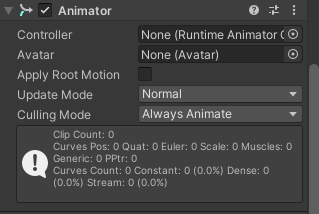
# 9 Game Animation

|  |  |  |
| --- | --- | --- |
| **NIM** | : | 2118018 |
| **Nama** | : | Syalsia Fatiha Yunkanabilla |
| **Kelas** | : | A |
| **Asisten Lab** | : | Natasya Octavia (2118034) |

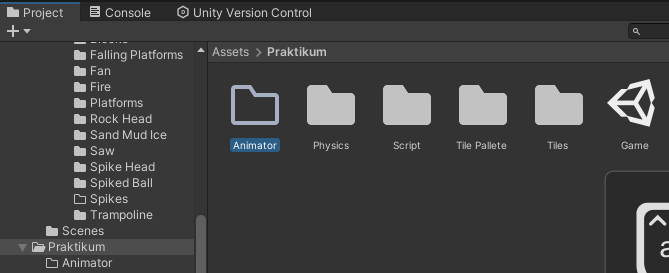
## 1.1 Tugas 1 : Membuat Character Animation

1. **Character Animation**
2. Buka aplikasi unity yang telah dibuat sebelumnya pada bab8 kemudian pada karakter player pergi ke inspector > add component dan tambahkan animator.



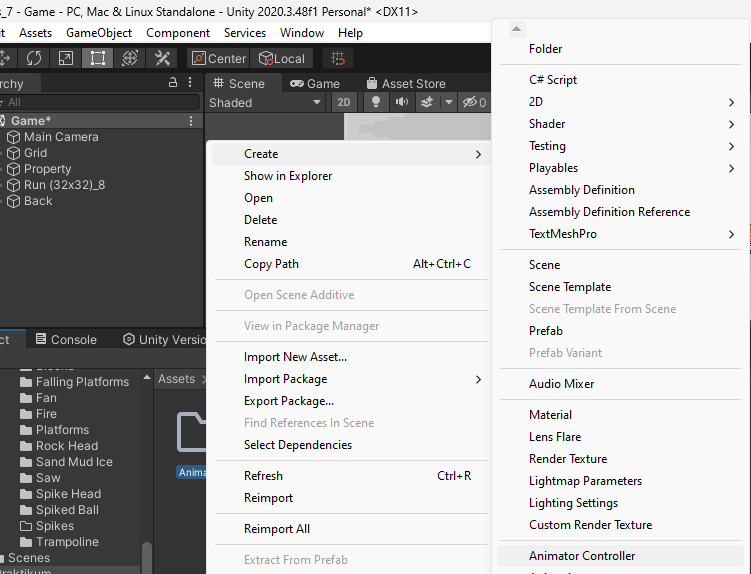
### 9.1 Tampilan Animator

1. Kemudian membuat folder animator pada folder praktikum dengan klik kanan pada praktikum kemudian pilih create folder.



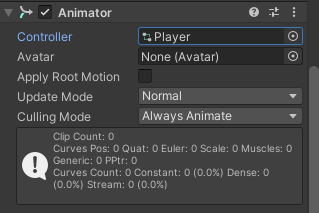
### 9.2 Folder Animator

1. Langkah selanjutnya, membuat Animator Controller pada folder Animator dengan cara klik kanan pada folder Animator pilih create > Animator Controller.



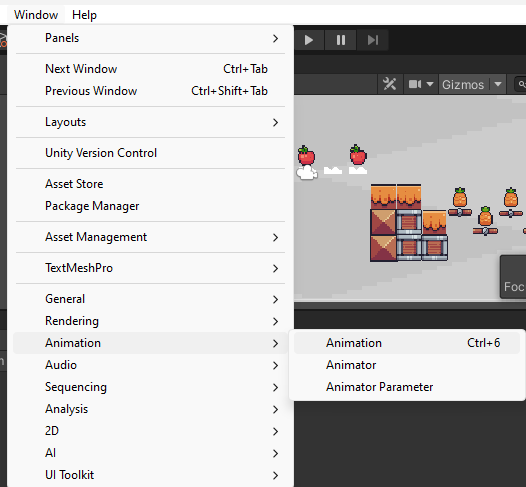
### 9.3 Create Animator Controller

1. Kemudian pada hierarcy player pergi ke inspector, pada Animator pilih setting Controller dan pilih Player.



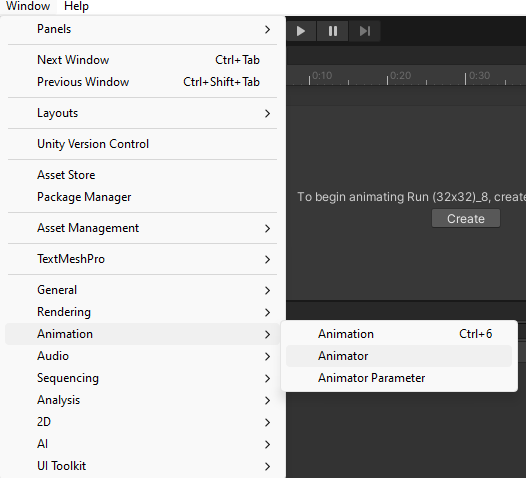
### 9.4 Settings Controller

1. Kemudian menambahkan menu panel animation dengan cara klik window kemudian pilih animation > animation.



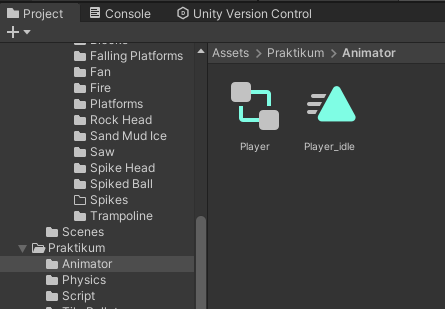
### 9.5 Panel Animation

1. Kemudian menambahkan menu panel animator dengan cara klik window kemudian pilih animation > animator.



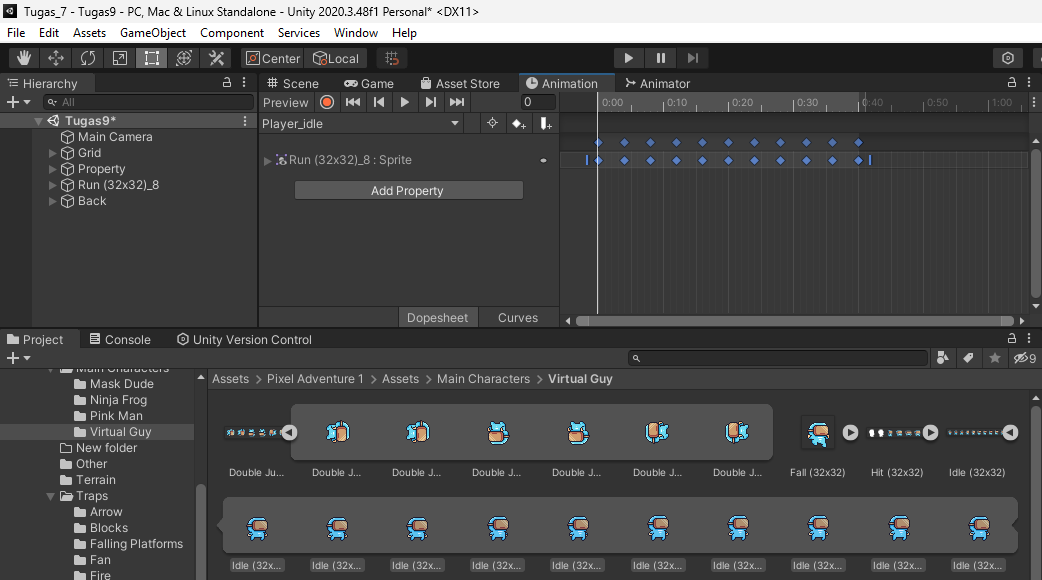
### 9.6 Panel Animator

1. Selanjutnya membuat animasi pada player idle dengan klik create pada panel animation kemudian simpan pada folder animator dengan nama player\_idle.



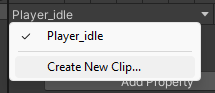
### 9.7 Player\_idle

1. Kemudian buka folder virtual guy pada player pilih idle kemudian tekan shift untuk semua karakter player idle dan drag & drop ke dalam tab animation serta geser kotak kecil hingga frame 40.



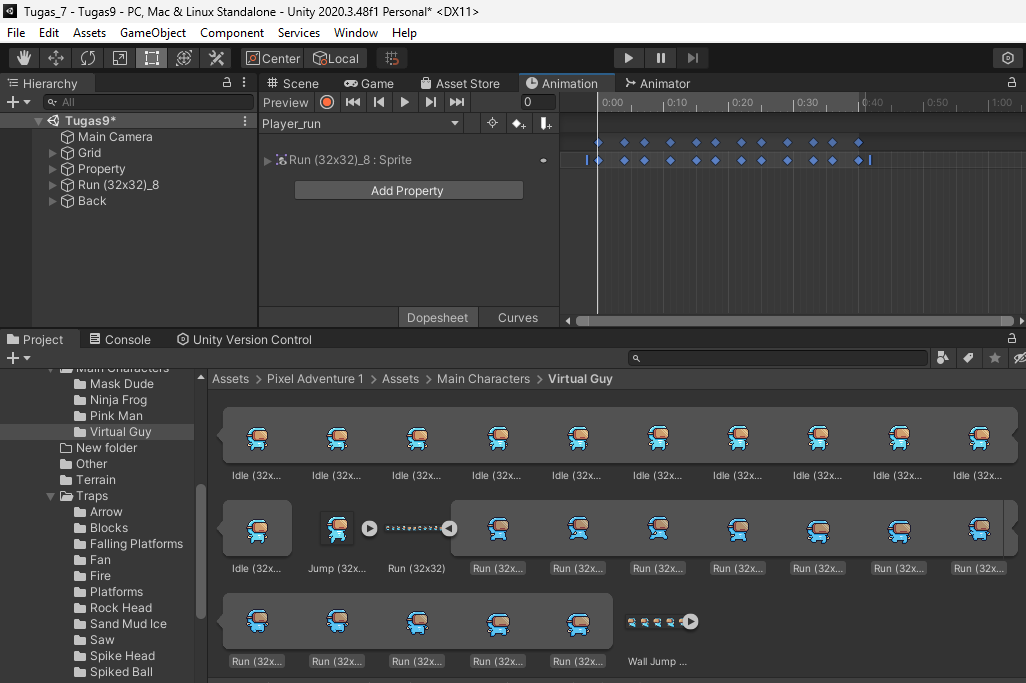
### 9.8 Drag & Drop Player\_idle

1. Setelah itu untuk membuat animasi baru caranya adalah pada Player\_idle kemudian klik segitiga dan pilih Create New Clip.



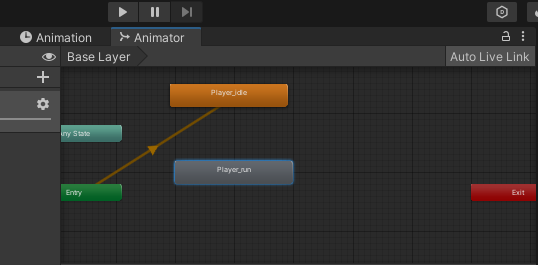
### 9.9 Create New Clip

1. Kemudian lakukan langkah yang sama pada point ke 8 dengan nama Player\_run, drag & drop player\_run ke dalam tab animation dan geser hingga frame 40.



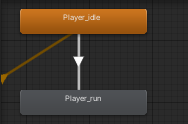
### 9.10 Player\_run

1. Selanjutnya cek pada tampilan tab animator maka tampilannya akan seperti berikut.



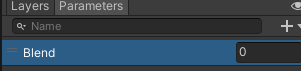
### 9.11 Tab Animator

1. Kemudian klik kanan pada Player\_idle pilih make transition dan arahkan panah ke Player\_run



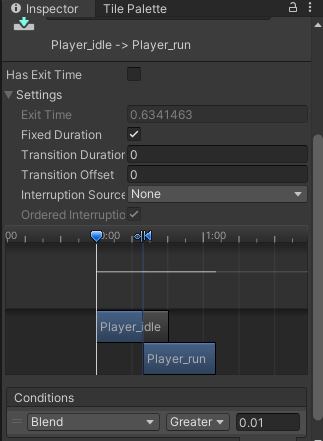
### 9.12 Transition Player\_idle to Player\_run

1. Lalu pergi ke tab parameter kemudian tambahkan tipe data float kemudian ubah namanya menjadi Blend.



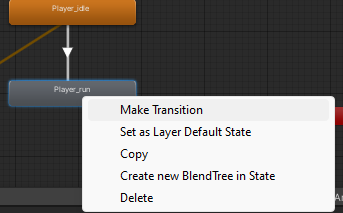
### 9.13 Create Blend

1. Kemudian pada condition pilih blend dengan greater nilai kondisinya 0.01 kemudian uncheck pada checklist Has Exit Time dan buat Transition Duration menjadi 0.



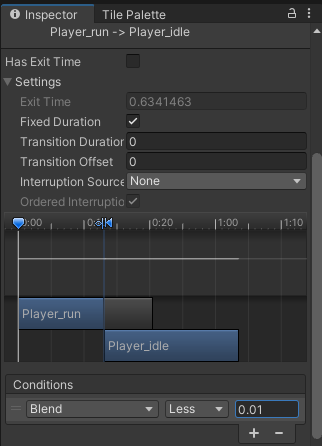
### 9.14 Condition Greater

1. Langkah selanjutnya adalah membuat make transition kebalikannya yaitu dari Player\_run ke Player\_idle.



### 9.15 Make Transition Player\_run to Player\_idle

1. Kemudian pada condition pilih blend dengan less yang nilai kondisinya 0.01 kemudian uncheck pada checklist Has Exit Time dan buat Transition Duration menjadi 0.



### 9.16 Condition Less

1. Kemudian tambahkan source code berikut agar animasi dapat sesuai ketika berjalan.

|  |
| --- |
| public class Player : MonoBehaviour  {  public Animator animator; |

1. Kemudian tambahkan source code berikut pada script komponen animator.

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

1. Langkah selanjutnya adalah menambahkan source code berikut pada fungsi FixedUpdate.

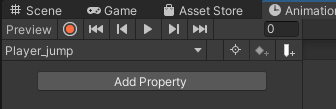
|  |
| --- |
| void FixedUpdate()  {  GroundCheck();  Move(horizontalValue, jump);  animator.SetFloat("Blend", Mathf.Abs(rb.velocity.x));  } |

1. Kemudian game dapat dijalankan dengan klik play maka player akan memiliki animasi ketika berhenti maupun ketika berjalan.



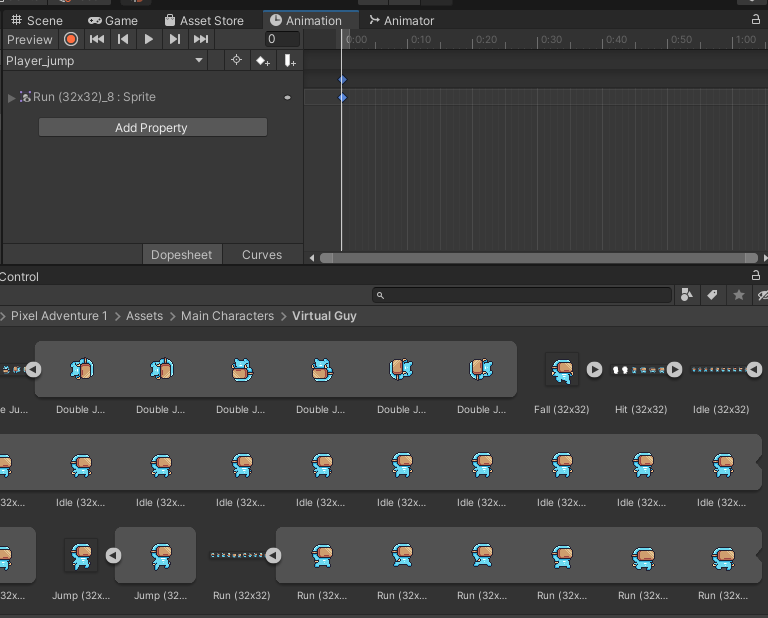
### 9.17 Tampilan Animasi

1. Kemudian pada membuat animasi baru dengan Create New Clip dengan nama Player\_jump.



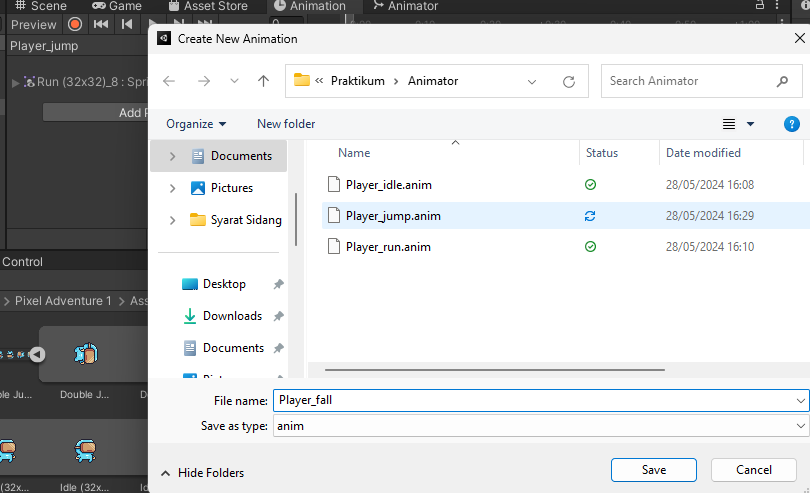
### 9.18 Create Player\_jump

1. Kemudian lakukan langkah yang sama dengan membuka folder Virtual Guy dengan mendrag & drop karakter jump ke dalam tab animation.



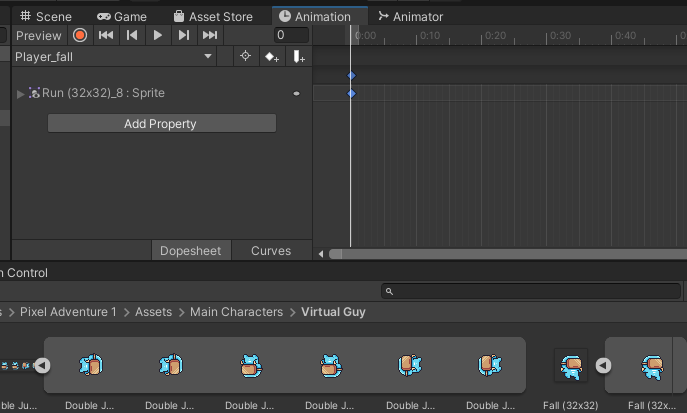
### 9.19 Drag & Drop Player\_jump

1. Kemudian simpan ke dalam folder praktikum > animator dengan nama Player\_jump.



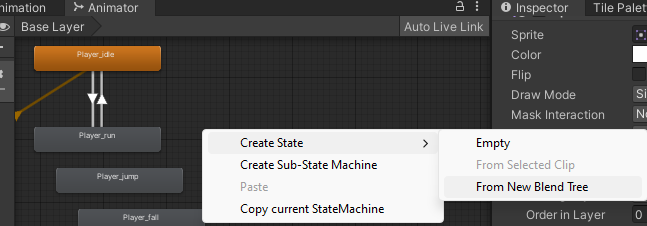
### 9.20 Folder Animator

1. Sama halnya dengan langkah sebelumnya yaitu dengan membuat Player\_fall yang kemudian mendrag & drop karakter Player\_fall ke dalam tab animation.



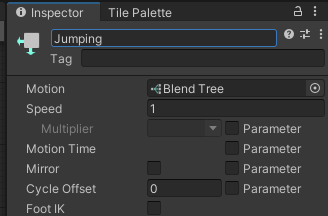
### 9.21 Tampilan Player\_fall

1. Langkah selanjutnya untuk menambahkan animasi ketika melompat, maka klik kanan pada tab animator kemudian Create State > pilih Empty.



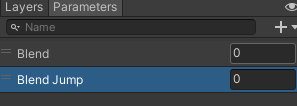
### 9.22 Create State Empty

1. Kemudian pada animation Blend Tree double klik menuju inspector dan ubah namanya menjadi Jumping .



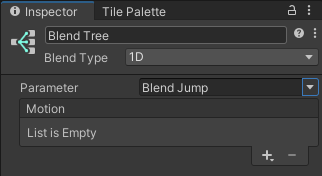
### 9.23 Tampilan Blend Tree

1. Kemudian pada menu parameter tambahkan parameter dengan tipe data Float dan ubah dengan nama Blend Jump.



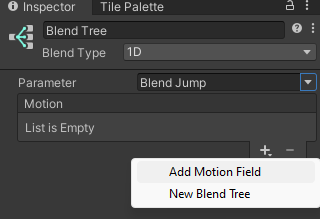
### 9.24 Tampilan Blend Jump

1. Selanjutnya pada inspector Blend Tree “Jumping” ubah parameter menjadi Blend Jumptekan play untuk mengecek apakah karakter player dapat melompat atau belum.



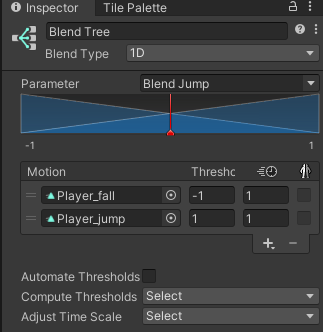
### 9.25 Tampilan Parameter Blend Tree

1. Kemudian pada motion tambahkan Add Motion Field klik tambah 2x untuk menambahkan Motion Field.



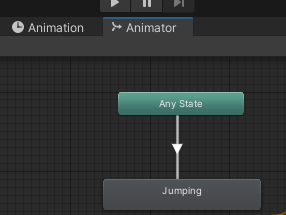
### 9.26 Add Motion Field

1. Kemudian pada Motion Field ubah menjadi Player\_fall dan Player\_jump kemudian sesuaikan Thresholdnya seperti tampilan berikut.



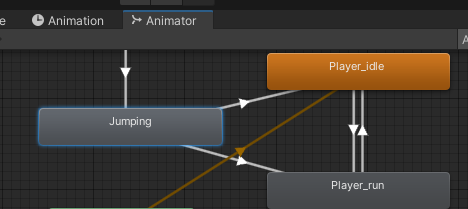
### 9.27 Motion Field

1. Kemudian pada based layer klik kanan pada Any State kemudian make transition ke Jumping.



### 9.28 Make Transition Any State to Jumping

1. Kemudian klik kanan pada jumping dan make transition ke Player\_idle dan Player\_run.



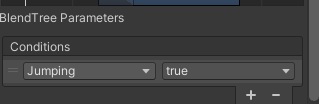
### 9.29 Make Transition Jumping to Player\_idle & Player\_run

1. Kemudian tambahkan parameter lagi dengan tipe data Boolean dan ubah namanya menjadi jumping.



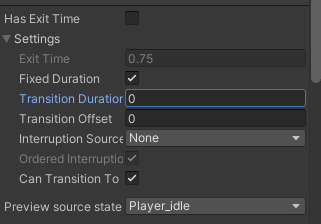
### 9.30 Tampilan Tipe Data Jumping

1. Kemudian klik anak panah jumping kemudian pada inspector tambahkan conditions jumping dan ubah menjadi true.



### 9.31 Condition Jumping

1. Kemudian ubah setting menjadi unchecklist pada Has Exit Time kemudian Transition Duration menjadi 0.



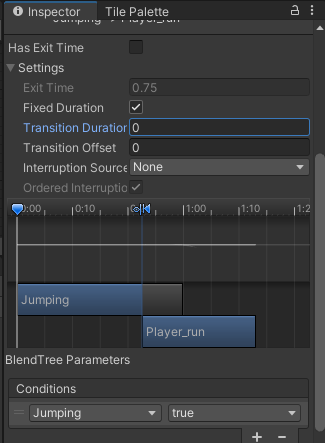
### 9.32 Tampilan Setting Jumping

1. Kemudian pada arah panah Player\_idle unchecklist Has Exit Time, Transition Duration menjadi 0 dan berikan condition jumping menjadi false.



### 9.33 Condition Player\_idle False

1. Kemudian pada arah panah Player\_run unchecklist Has Exit Time, Transition Duration menjadi 0 dan berikan condition jumping menjadi true.



### 9.34 Condition Player\_run False

1. Langkah selanjutnya adalah tambahkan script pada player dengan source code berikut pada fungsi Update.

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

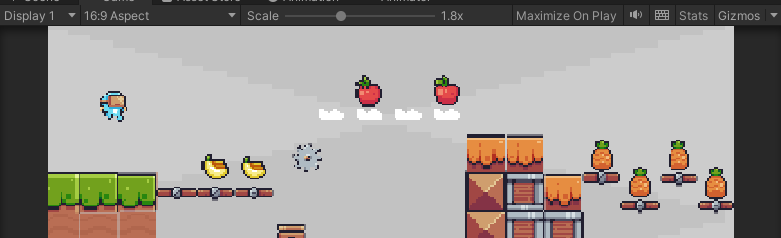
1. Selanjutnya tambahkan juga script pada fungsi FixedUpdate dengan source code berikut ini.

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

1. Kemudian tambahkan source code pada GroundCheck untuk melakukan jumping dengan code berikut.

|  |
| --- |
| void GroundCheck()  {  isGrounded = false;  Collider2D[] colliders = Physics2D.OverlapCircleAll(groundcheckCollider.position, groundCheckRadius, groundLayer);  if (colliders.Length > 0){  isGrounded = true;  }  animator.SetBool("Jumping", !isGrounded) ;  } |

1. Kemudian tekan play untuk mengecek apakah karakter player dapat bergerak dengan animasi atau belum.



9.35 Tampilan Animasi

1. Langkah terakhir adalah merender hasil tampilan game yang sudah dibuat dengan animasi game tersebut.



### 9.36 Hasil Tampilan Render