



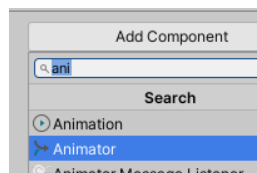
TUGAS PERTEMUAN: 9

Game Animation

NIM	:	2118101
Nama	:	Marie Pangestu
Kelas	:	C
Asisten Lab	:	Rifal Rifqi Rhomadon (2218106)

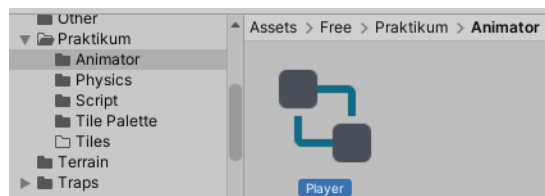
1.1 Tugas 1 : Membuat Character Animation

1. Tekan karakter pada hierarchy lalu tambahkan komponen “Animator” pada inspector.



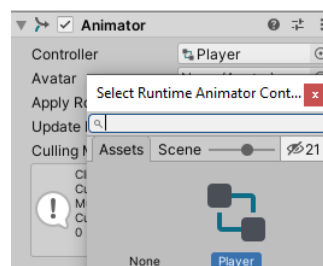
Gambar 9.1 Add Animator

2. Buat folder baru didalam folder Praktikum dengan nama “Animator”, kemudian klik kanan didalam folder animator>animator controller> nama “Player”.



Gambar 9.2 Add Animator Controller Player

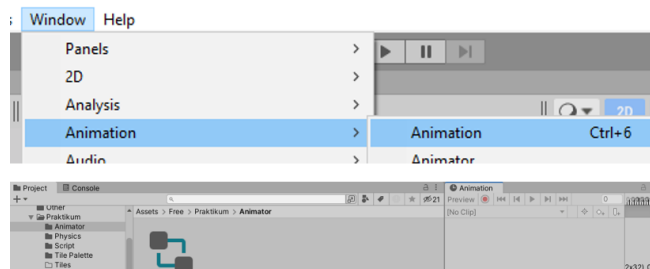
3. Tekan animator pada inspector karakter, ubah controller menjadi player yang baru saja dibuat.



Gambar 9.3 Ubah Controller Animator Player

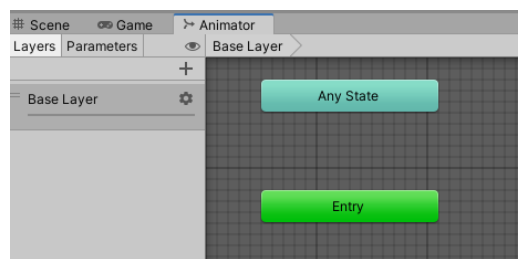


4. Tambahkan animation, window>Animation>Animation, kemudian drag hingga tampilan split seperti gambar 9.4.



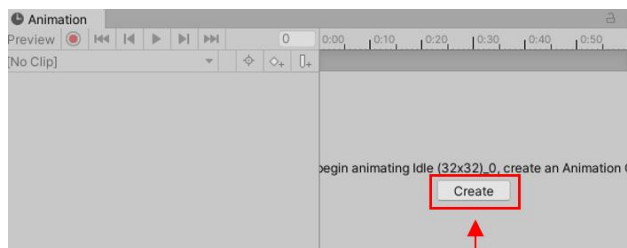
Gambar 9.4 Add Animation

5. Tambahkan juga animator, window>Animation>Animator.



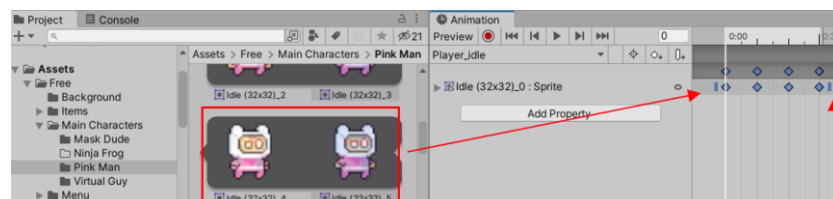
Gambar 9.5 Add Animator

6. Kemudian tekan player pada hierarchy, lalu tekan tombol create pada animation di split panel project untuk menambahkan animasi, lalu simpan didalam folder animator beri nama 'Player_idle'.



Gambar 9.6 Create Animation Player

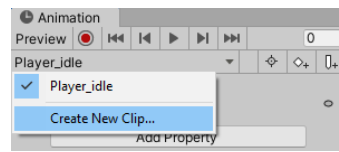
7. Cari idle pada file karakter idle 1,4, dan 5, lalu drag and drop menuju keyframe dan Tarik kotak biru kecil sebelah kanan untuk menyesuaikan frame.



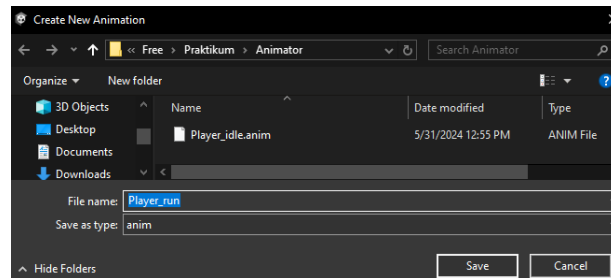
Gambar 9.7 Add Player Idle Animation



8. Tekan player_idle lalu pilih create new clip beri nama 'Player_run' simpan di folder yang sama.

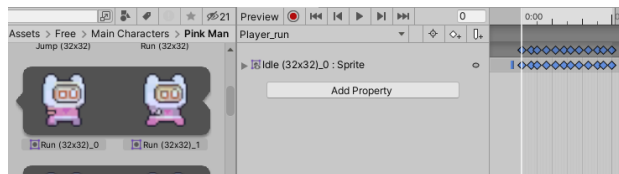


Gambar 9.8 Create New Clip Player Run



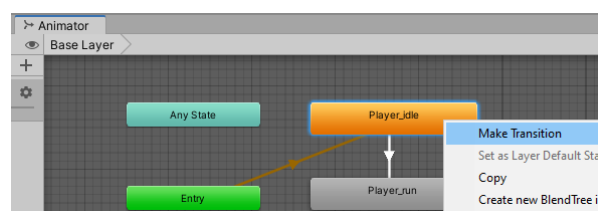
Gambar 9.9 Simpan Player Run

9. Tambahkan animasi run dari file project kedalam frame player run seperti Langkah 7.



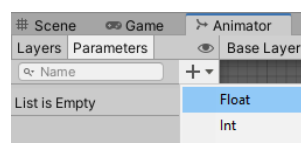
Gambar 9.10 Add Player Run Animation

10. Beralih pada animator, klik kanan pada player_idle lalu tekan make transition dan arahkan panah menuju player_run.



Gambar 9.11 Make Transition Idle To Run

11. Kemudian tekan tab parameter dan tekan tanda (+) untuk menambah parameter dengan tipe data float dan nama "Blend".



Gambar 9.12 New Parameter Blend

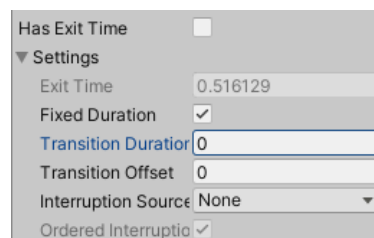


12. Kemudian tekan panah putih dari idle ke run, lalu pada inspector tekan (+) pada condition dan tambahkan parameter Blend yang sudah dibuat dengan nilai 0.01.



Gambar 9.13 Blend Condition Player Idle And Run

13. Lalu diatas condition, unchecklist 'Has Exit Time' dan ubah nilai transition duration menjadi 0.



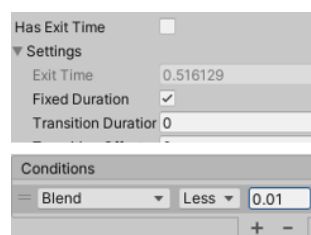
Gambar 9.14 Unchecklist Has Exit Time and Transition Duration

14. Buat transisi juga dari player_run menuju player_idle.



Gambar 9.15 Transition Run to Idle

15. Lalu unchecklist has exit time, ubah nilai transition duration menjadi 0, dan tambahkan condition blend dan ubah operatornya dari 'Greater' menjadi 'Less' dengan nilai 0.01.



Gambar 9.16 Settings Inspector Dari Transition Run to Idle



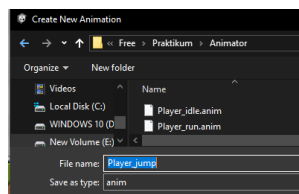
16. Tambahkan kode berikut pada script player, kemudian tekan play untuk uji game apakah animasi sudah teraplikasi.

```
public class Player : MonoBehaviour
{
    public Animator animator;
    Rigidbody2D rb;

    private void Awake()
    {
        rb = GetComponent<Rigidbody2D>();
        animator = GetComponent<Animator>();
    }

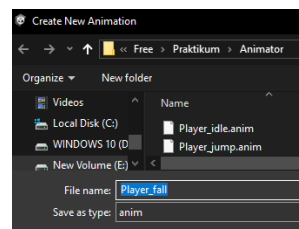
    void FixedUpdate()
    {
        GroundCheck();
        Move(horizontalValue, jump);
        animator.SetFloat("Blend", Mathf.Abs(rb.velocity.x));
    }
}
```

17. Tekan karakter pada hierarchy, lalu create new clip dengan nama 'Player_jump'.



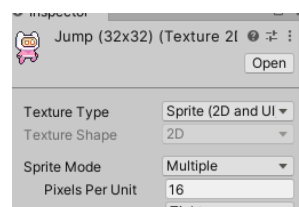
Gambar 9.17 Create New Clip Player Jump

18. Tambahkan animasi dengan nama 'Player_fall'.



Gambar 9.18 Create New Clip Player Fall

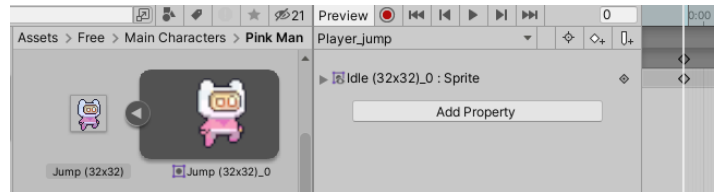
19. Pastikan sprite sudah dalam mode multiple dengan nilai 16 untuk pixel per unit



Gambar 9.19 Cek Sprite Mode dan Pixel Per Unit

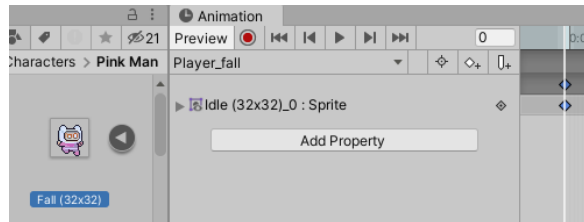


20. Kemudian drag and drop karakter jump kedalam frame animasi jump



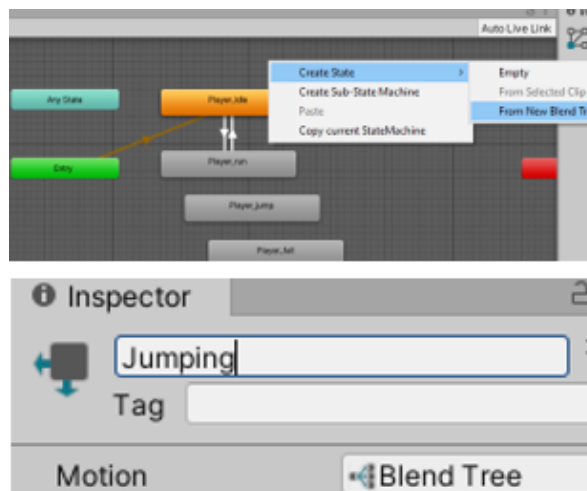
Gambar 9.20 Drag and Drop Jump Character

21. Lakukan Langkah 19-20 untuk animasi fall.



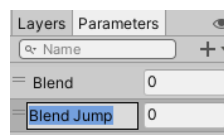
Gambar 9.21 Fall Character

22. Kemudian untuk menambahkan animasi ketika melompat. Klik kanan pada menu Animator, di area kosong , pilih Create State>From New Blend Tree ubah nama menjadi 'Jumping'.



Gambar 9.22 Add New Blend Tree Jumping

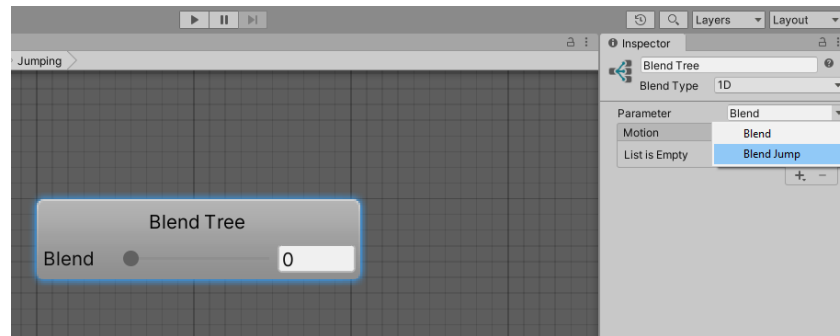
23. Buat parameter baru dengan nama 'Blend Jump'.



Gambar 9.23 Tag Camera

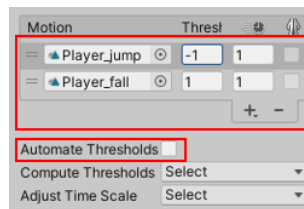


24. Double klik pada blend tree dengan nama 'Jumping', lalu tekan blend tree dan ubah parameter pada inspector menjadi blend jump.



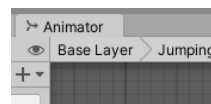
Gambar 9.24 Ubah Parameter Blend Tree Jumping Menjadi Blend Jump

25. Lalu tambahkan dua motion untuk player_jump dan player_fall, unchecklist automatic thresholds dan ubah nilai motion seperti gambar 9.25.



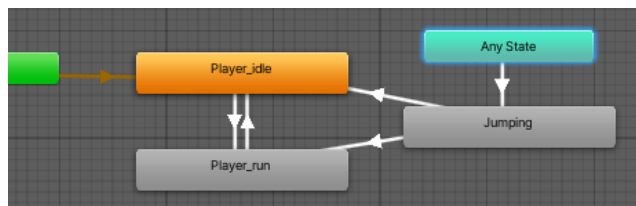
Gambar 9.25 Add Motion Player Jump and Run

26. Kembali ke base layer



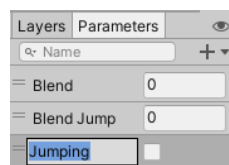
Gambar 9.26 Kembali Ke Base Layer

27. Tambahkan transisi baru seperti gambar 9.27



Gambar 9.28 Buat Transisi Dari Any State ke Jumping, Idle, Run

28. Tambahkan parameter baru bertipe data Bool dengan nama 'Jumping'.



Gambar 9.29 New Parameter Jumping

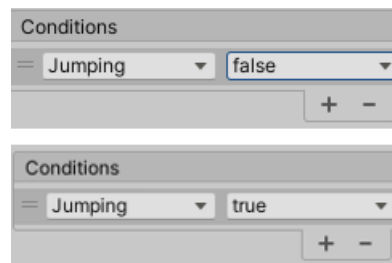


29. Tekan panah yang mengarah dari Any State menuju Jumping, dan ubah pengaturan inspector seperti gambar berikut



Gambar 9.30 Inspector Any State Arrow to Jumping

30. Klik panah yang mengarah ke **Player_idle** dan **Player_run**, pada **inspector** tambahkan **condition**, pilih **condition Jumping**, pada arah panah ke **player_idle** ubah menjadi **false**, pada arah panah ke **player_run** ubah menjadi **true**.



Gambar 9.31 Inspector From Jumping to Idle and Run

31. Tambahkan script tersebut pada script player

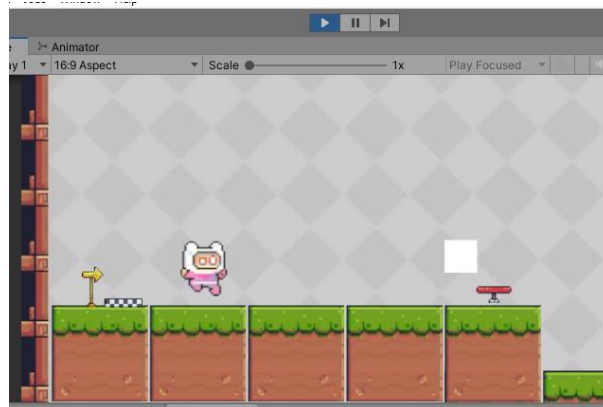
```
void Update ()
{
    horizontalValue = Input.GetAxisRaw("Horizontal");
    if (Input.GetButtonDown("Jump"))
    {
        animator.SetBool("Jumping", true);
        jump = true;
    }
}

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);
}
```




32. Tekan play untuk uji coba animasi



Gambar 9.32 Hasil Tampilan Game Animation

1.3 Kuis

```
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", 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);
        }
    }
}
```



Analisa :

Pada fungsi `HandleJumpInput`, Boolean diberi nilai `true` jika memenuhi kondisi yaitu ketika tombol Space ditekan, jika tidak maka diberi nilai `false`.

Memperbaiki logika di bagian fungsi `HandleMovementInput` agar `isIdle` diatur menjadi `'true'` hanya ketika `'move == 0'`. Dan menambahkan `'animator.SetBool("isWalking", true)'` ketika `'move'` tidak sama dengan 0.

Menyesuaikan `'transform.localScale'` agar karakter menghadap ke arah yang benar berdasarkan nilai `'move'`.