



TUGAS PERTEMUAN: 9

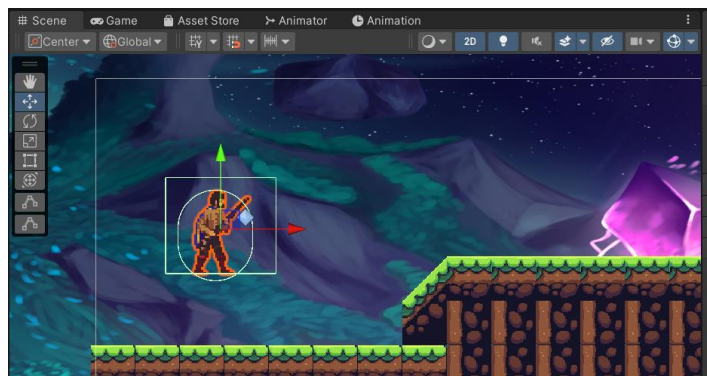
GAME ANIMATION

NIM	:	2118119
Nama	:	Muhammad Yazid Abu Sahal
Kelas	:	D
Asisten Lab	:	BERCHMANS BAYU BIN JAYA (2218034)

9.1 Tugas 9 : Mengimplementasikan Game Animation

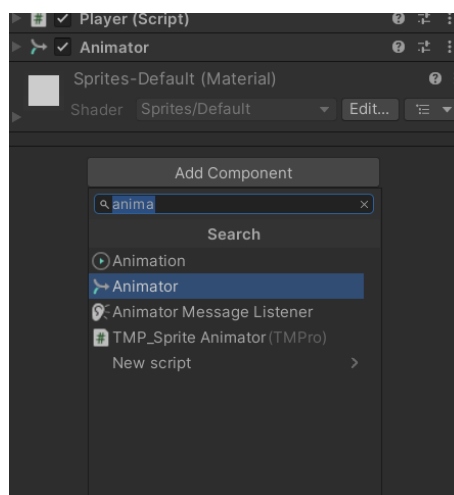
A. Membuat Character Animation

1. Langkah pertama , buka project seblumnya.



Gambar 9.1 Tampilan *Project Unity*

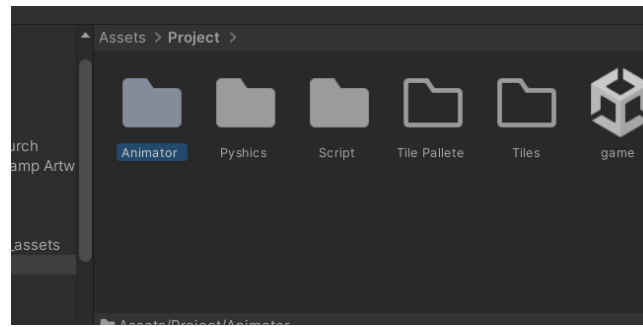
2. *Add Component* baru pada Karakter



Gambar 9.2 Add Component Animator

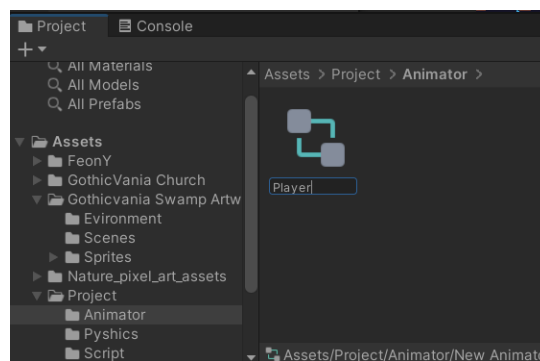


3. Buatlah folder baru di Project dengan nama Animator.



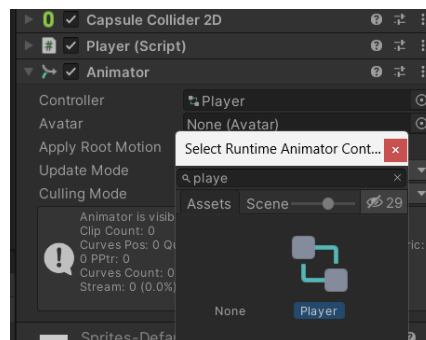
Gambar 9.3 *Capsule Collider Player*

4. Buatlah file animator *controller*. Caranya klik kanan folder animator, lalu pilih *create* dan animator *controller*. Beri nama Player.



Gambar 9.4 *Animator Controller*

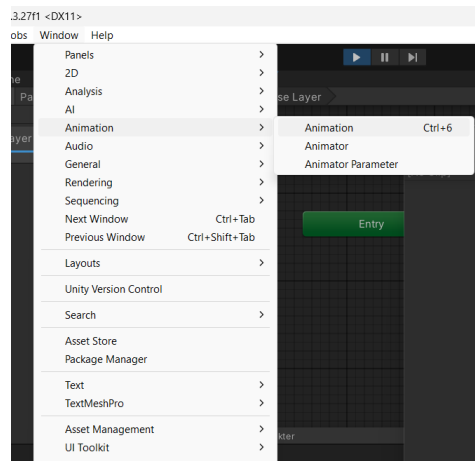
5. Di bagian komponen animator, jadikan controller menjadi Player.



Gambar 9.5 *Component Animator*

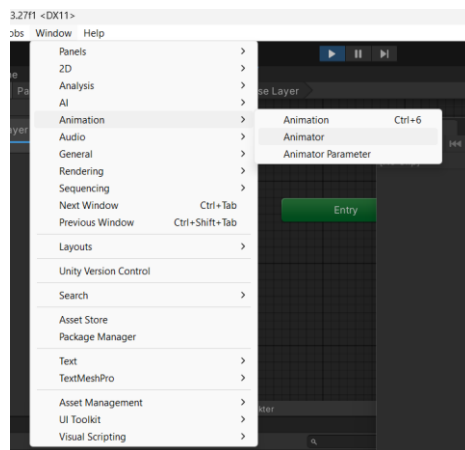


6. Tambahkan window animation dengan cara dibawah ini.



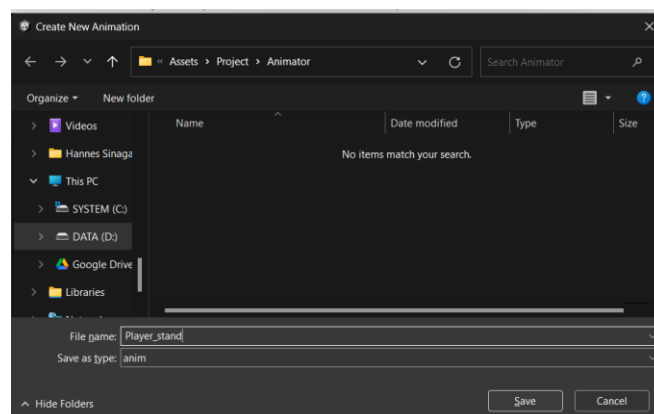
Gambar 9.6 Menu Panel *Animation*

7. Tambahkan juga window Animator.



Gambar 9.7 Menu Panel *Animator*

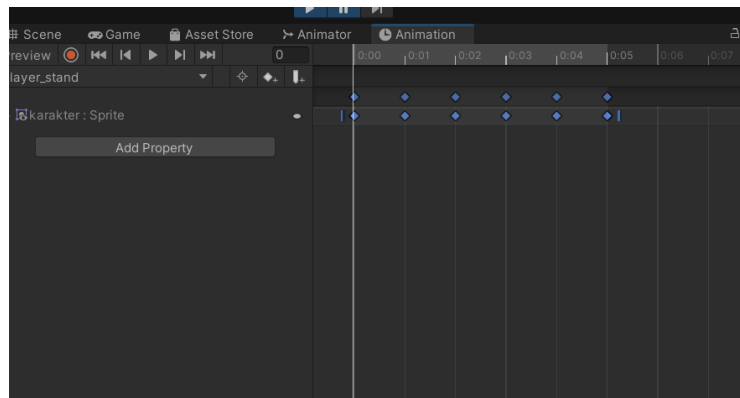
8. Di window animation, buat button create dan beri nama Player_stand.



Gambar 9.8 *Create Animation Clip*

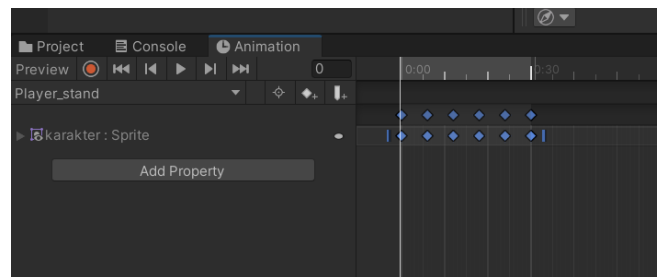


9. Drag and drop asset player stand dan player idle.



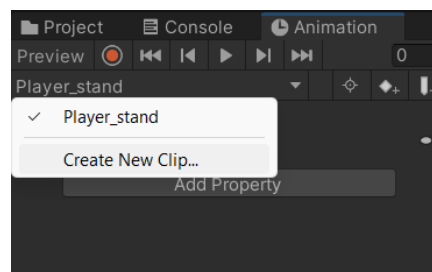
Gambar 9.9 *Player_idle Animation*

10. Tarik frame sampai di frame 30 agar gerak nya tidak terlalu cepat.



Gambar 9.10 *Player_run Animation*

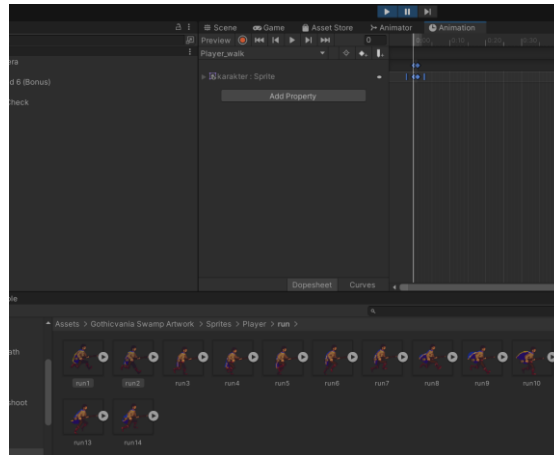
11. Buat clip baru dan beri nama Player_walk.



Gambar 9.11 *Transition Player_idle dengan Player_run*

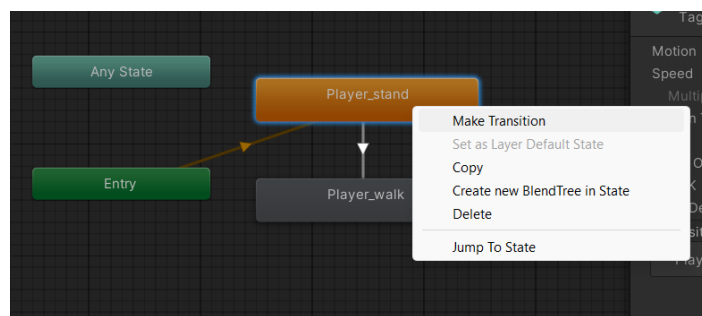


12. Selanjutnya drag and drop ke frame animation untuk asset run1 dan run2.
Ubah durasinya menjadi 25



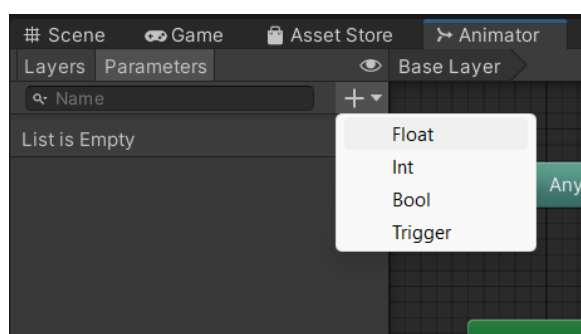
Gambar 9.12 *Conditions Blend*

13. Pada window animation, buat make transition dari player_stand ke player_walk.



Gambar 9.13 *Make Transition*

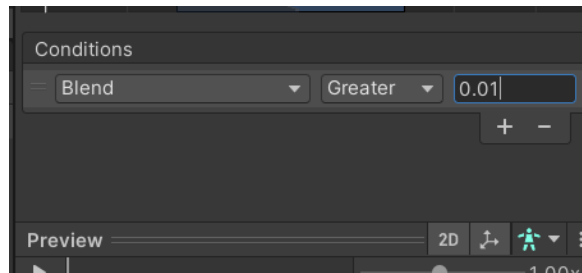
14. Buat parameters baru dengan tipe data float dan beri nama Blend.



Gambar 9.14 *New Parameters*

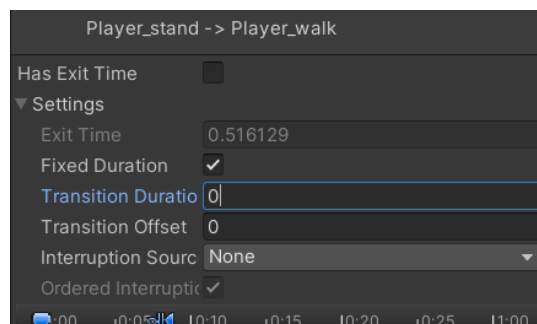


15. Klik arah panah player_stand ke player_walk. Buat conditions baru dan atur seperti dibawah ini.



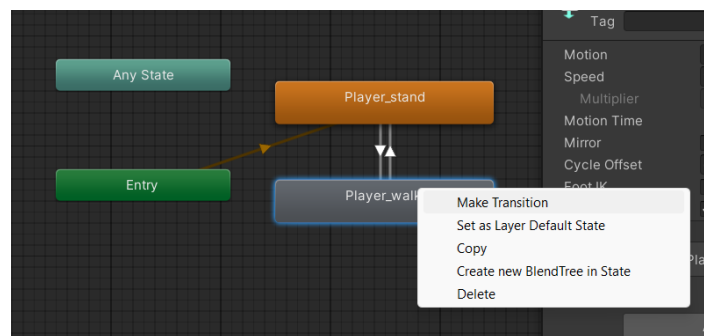
Gambar 9.15 *New Conditions*

16. Di bagian setting, *unchecklist Has Exit Time* ubah nilai *Transition Duration* menjadi 0.



Gambar 9.16 *Atur Durasi Transisi*

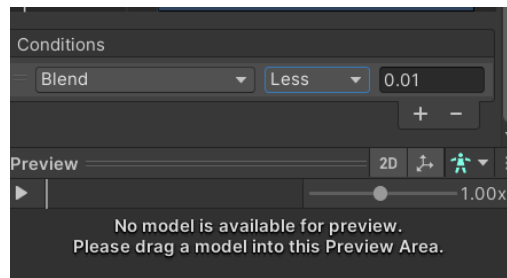
17. Buat transition dari arah player_walk ke player_stand.



Gambar 9.17 *New Transition*



18. Lakukan langkah ke-15 dan 16, hanya saja perbedaan terletak pada bagian greater diganti dengan less.



Gambar 9.18 Atur *Conditions*

19. Tambahkan source code berikut untuk mendefinisikan class animator

```
public Animator animator;
```

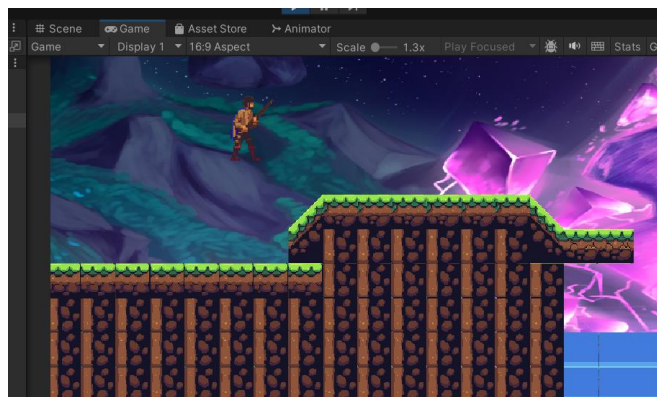
20. Tambahkan source code berikut pada void Awake().

```
animator = GetComponent<Animator>();
```

21. Tambahkan source code berikut pada void FixedUpdate()

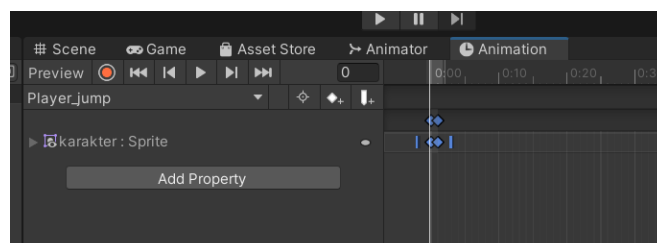
```
animator.SetFloat("Blend", Mathf.Abs(rb.velocity.x));
```

22. Jalankan untuk menguji apakah berhasil.



Gambar 9.19 Menjalankan Game

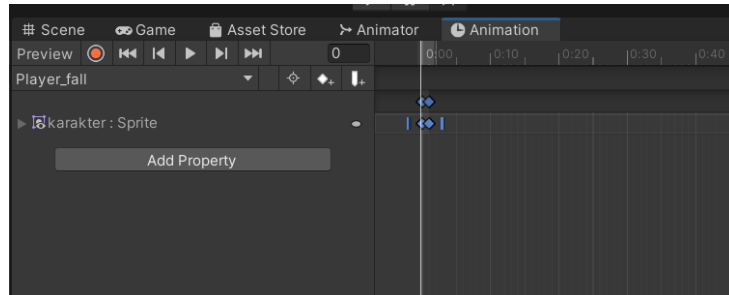
23. Buat clip baru dan beri nama Player_jump dan drag and drop asset nya ke frame.



Gambar 9.20 Clip *Player_jump*

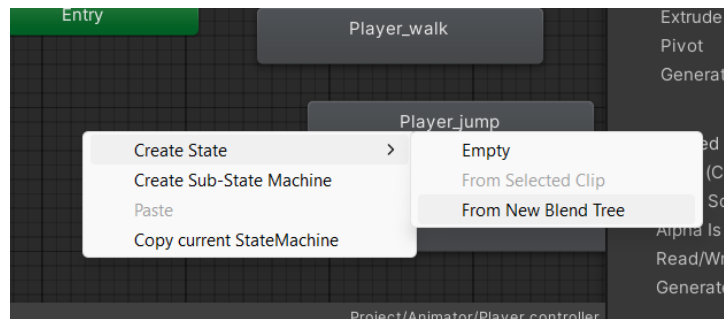


24. Buat clip baru lagi dan beri nama `player_fall` dan drag and drop asset nya ke frame.



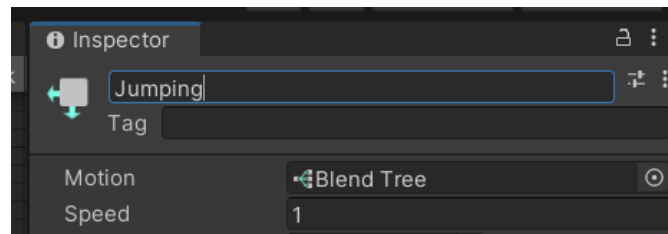
Gambar 9.21 *Clip Player_fall*

25. Di window animator, klik kanan kemudian *create state* seperti dibawah ini.



Gambar 9.22 *Create State*

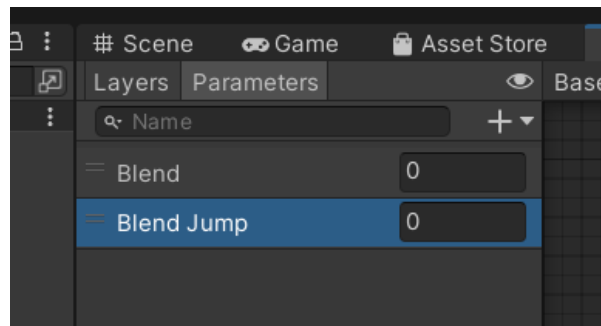
26. ubah namanya menjadi jumping pada bagian inspector.



Gambar 9.23 Mengubah nama *Blend_tree*

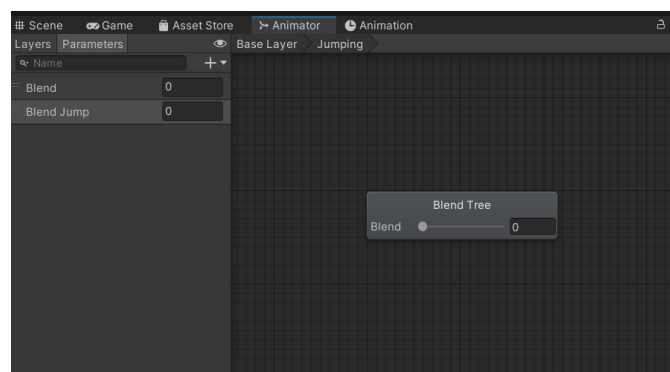


27. buat parameters baru dengan tipe data float dan beri nama Blend Jump.



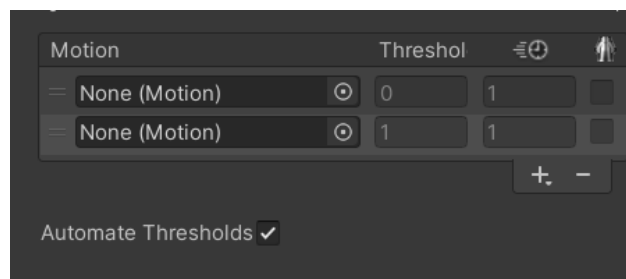
Gambar 9.24 *Blend Jump*

28. Klik 2 kali pada blend jumping.



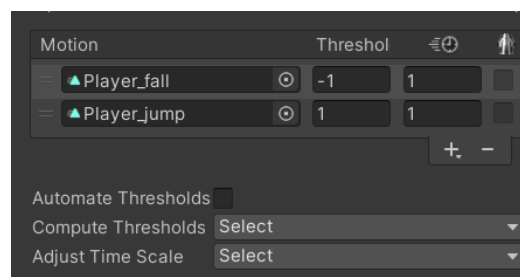
Gambar 9.25 *Layer Jumping*

29. Tambahkan dua motion field seperti berikut.



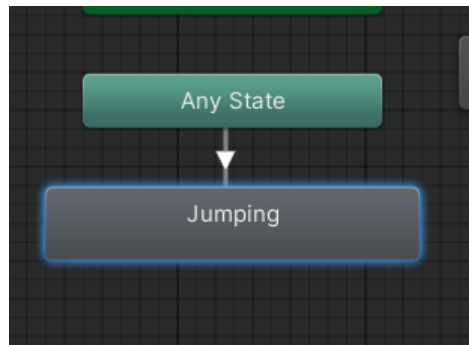
Gambar 9.26 *Motion Field*

30. Atur nilai *motion* nya seperti dibawah ini.



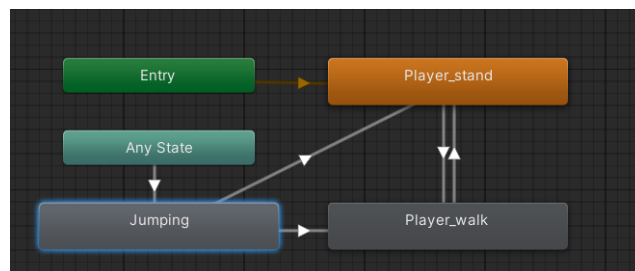
Gambar 9.27 Mengatur Nilai *Motion Field*

31. Buat transition dari any state ke jumping.



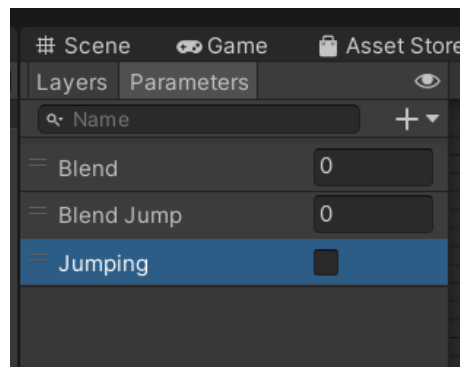
Gambar 9.28 Buat Transisi Baru

32. Buat juga transition dari jumping ke player_stand dan player_walk.



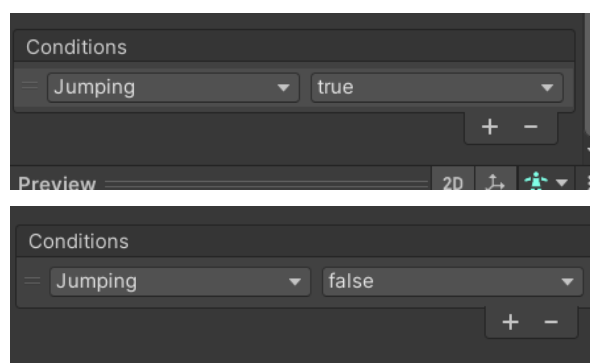
Gambar 9.29 Transisi Baru

33. Buat parameters baru dengan tipe data Boolean dan beri nama Jumping.



Gambar 9.30 Parameters *Jumping*

34. Klik arah panah jumping ke player_stand dan player_walks, lalu ubah bagian conditions seperti berikut. Kemudian, atur bagian settings seperti langkah ke-16.



Gambar 9.31 Mengatur *Conditions*



35. Tambahkan source code berikut untuk void Update().

```
if (Input.GetButtonDown("Jump")) {  
    animator.SetBool("Jumping", true);  
    jump = true;  
}
```

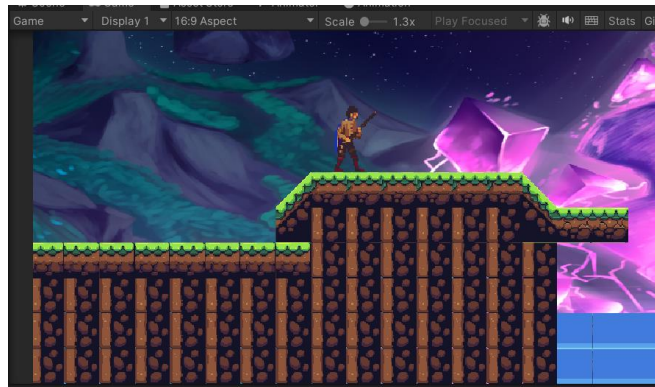
36. Tambahkan source code berikut pada void FixedUpdate()

```
animator.SetFloat("Blend Jump", rb.velocity.y);
```

37. Tambahkan source code berikut pada void GroundCheck()

```
animator.SetBool("Jumping", !isGrounded);
```

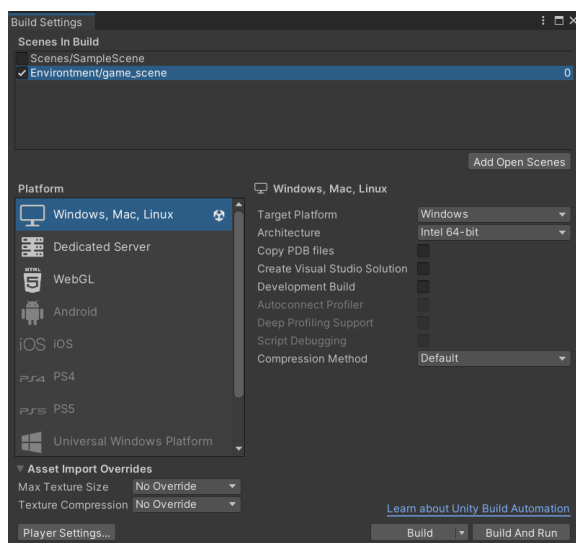
38. Coba jalankan dan lihat hasilnya.



Gambar 9.32 Hasil Akhir

B. Render

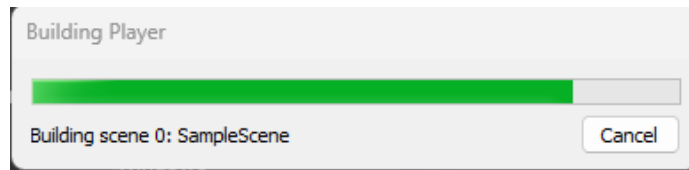
1. Untuk proses merender pada file kemudian pilih Build Settings atau menekan Ctrl + Shift + B, setelah masuk Setting Build pilih PC, Mac dan Linux. Jangan lupa pada scenes in build mencentang project kita, kalo belum ada tekan Add Open Scenes.



Gambar 9.26 Render Settings



2. Setelah itu kita klik Build dan kita pilih project jadinya akan disimpan dimana.
Dan tunggu hasilnya.



Gambar 9.27 Building Render

C. Link Pengumpulan Github



KUIS

KUIS PERTEMUAN 9 👍

```
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);
    }
}
```

Analisa :

Untuk penjelasan kode diatas yaitu pada baris `animator.SetBool("isJumping",);` dan `animator.SetBool("isJumping",);`, nilai boolean tidak diberikan. Fungsi `SetBool` membutuhkan dua argumen: nama parameter dan nilai boolean, sehingga kita harus memberikan nilai `true` atau `false`. Selain itu, penggunaan `rb.AddForce(Vector2.up * jumpForce, ForceMode2D.Impulse);` salah karena variabel `jumpForce` tidak didefinisikan dalam. Seharusnya menggunakan `jumpPower`, sesuai dengan deklarasi variabel sebelumnya. Terakhir, terdapat kesalahan logika di `HandleMovementInput`, di mana kondisi `else if (move > 0)` tidak akan pernah tercapai karena kondisi `if (move != 0)` sudah menangani semua kasus di mana `move` tidak sama dengan 0. Seharusnya, kita menggunakan `if` terpisah untuk menangani `move > 0` dan `move < 0`.