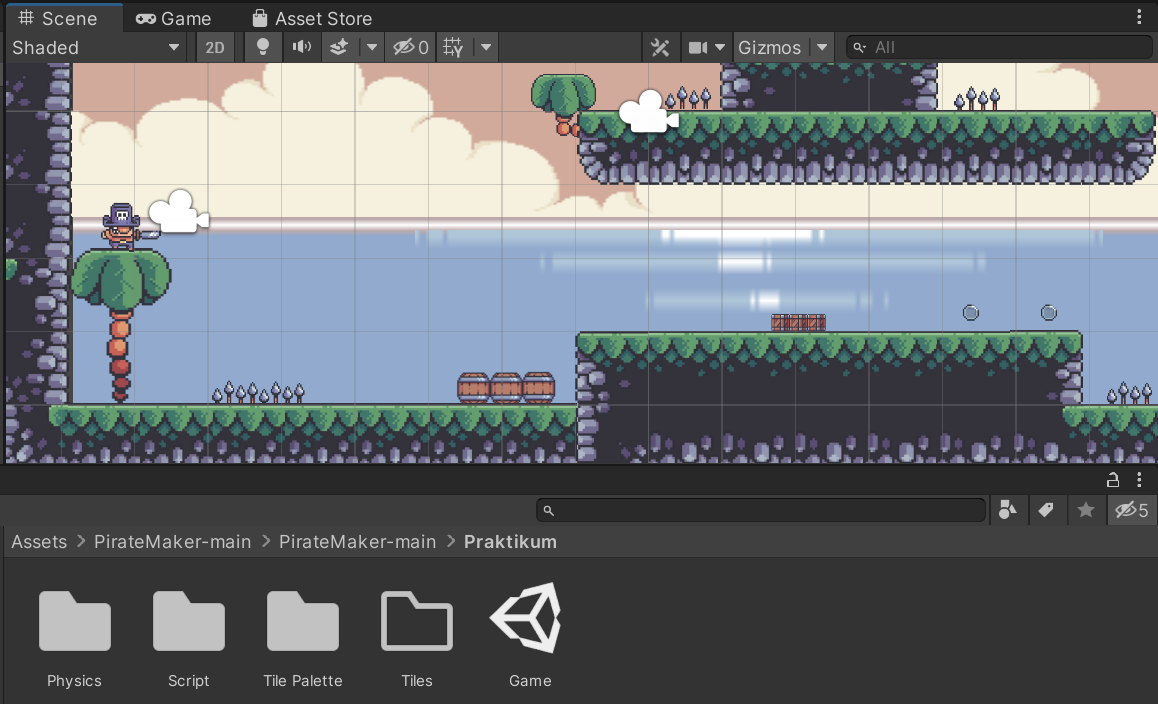
# 9 GAME ANIMATION

|  |  |  |
| --- | --- | --- |
| **NIM** | : | 2118109 |
| **Nama** | : | Ahmad Alif Al Ayubi |
| **Kelas** | : | D |
| **Asisten Lab** | : | Zain Aryanta (2118051) |

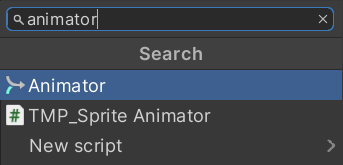
## Tugas 1 : Membuat Character Animation

1. Buka kembali projek sebelumnya yang sudah dapat digerakkan dengan *keyboards*.



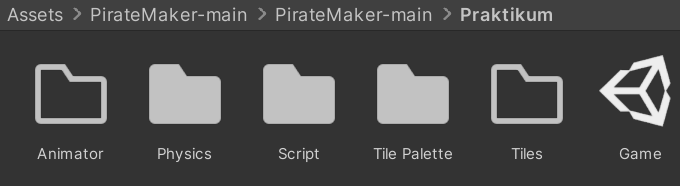
### 9.1 Membuka Projek Unity

1. Kemudian klik karakternya lalu pada *inspector* tambahkan *component* *Animator*.



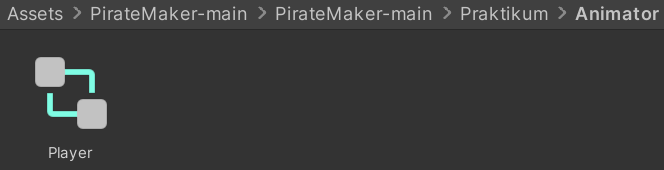
### 9.2 Menambahkan Component Animator

1. Tambahkan *folder* baru pada *folder* “Praktikum” berikan nama *folder* dengan nama “*Animator*”.



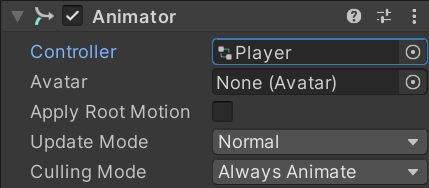
### 9.3 Menambahkan Folder Baru

1. Tambahkan *file* “*Animator Controller*” pada *folder* “*Animator*” dengan cara klik kanan pada *folder* lalu *Create* > *Animator Controller*, ubah nama menjadi “*Player*”.



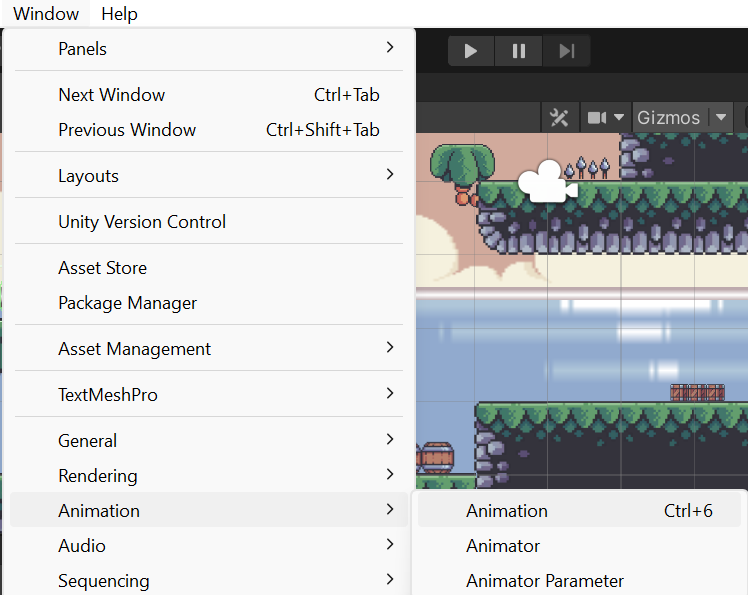
### 9.4 Menambahkan File Animator Controller

1. Lalu klik pada karakter bagian *Component Animator*, pada Controller ubah menjadi *Player*.



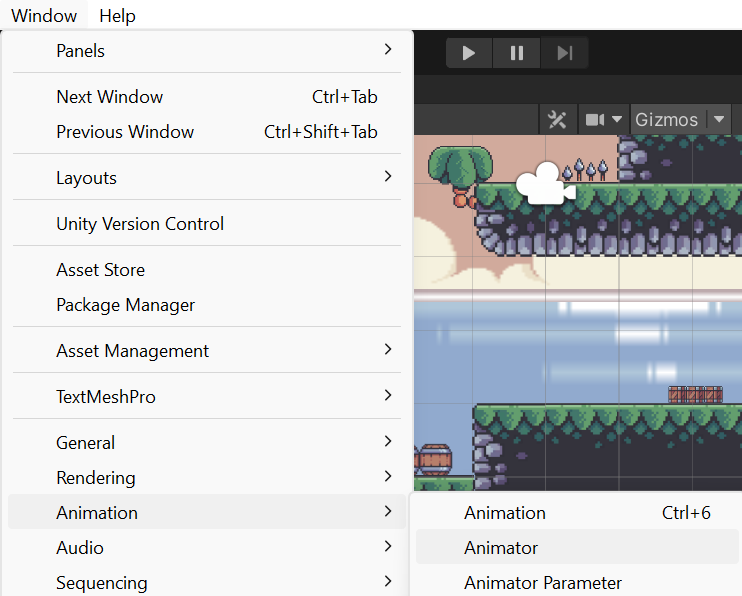
### 9.5 Setting Controller

1. Tambahkan *menu panel Animation* dengan cara klik *menu Window* > *Animation* > *Animation*.



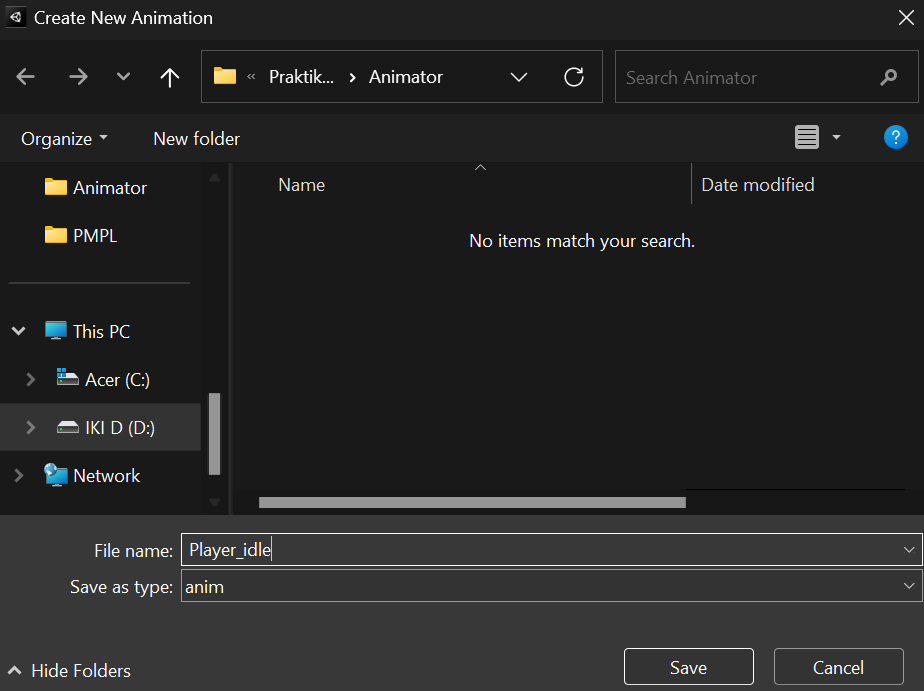
### 9.6 Menambahkan Window Animation

1. Tambahkan lagi *Window* baru dengan cara klik *menu Window* > *Animation* > *Animator.*



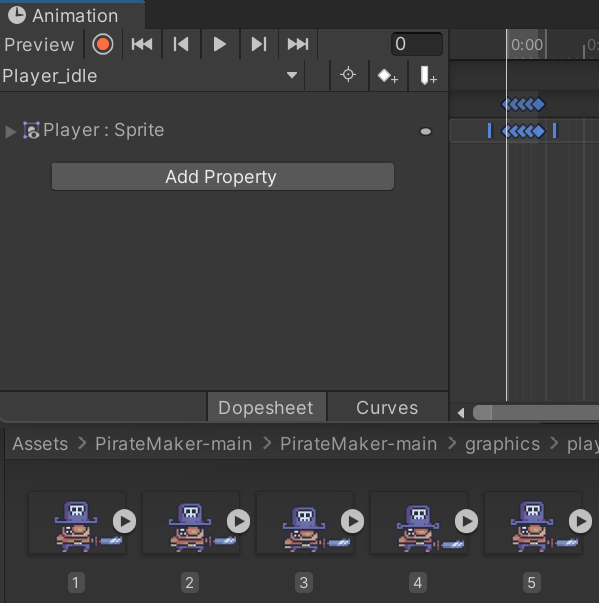
### 9.7 Menambahkan Window Animator

1. Klik Create pada Animation lalu simpan pada folder animator dan beri nama “Player\_idle”.



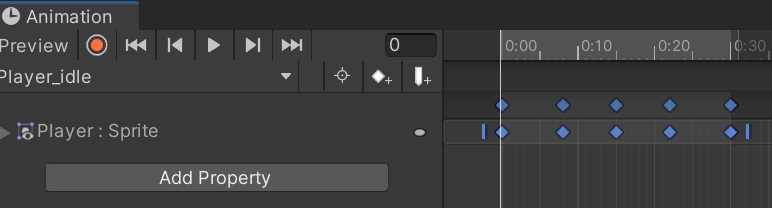
### 9.8 Menyimpan File

1. Kemudian masukkan *asset player idle*, masukkan semua gambar *player* *idle* ke *tab Animation*.



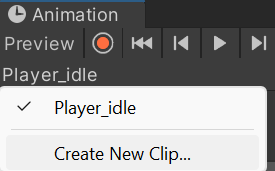
### 9.9 Meletakkan Asset Iddle

1. Lalu tekan Ctrl + A pada *tab Animation* kemudian geser kotak ke *frame* 30.



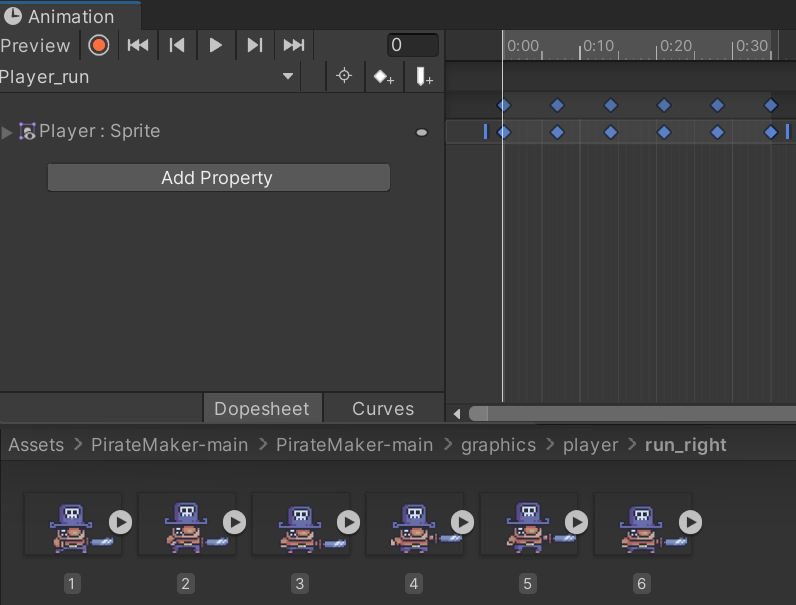
### 9.10 Menambahkan Frame

1. Tambahkan animasi bari dengan klik pada “Player\_idle” lalu pilih *Create New Clip*, beri nama “Player\_run” dan simpan pada *folder* *Animator*.



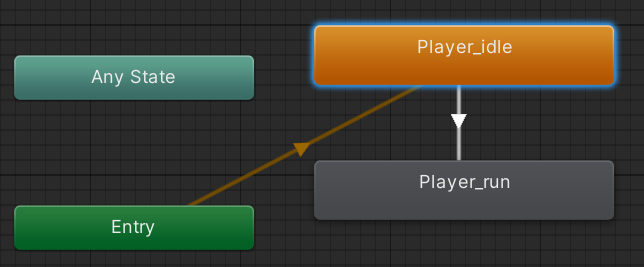
### 9.11 Menambahkan Animasi

1. Selanjutnya tambahkan *asset run*, masukkan semua gambar animasi *run* dan masukkan ke *tab Animation*, lalu geser frame sampai frame 35.



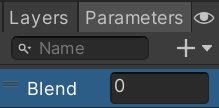
### 9.12 Menambahkan Animasi Run

1. Lalu pada *menu Animator*, buat transisi antara Player\_idle dan Player\_run dengan klik kanan pada Player\_idle pilih *Make Transition* lalu tarik ke Player\_run.



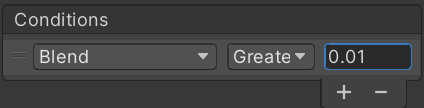
### 9.13 Membuat Transisi

1. Lalu pada *tab parameter* tambahkan tipe data *Float* dengan menekan *icon* tambah dan rubah namanya menjadi “Blend”.



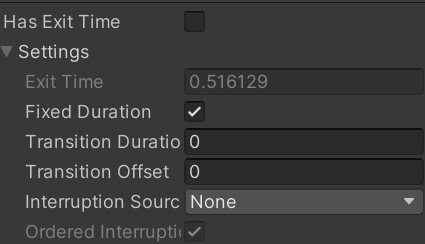
### 9.14 Menambahkan Parameter

1. Klik panah putih antara Player\_idle dan Player\_run, Kemudian pada bagian *conditions* klik *icon* tambah atur menjadi “Blend” dan beri nilai *conditions* menjadi 0.01.



### 9.15 Menambahkan Conditions

1. Pada bagian *settings* hilangkan centang pada *Has Exit Time* dan beri nilai *Transition Duration* menjadi 0.



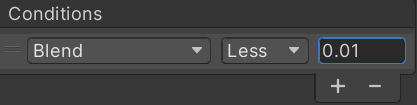
### 9.16 Mengatur Settings

1. Lalu buat transisi kembali dari Player\_run ke Player\_idle dengan cara yang sama.



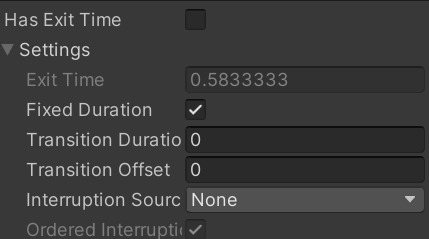
### 9.17 Membuat Transisi

1. Tambahkan lagi *parameter* dengan tipe *data* *float*, ubah nama menjadi “Blend” lalu rubah operator menjadi *Less* dan berikan nilai 0.01.



### 9.18 Menambahkan Parameter Baru

1. Pada bagian *settings* hilangkan juga centang pada *Has Exit Time* dan atur nilai *Transition Duration* menjadi 0.



### 9.19 Mengatur Setting

1. Lalu rubah *code* pada *class Player* dengan *code* dibawah ini.

|  |
| --- |
| Public Animator animator; |

Analisa

Pada *code* diatas merupakan pendeklarasian sebuah variabel publik dengan tipe data *Animator* yang bernama *animator*. Tipe data *Animator* merupakan komponen yang mengontrol animasi dari sebuah *GameObject*.

1. Tambahkan juga *code* dibawah ini pada *void Awake*().

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

Analisa

Pada *code* diatas digunakan untuk mencari komponen bernama "*Animator*" yang terdapat pada *GameObject* dan akan disimpan dengan variabel *animator* yang digunakan untuk mengontrol animasi dari *GameObject*.

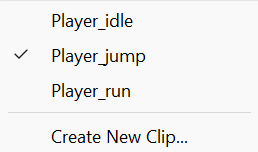
1. Tambahkan *code* dibawah ini pada *void FixedUpdate()*.

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

Analisa

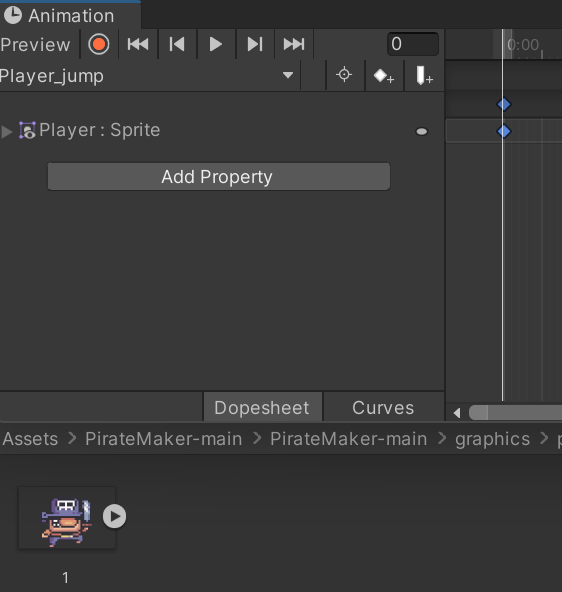
Pada *code* diatas digunakan untuk mengatur *parameter* "*Blend*" pada "*Animator*" dengan nilai absolut dari kecepatan *horizontal* x dari "*Rigidbody2D*" yang digunakan untuk mengatur animasi berjalan dan berlari.

1. Kemudian tambahkan animasi baru dengan nama “Player\_jump” dan letakkan pada folder *animator*.



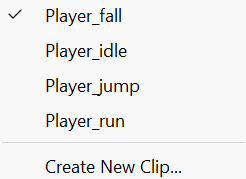
### 9.20 Menambahkan Animasi

1. Lalu tambahkan *asset* melompat pada *folder* lalu letakkan pada *tab Animation*.



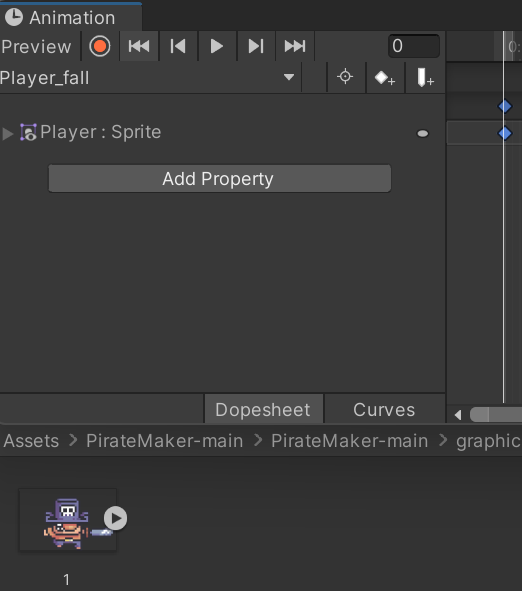
### 9.21 Meletakkan Asset Melompat

1. Kemudian buat animasi baru lagi dengan nama “Player\_fall” dan simpan pada *folder Animator*.



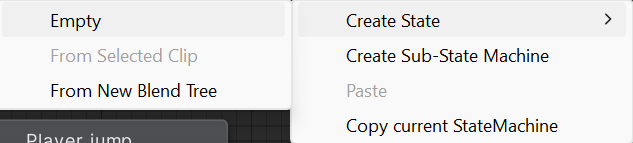
### 9.22 Menambahkan Animasi Fall

1. Lalu tambahkan *asset player fall* pada *tab Animation*.



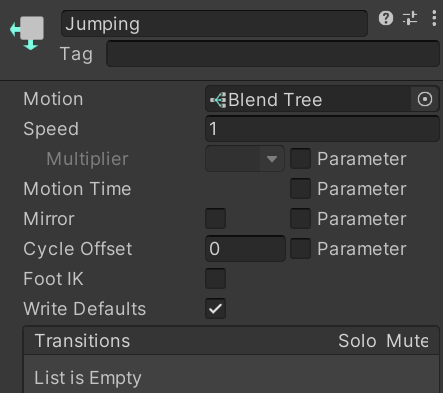
### 9.23 Menambahkan Asset Fall

1. Lalu klik kanan pada *menu Animator* lalu pilih *Create State* > *From New Blend Tree*.



### 9.24 Membuat Animasi Jump

1. Lalu klik *Blend Tree*, rubah namanya pada *inspector* menjadi *Jumping*.



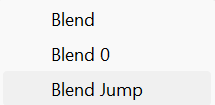
### 9.25 Merubah Nama Blend Tree

1. Tambahkan *parameter* baru dengan tipe data *Float* lalu rubah namanya menjadi “*Blend Jump*”.



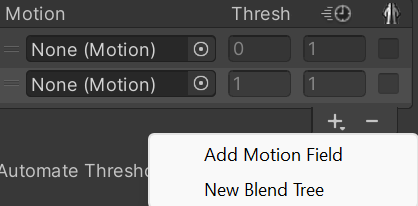
### 9.26 Menambahkan Parameter

1. Klik 2 kali pada *Blend Tree* “*Jumping*” lalu pada *inspector* ubah parameternya menjadi “*Blend Jump*”.



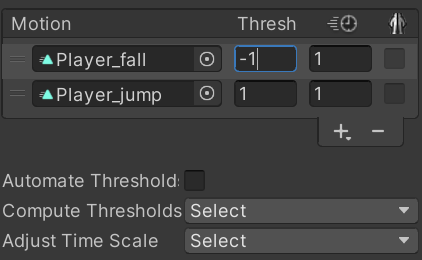
### 9.27 Merubah Parameter

1. Tambahkan dua *Motion Field* dengan menekan *icon* tambah dan pilih *Add Motion Field*.



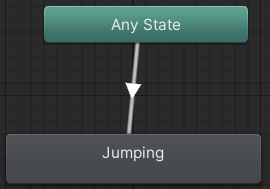
### 9.28 Menambahkan Motion Field

1. Pada bagian *None* tambahkan “Player\_fall” dan “Player\_jump”, Kemudian hilangkan centang pada “*Automate Thresholds*” dan berikan nilai seperti dibawah.



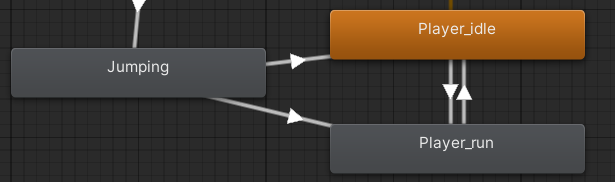
### 9.29 Menambahkan Nilai Threshold

1. Pada *Base Layer* klik kanan pada *Any State* pilih *Make Transition* dan arahkan pada *Jumping*.



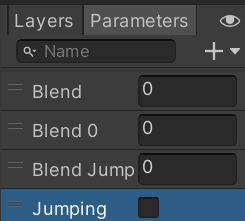
### 9.30 Menghubungkan Dengan Jumping

1. Lalu hubungkan *Jumping* dengan Player\_idle dan Player\_run.



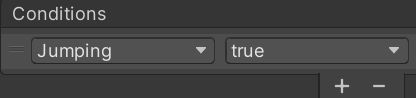
### 9.31 Menghubungkan Jumping

1. Tambahkan *parameter* dengan tipe data *Bool* dan rubah namanya menjadi *Jumping*.



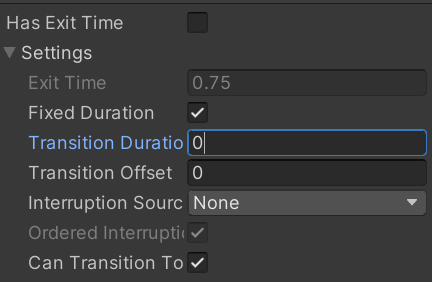
### 9.32 Menambahkan Parameter Jumping

1. Klik panah yang mengarah ke *Jumping*, pada *inspector* tambahkan *condition Jumping* dan rubah nilainya menjadi *True*.



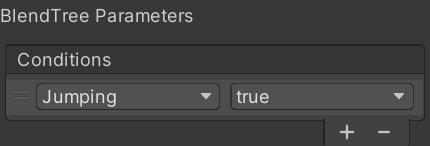
### 9.33 Menambahkan Conditions Jumping

1. Lalu pada *Settings* hapus centang pada *Has Exit Time* dan rubah nilai *Transition Duration* menjadi 0.



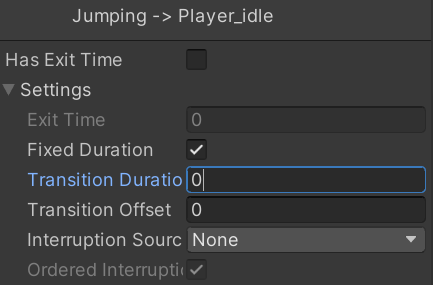
### 9.34 Mengatur Settings

1. Lalu tambahkan *condition Jumping* pada panah Player\_idle dan Player\_run, kemudian untuk Player\_idle menjadi *false* dan Player\_run menjadi *true*.



### 9.35 Menambahkan Conditions Jumping

1. Lalu pada *Setting* Player\_idle dan Player\_run berikan nilai 0 untuk *Transition Duration* dan hapus centang pada *Has Exit Time*.



### 9.36 Mengatur Setting

1. Lalu pada *script Player* tambahkan *code* berikut pada fungsi *Update*.

|  |
| --- |
| animator.SetBool("Jumping", true); |

Analisa

Pada *code* diatas digunakan untuk memeriksa apakah tombol yang digunakan untuk melompat telah ditekan atau belum. Jika iya maka akan mengatur parameter "*Jumping*" pada "*Animator*" ke kondisi *true* dan mengatur "*Jump*" ke *true*.

1. Tambahkan code berikut pada pada fungsi FixedUpdate

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

Analisa

Pada *code* diatas digunakan untuk mengatur *parameter* "*Blend* *Jump*" pada "*Animator*" dengan nilai kecepatan vertikal y dari *Rigidbody2D* yang digunakan untuk mengatur animasi lompat atau jatuh.

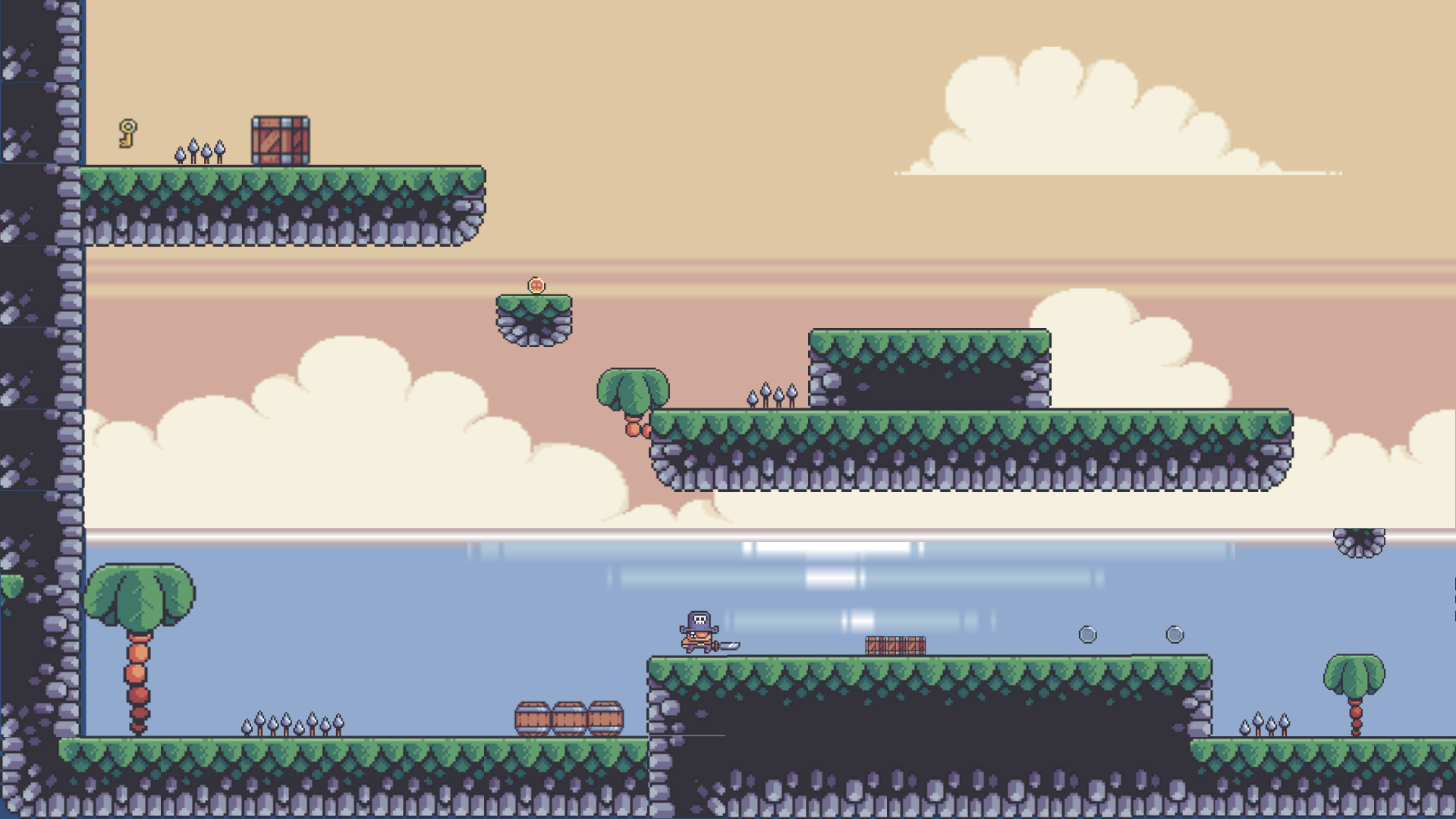
1. Kemudian tambahkan *code* berikut pada *method* GroundCheck().

|  |
| --- |
| animator.SetBool("Jumping", !isGrounded); |

Analisa

Pada *code* diatas digunakan untuk mengatur *parameter* "*Jumping*" pada "*Animator*" ke *true* jika pemain tidak menyentuh tanah dan *false* jika pemain menyentuh tanah.

1. Kemudian lakukan *test* dengan melakukan render terlebih dahulu, setelah itu buka projeknya.



### 9.37 Melakukan Uji Coba

## Kuis

Code masih terdapat kesalahan

|  |
| --- |
| 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);  }  } |

Setelah diperbaiki

|  |
| --- |
| 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);  }  } |

Analisa

Pada code yang salah yang terletak di fungsi *HandleJumpInput* yang menggunakan kondisi *if else*, pada kondisi *if* tidak terdapat *parameter* yang bernilai *true* atau *false*, dengan melakukan perbaikan maka Ketika tombol *space* ditekan maka akan bernilai *true* maka(“*isJumping*”, *true*) kemudian ditambahkan kondisi jika tombol *space* tidak ditekan dengan menambahkan kondisi else(“*isJumping*”, *false*).

Lalu pada fungsi *HandleMovementInput* dengan kondisi *if* (!=1) dirubah menjadi (==0) karena kondisi tersebut memberikan perintah karakter tidak digerakkan, maka perintah dibuat menjadi (“*isIdle*”, *true*) kemudian (“*isWalking*”, *false*). Untuk *transform*.*Translate*() seharusnya diletakkan pada kondisi jika karakter digerakkan yaitu kondisi *else* karena kondisi (ini merupakan perinah jika karakter digerakkan. Maka untuk kondisi *else* tidak adanya perintah yang valid, pada perintah tersebut harusnya terdapat (“*isIdle*”, *false*) (“*isWalking*”, *true*) dan *transform*.*Translate*(*Vector3.right \* move \* Time.deltaTime*) dan kenapa *Vector3.right* yang awalnya *Vector3.left*, karena jika left maka pergerakan karakter tidak sama dengan arah yang sebenarnya atau terbalik. Kemudian untuk kondisi *if* yang kedua harusnya *move0* itu juga *parameter vector3* nya harus bernilai sama maka *Vector3*(1, 1, 1) untuk karakter menghadap ke kanan.