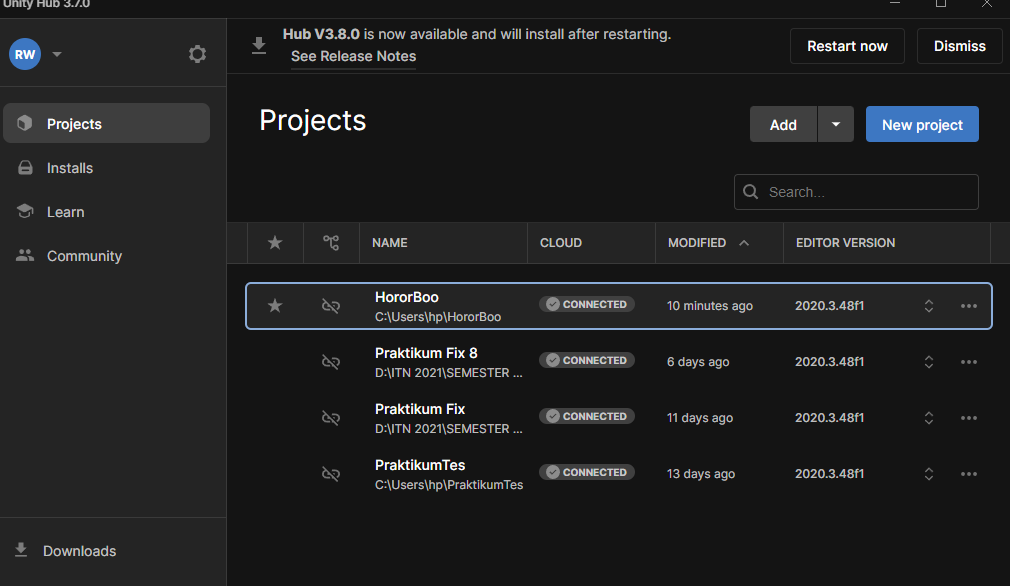
# 9 Character Animation

|  |  |  |
| --- | --- | --- |
| **NIM** | : | 2118055 |
| **Nama** | : | Ridho Arif Wicaksono |
| **Kelas** | : | B |
| **Asisten Lab** | : | Difa Fisabililah (2118052) |

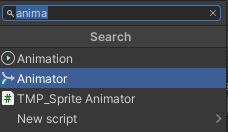
## 1.1 Tugas : Character Animation

1. Buka unity hub lalu klik pada project sebelumnya



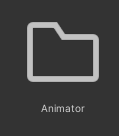
Gambar 9. 1 Project Baru

1. Setelah itu klik pada karakter lalu masuk kedalam abgian inspector, dan klik add component tulis “Animator”



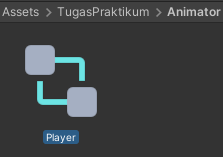
Gambar 9. 2 Menambah Komponen Animator

1. Pada folder Tugas Praktikum buat folder baru bernama “Animator” untuk menyimpan file animasi



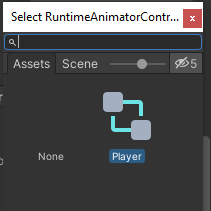
Gambar 9. 3 Folder Animator

1. Membuat file animator controller dengan cara klik kanan create->Animator Controller buat animator bernama “Player” di dalam folder Animator



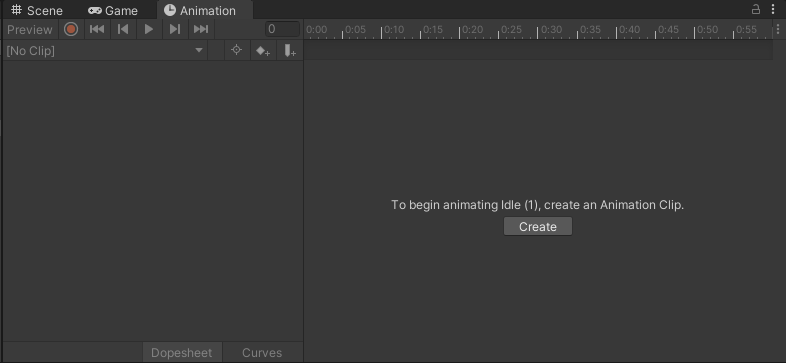
Gambar 9. 4 File Animator Controller

1. Setelah itu klik pada “Idle 1” lalau masuk kedalam inspector lalu ke animator pada bagian Animator Controller ubah Controller menjadi Player



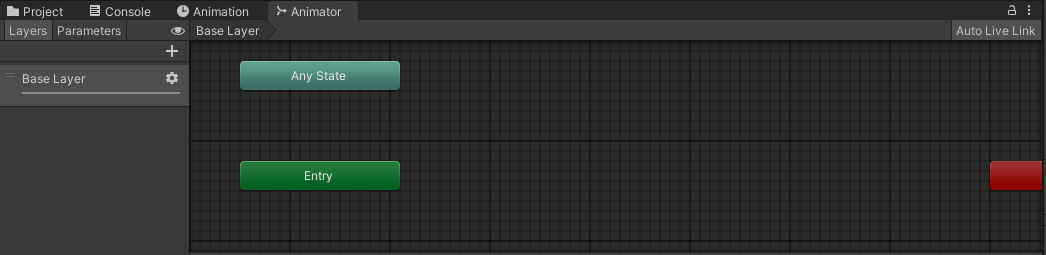
Gambar 9. 5 Setting Controller Animator

1. Setelah itu tambah jendela bagian animasi, dengan cara klik Window->animation->Animation, setelah itu letakan di bawah



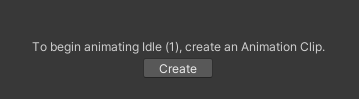
Gambar 9. 6 Jendela Bagian Animasi

1. Setelah itu masuk kedalam menu Window->Animation->Animator setelah itu geser jendela Animator ke bawah sebelah menu Animation



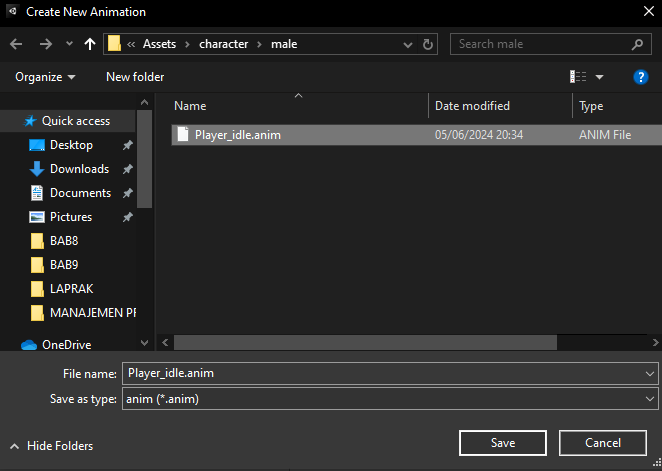
Gambar 9. 7 Halaman Animator

1. Setelah itu buat animasi baru dengan cara klik Idle 1 lalu masuk kedalam jendela Animation lalu klik Create



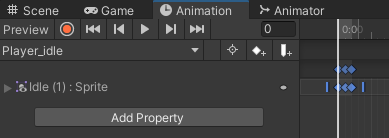
Gambar 9. 8 Membuat Animasi Baru

1. Setealh itu klik Create beri nama Player\_idle dengan format anim, file ini digunakan untu menyimpan animasi dari kondisi karakter



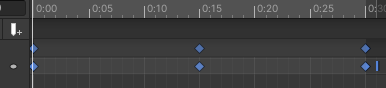
Gambar 9. 9 Membuat Animasi Idle

1. Setelah itu klik Save masuk kedalam folder Tugas Praktikum cari file bernama Idle 1, Idle 5, dan Idle 7 drag dan drop kedalam Animation



Gambar 9. 10 Animasi Idle

1. Setelah itu agar membuat animasi bertahan pada detik lama tidak berjalan terlalu cepat, masuk kedalam jendela Animation lalu geser kotak kotak kecil sampai frame 35



Gambar 9. 11 Mengatur Detik Animasi

1. Setelah coba jalankan game nya untuk mencoba apakah animasi telah berhasil dibuat



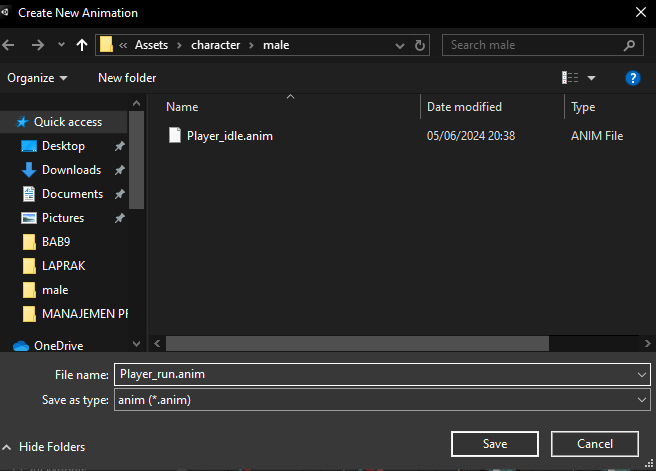
Gambar 9. 12 Menjalankan Game

1. Setelah itu buat animasi baru lagi bernama Player Run tetap di hierarki Idle 1 (Karakter), klik Player\_idle lalulalu klik New Clip



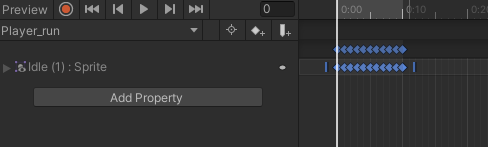
Gambar 9. 13 Create New Clip

1. Simpan dengan nama Player\_run.anim



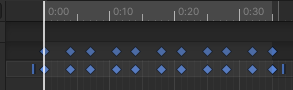
Gambar 9. 14 Membuat Animasi Berlari

1. Setelah itu masuk kedalam animasi Player\_run lalu drag drop file yang bernama Walk(1)-Walk(8)



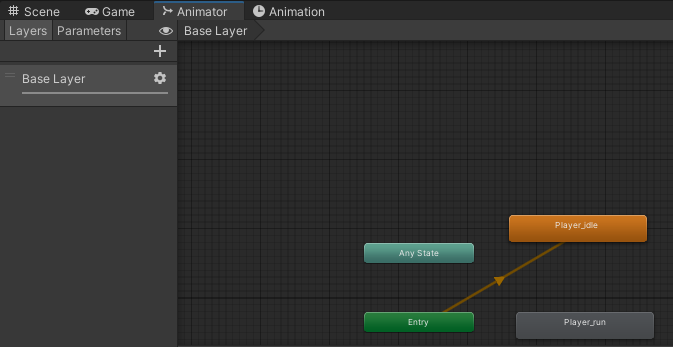
Gambar 9. 15 Animasi Run

1. Setelah itu geser animasi sampai dengan detik 35



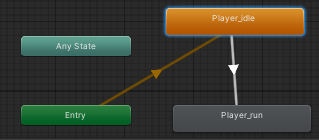
Gambar 9. 16 Memperpanjang Animasi

1. Setelah itu masuk kedalam jendela Animator



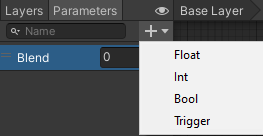
Gambar 9. 17 Masuk Kedalam Animator

1. Setelah itu klik kiri pada Player\_idle lalu klik Make Transition arahkan kedalam Player\_run



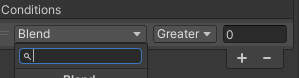
Gambar 9. 18 Transition Player\_idle

1. Setelah itu masuk kedalam tab parameter pada jenldea Animator, lalu klik + degan tipe data float, ubah Namanya menjeadi Blend



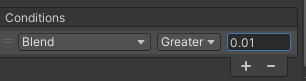
Gambar 9. 19 Membuat Tipe Data Baru

1. Setelah itu masuk kedalam Layers lalu klik pada panah warna putih, pada bagian conditions



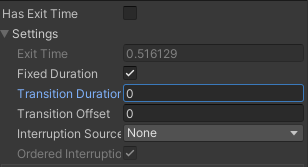
Gambar 9. 20 Bagian Conditions

1. Ubah greater menjadi 0.01



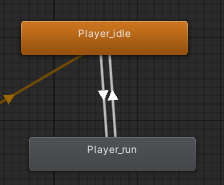
Gambar 9. 21 Kondisi 0.01

1. Setelah itu masuk kedalam setting inspector centang pada bagian Has Exit Time atur bagian Transition duration menjadi 0



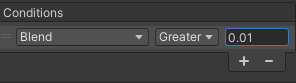
Gambar 9. 22 Setting Inspector

1. Buat transisi baru dari player run\_ke player idle



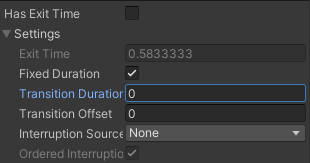
Gambar 9. 23 Transisi baru Player\_run

1. Setelah itu dengan cara yang sama klik bagian panah lalu atur bagian conditions, klik + lalu atur ke 0.01 greather



Gambar 9. 24 Mengatur Kondisi

1. Centang pada bagian Has Exit Time dan Transition Duration menjadi 0



Gambar 9. 25 Has Exit Time

1. Setelah itu masuk kedalam source code Player.cs, ubah pada bagian bawah class Player tambahkan kode program berikut

|  |
| --- |
| public Animator animator; |

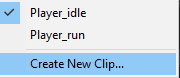
1. Setelah itu tambahkan kode berikut pada method Awake()

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

1. Tambahkan kode berikut pada method fixedUpdate()

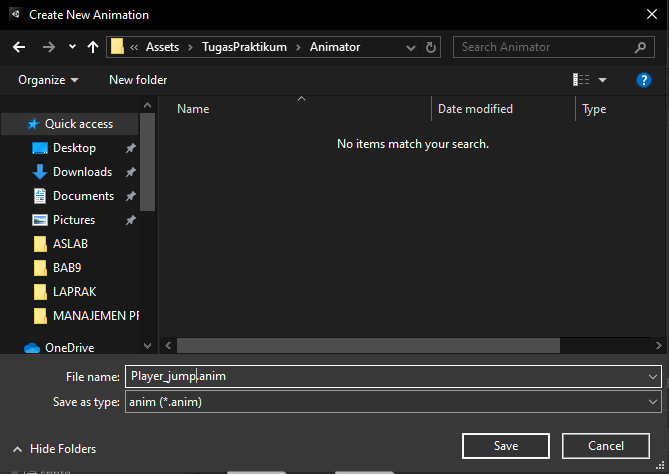
|  |
| --- |
| void FixedUpdate()  {  GroundCheck();  Move(horizontalValue, jump);  animator.SetFloat("Blend", Mathf.Abs(rb.velocity.x));} |

1. Setelah itu masuk kedalam jendela animation pada Idle (1), lalu buat variabel baru yuntuk enangani animasi loncat



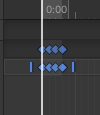
Gambar 9. 26 Membuat Animasi Baru

1. Beri nama Player\_jump.anim



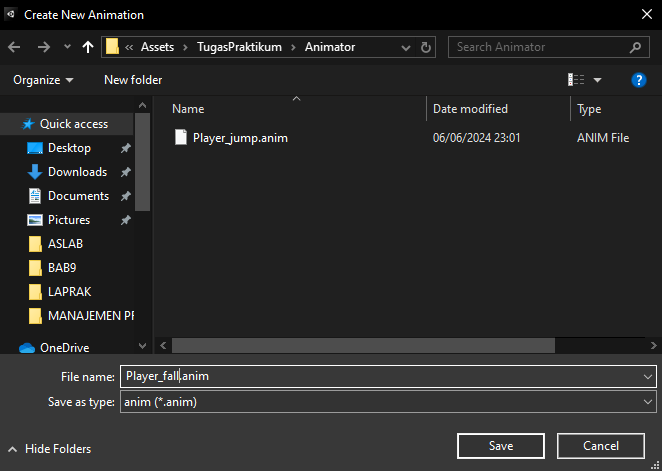
Gambar 9. 27 Membuat Animasi Baru

1. Setelah itu drag drop gambar Dead(2) kedalam animasi



Gambar 9. 28 Membuat Animasi Jump

1. Setelah itu buat animasi baru dengan nama Player\_fall



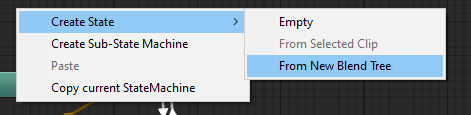
Gambar 9. 29 Animasi Player Fall

1. Lalu drag and drop gambar Dead(3) kedalam player fall



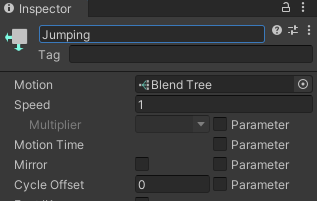
Gambar 9. 30 Membuat Animasi Player Fall

1. Masuk kedalam tab Animator setelah itu klik kanan Create State->From New Blend Tree



Gambar 9. 31 Membuat State Baru

1. Pada animator blend tree yang baru ubah nama menjadi Jumping



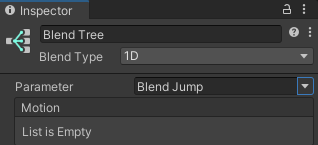
Gambar 9. 32 Membuat Blend Tree

1. Tambahkan parameter baru dengan nama Blend Jump dengan tipe data Float



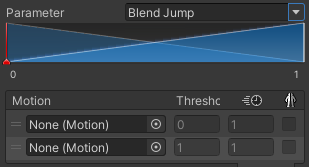
Gambar 9. 33 Parameter Blend Jump

1. Klik dua kali pada Tree Jumping lalu atur Blend Tree, setelah itu pada inspector ubah parameter Blend Tree menjadi Blend Jump



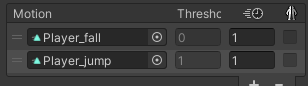
Gambar 9. 34 Mengubah Parameter Blend Tree

1. Klik icon + dibawah List Empty lalu buat dua motion baru



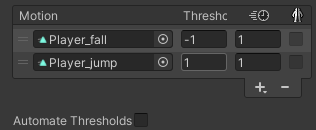
Gambar 9. 35 Membuat Motion Field

1. Ubah bagian Motion menjadi Motion Fall dan Motion Jump



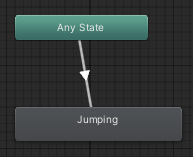
Gambar 9. 36 Motion Fall dan Motion Jump

1. Atur threshold seperti gambar 9.40 jangan lupa untuk hapus centang Automate Thresold



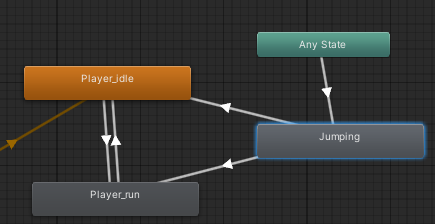
Gambar 9. 37 Mengatur Thresold

1. Masuk kedalam Base Layer pada Tree Any State klik kanan->make transition lalu arahkan panah tersebut kedalam Tree Jumping



Gambar 9. 38 Mengatur State Jumping

1. Kembali klik kanan pada Tree Jumping->make transition lalu arahkan kedalam Player Idle dan Player Run



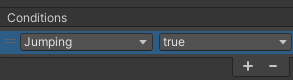
Gambar 9. 39 State Jumping

1. Tambah parameter bernama Jumping dengan tipe data Boolean ubah nilainya menjadi true



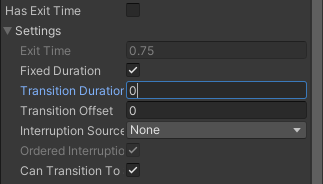
Gambar 9. 40 Parameter Jumping

1. Untuk melakukan kondisi pada posisi jumping klik kana panah yang mengarah pada Tree Jumping, tambahkan kondisi Jumping dan set menjadi true



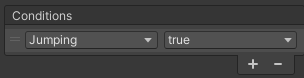
Gambar 9. 41 Kondisi Jumping

1. Pada transition duration ubah menjadi 0



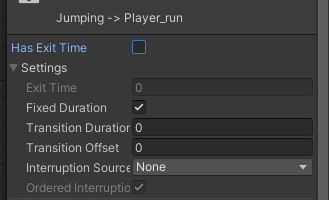
Gambar 9. 42 Mengatur Transition Duration

1. Klik panah yang mengarah pada Player\_run ubah tambahkan kondisi dan ubah menjadi Jumping dan set nilainya menjadi true



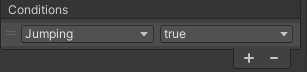
Gambar 9. 43 Kondisi Jumping Player\_run

1. Setelah itu masuk pada menu setting ubah pada Transition Duration ubah menjadi 0 dan Has Exit Time jangan dicentang



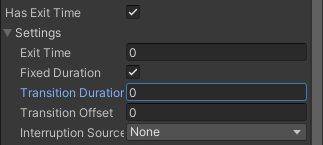
Gambar 9. 44 Setting Jumping Player\_run

1. Klik pada panah yang mengarah pada Player\_idle lalu buat kondisi baru, ubah kondisinya menjadi Jumping dan set nilainya menjadi true



Gambar 9. 45 Kondisi Jumping Player\_idle

1. Pada bagian Has Exit Time hilangkan centang dan ubah pada transition duration menjadi 0



Gambar 9. 46 Has Exit Time

1. Buka kode program Player.cs pada method Update() ubah dan tambahkan kode berikut

|  |
| --- |
| if (Input.GetButtonDown("Jump"))  animator.SetBool("Jumping", true);  jump = true;  else if (Input.GetButtonUp("Jump"))  jump = false; |

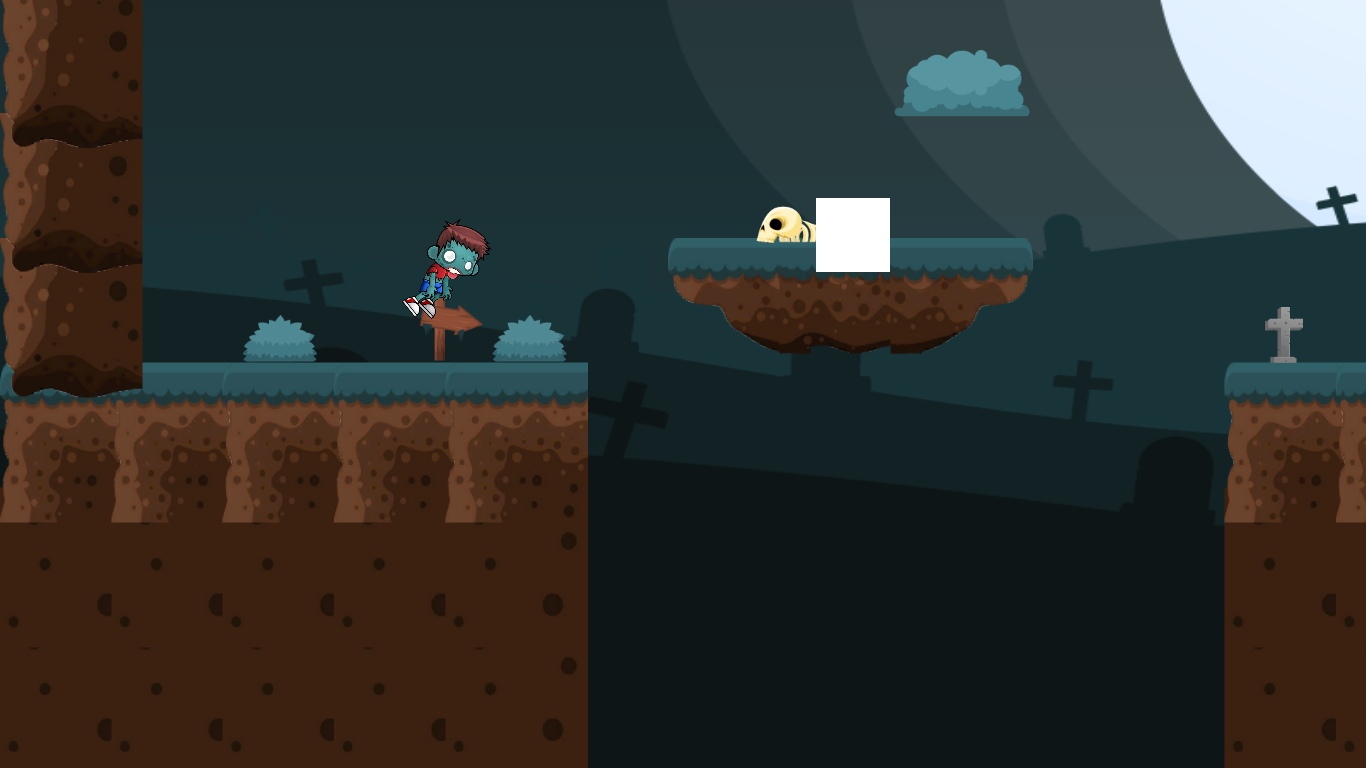
1. Pada method FixedUpdate() ubah dan tambahkan kode berikut

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

1. Pada method GroundCheck() ubah dan tambahkan kode berikut

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

1. Hasil Akhir



Gambar 9. 47 Hasil Akhir

**Link Github Pengumpulan**

Link : https://github.com/arifmalabar/2118055\_PRAK\_ANIGAME.git