada menu Animator, di area kosong , pilih Create State>From New Blend Tree

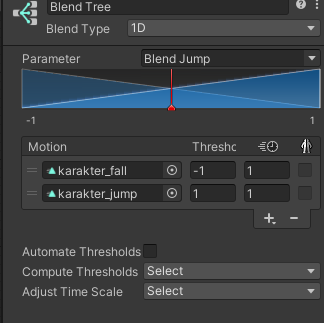
Pada Animator klik Blend Tree, di menu Inspector, ubah namanya menjadi Jumpig



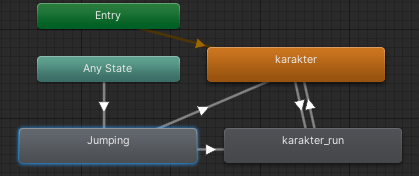
1. Pada menu Parameteres tambahkan parameter tipe data Float tekan icon + dan ubah namanya menjadi “Blend Jump”.



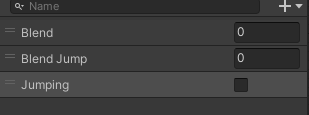
1. Pada menu Animator, Klik dua kali pada Blend Tree “Jumping”, Tekan pada Blend Tree. Lalu atur menjadi seperti berikut.



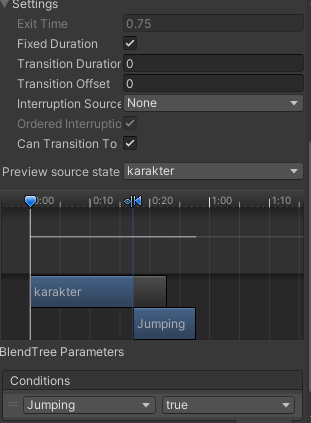
1. Pada Base layer buat transisi menjadi seperti berikut



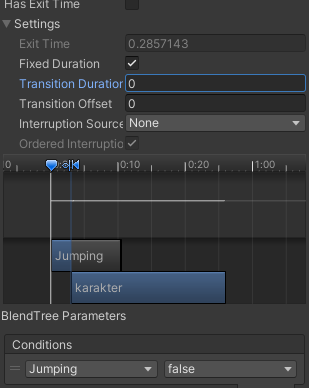
1. Tambahkan parameter transisi dengan tipe data Bool tekan icon + dan ubah namanya menjadi “Jumping”.



1. Klik panah yang mengarah ke Jumping, pada inspector tambahkan condition, pilih condition Jumping dan ubah nilainya menjadi true. Klik Settings dan ubah nilai Transition Duration menjadi 0 dan hilangkan centang Has Exit Time.



1. Klik panah yang mengarah ke karakter dan karakter\_run, pada inspector tambahkan condition, pilih condition Jumping, pada arah panah ke karakter ubah menjadi false, pada arah panah ke karakter\_run ubah menjadi true. Klik Settings dan ubah nilai Transition Duration menjadi 0 dan hilangkan centang Has Exit Time.



1. Ubah source code pada player menjadi seperti berikut

|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class player : MonoBehaviour  {    public Animator animator;    Rigidbody2D rb;    [SerializeField] Transform groundcheckCollider;    [SerializeField] LayerMask groundLayer;    const float groundCheckRadius = 0.2f; // +    [SerializeField] float speed = 1;    [SerializeField] float jumpPower = 500;    float horizontalValue;    [SerializeField] bool isGrounded; // +    bool facingRight;    bool jump;    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);    }    private void Awake()    {      rb = GetComponent<Rigidbody2D>();      animator = GetComponent<Animator>();    }    void Update ()    {      horizontalValue = Input.GetAxisRaw("Horizontal");      if (Input.GetButtonDown("Jump")){        animator.SetBool("Jumping",true);        jump = true;      }      else if (Input.GetButtonUp("Jump"))          jump = false;    }    void Move(float dir, bool jumpflag)    {      #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(4, 4, 1);        facingRight = false;      }      else if (!facingRight && dir > 0)      {        // ukuran player        transform.localScale = new Vector3(4, 4, 1);        facingRight = true;      }      if(isGrounded && jumpflag)      {          isGrounded = false;          jumpflag = false;          rb.AddForce(new Vector2(2f, jumpPower));      }      #endregion    }  } |

1. Kuis
2. Analisa source code berikut

|  |
| --- |
| void HandleJumpInput()      {          if (Input.GetKeyDown(KeyCode.Space))          {              animator.SetBool("isJumping", );              rb.AddForce(Vector2.up \* jumpForce, ForceMode2D.Impulse);          }          else if (Input.GetKey(KeyCode.Space))          {              animator.SetBool("isJumping",);          }      }      void HandleMovementInput()      {          float move = Input.GetAxis("Horizontal");          if (move != 1)          {              animator.SetBool("isIdle", true);              transform.Translate(Vector3.left \* move \* Time.deltaTime);          }          else          {              animator.SetBool("isWalking", false);          }          if (move != 0)          {              transform.localScale = new Vector3(-4, 1, 1);          }          else if (move > 0)          {              transform.localScale = new Vector3(1, 2, 1);          }      } |

Perbaikan

|  |
| --- |
| void HandleJumpInput()  {  if (Input.GetKeyDown(KeyCode.Space))  {  animator.SetBool("isJumping", true); // Diperbaiki: Menambahkan nilai boolean  rb.AddForce(Vector2.up \* jumpForce, ForceMode2D.Impulse);  }  else if (Input.GetKey(KeyCode.Space))  {  animator.SetBool("isJumping", true); // Diperbaiki: Menambahkan nilai boolean  }  }  void HandleMovementInput()  {  float move = Input.GetAxis("Horizontal");  if (move == 0) // Diperbaiki: Ubah kondisi dari move != 1 menjadi move == 0  {  animator.SetBool("isIdle", true);  }  else  {  animator.SetBool("isIdle", false); // Ditambahkan: Menonaktifkan animasi idle saat bergerak  animator.SetBool("isWalking", true); // Ditambahkan: Mengaktifkan animasi berjalan saat bergerak  transform.Translate(Vector3.right \* move \* Time.deltaTime); // Diperbaiki: Mengubah ke Vector3.right untuk gerakan horizontal  if (move < 0) // Diperbaiki: Menambahkan kondisi untuk gerakan ke kiri  {  transform.localScale = new Vector3(-1, 1, 1); // Gerakan ke kiri  }  else if (move > 0) // Diperbaiki: Menambahkan kondisi untuk gerakan ke kanan  {  transform.localScale = new Vector3(1, 1, 1); // Gerakan ke kanan  }  }  } |

Penjelasan

Kode diperbaiki dengan menambahkan nilai boolean `true` pada `animator.SetBool("isJumping", true);` di `HandleJumpInput`. Dalam `HandleMovementInput`, kondisi diperbaiki untuk memisahkan gerakan ke kiri dan ke kanan. Animasi idle dinonaktifkan dan animasi berjalan diaktifkan saat bergerak. Skala karakter disesuaikan berdasarkan arah gerakan, memperbaiki logika skala yang salah sebelumnya. Perbaikan ini memastikan animasi dan gerakan karakter berfungsi sesuai..