



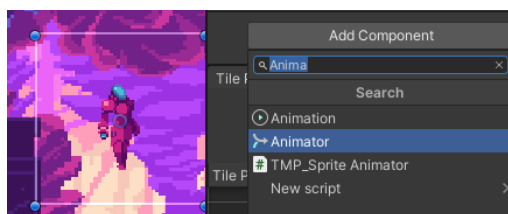
TUGAS PERTEMUAN: 9

MEMBUAT ANIMASI KARAKTER

NIM	:	2118030
Nama	:	Putra Prasetya Utama
Kelas	:	D
Asisten Lab	:	Aprillia Dwi Dyah S. (2118143)

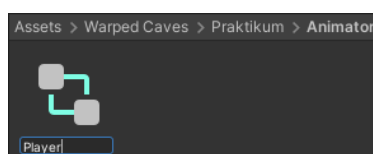
9.1 Tugas 1 : Langkah-langkah Membuat Animasi Karakter Pada Game

1. Buka *project* sebelumnya, lalu klik pada karakter setelah itu menuju ke *inspector* dan tambahkan komponen *Animator*.



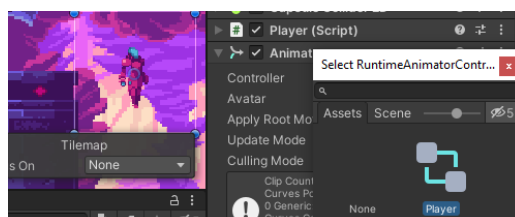
Gambar 9.1 Menambahkan Komponen Animator Pada Karakter

2. Buat folder *Animator* didalam folder *Praktikum*. Lalu tambahkan File *Animator Controller* didalam folder *Animator* dan ubah namanya menjadi *Player*.



Gambar 9.2 Menambahkan Animator Controller

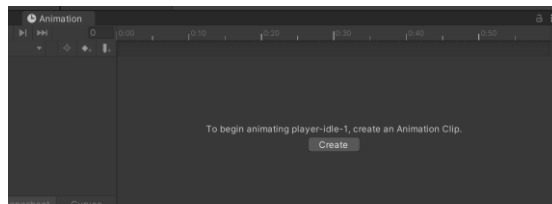
3. Setelah itu tambahkan panel *animation* dengan menekan CTRL + 6 atur panel sesuai keinginan. Lalu klik pada *Player* lalu pergi ke bagian komponen *Animator* yang sudah ditambahkan. Pada *Controller* ubah menjadi *Player*.



Gambar 9.3 Menambahkan Controller Player Pada Komponen Animator

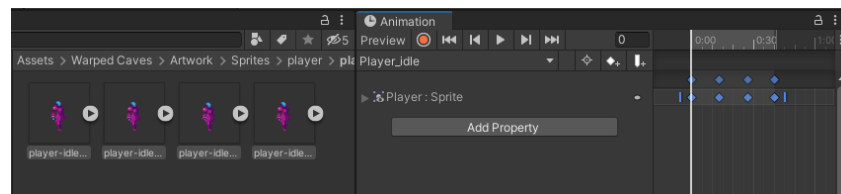


4. Tambahkan panel *Animator* dibagian *Window* pilih *Animation* lalu pilih *Animator*. Setelah itu klik *Player* lalu pilih *Create* pada panel *Animation*.



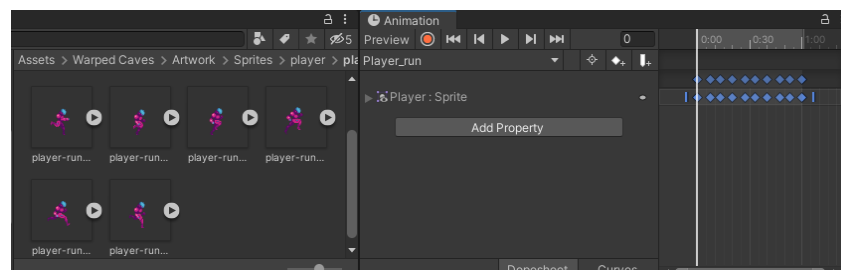
Gambar 9.4 Membuat Animasi Untuk Player

5. Lalu simpan pada folder *Animator* yang sudah dibuat sebelumnya dan beri nama *Player_idle*. Lalu temukan folder *player-idle*, pilih semuanya lalu *drag* dan *drop* di panel *Animation*. Lalu panjangkan garis biru ke detik 0:40.



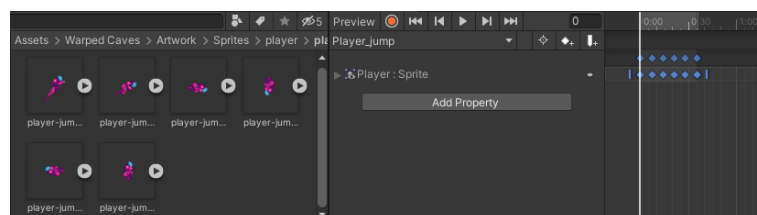
Gambar 9.5 Menambahkan Player Idle Ke Panel Animation

6. Lalu klik *Player_idle* lalu pilih *create new clip* dan beri nama *Player_run*. lu temukan folder *player-run*, pilih semuanya lalu *drag* dan *drop* di panel *Animation*. Lalu panjangkan garis biru ke detik 1:00.



Gambar 9.6 Menambahkan Player Run Ke Panel Animation

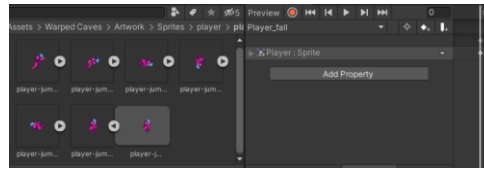
7. Lalu *create new clip* dan beri nama *Player_jump*. lu temukan folder *player-jump*, pilih semuanya lalu *drag* dan *drop* di panel *Animation*. Lalu panjangkan garis biru ke detik 0:35. Lalu hapus *keyframe* terakhir.



Gambar 9.7 Menambahkan Player Jump Ke Panel Animation

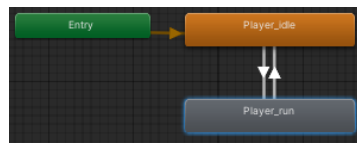


8. Lalu *create new clip* dan beri nama *Player_fall*. lu temukan folder *player-jump*, pilih paling terakhir saja lalu *drag* dan *drop* di panel *Animation*.



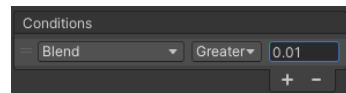
Gambar 9.8 Menambahkan Player Fall Ke Panel Animation

9. Lalu pergi ke panel *Animator* lalu klik kanan pada *Player_idle* lalu pilih *Make Transition* dan pilih tujuannya *Player_run*. Lakukan sebaliknya juga hingga terdapat arah panah berwarna putih.



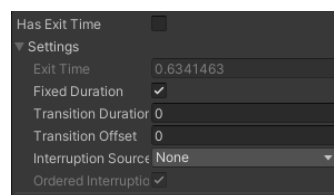
Gambar 9.9 Membuat Transisi

10. Pergi ke *tab* parameter lalu tambahkan parameter *float* baru dan ubah namanya menjadi *Blend*. Klik panah putih yang mengarah ke *Player_run*, pada bagian *conditions* icon tambah kemudian atur menjadi “Blend”. Lalu atur menjadi 0.01.



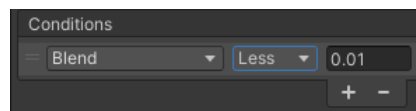
Gambar 9.10 Menambahkan Kondisi Dengan Operator Greater

11. Lalu buka bagian *Setting* dan atur seperti gambar dibawah ini.



Gambar 9.11 Melakukan Setting Pada Panah Tujuan Player_run

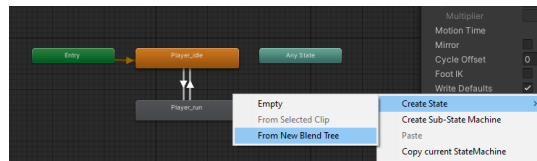
12. Lalu klik panah putih yang mengarah ke *Player_idle* dan atur seperti pada nomor 9 tapi ubah operator menjadi *Less*. Lalu atur seperti pada gambar 10 juga.



Gambar 9.12 Menambah Kondisi Dengan Operator Less

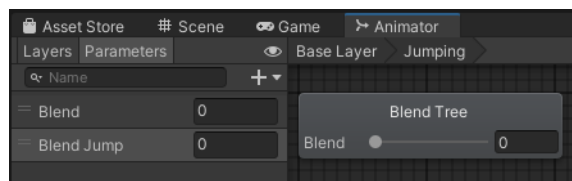


13. Klik kanan pada panel *Animator*, pilih *Create State* lalu pilih *From New Blend Tree*. Ubah namanya menjadi *Jumping*.



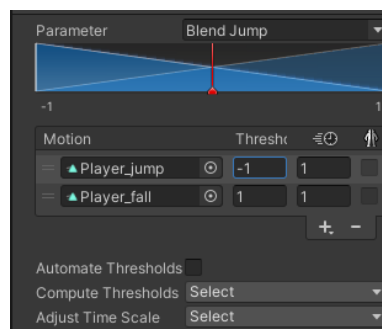
Gambar 9.13 Membuat Blend Tree Baru

14. Buat parameter baru tipe *Float* dan ubah namanya menjadi *Blend Jump*.
Lalu klik dua kali pada *Blend Tree Jumping* yang sudah dibuat sebelumnya.



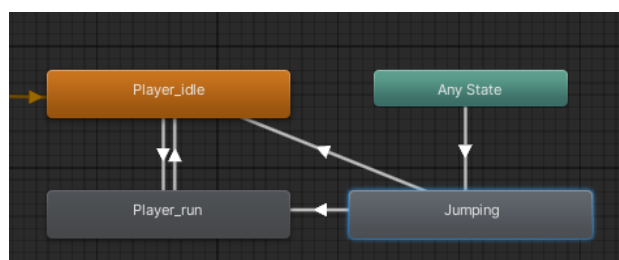
Gambar 9.14 Parameter Baru Dan Masuk Ke Blend Tree Jumping

15. Pada *inspector* ubah parameter menjadi *Blend Jump*. Lalu tambahkan dua *Motion Field*. Hilangkan centang “Automate Thresholds”. Lalu atur seperti gambar berikut.



Gambar 9.15 Menambahkan Dua Motion

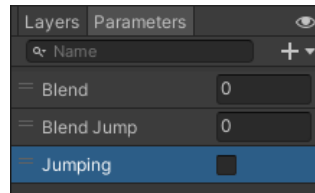
16. Kembali ke *Base Layer*, klik kanan *Any State*, pilih *Make Transition* dan arahkan panahnya ke *Jumping*. Klik kanan *Jumping*, pilih *Make Transition* dan arahkan panahnya ke *Player_idle* dan *Player_run*.



Gambar 9.16 Menambahkan Transisi Baru

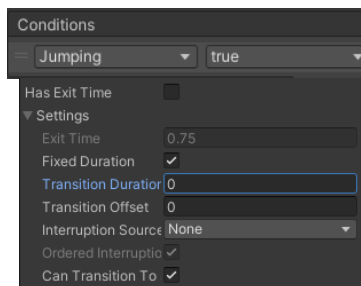


17. Buat parameter baru dengan tipe *Bool* lalu ubah namanya menjadi *Jumping*.



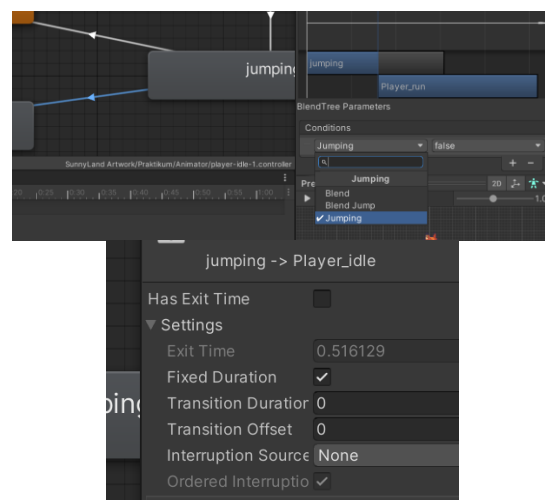
Gambar 9.17 Menambah Parameter Jumping

18. Klik panah yang mengarah ke *Jumping*, pada inspector tambahkan *condition*, pilih *condition Jumping* dan ubah nilainya menjadi *true*. Klik *Settings* dan ubah nilai *Transition Duration* menjadi 0 dan hilangkan centang *Has Exit Time*.



Gambar 9.18 Menambah Kondisi dan Mengatur Setting

19. 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*. Klik *Settings* dan ubah nilai *Transition Duration* menjadi 0 dan hilangkan centang *Has Exit Time*.



Gambar 9.19 Mengubah Juga Pada Panah Player_idle dan Player_run



20. Ubah *source code* pada *script Player* menjadi seperti dibawah ini agar *Player* dapat memiliki animasi saat *idle* dan saat melompat.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Player : MonoBehaviour{
    public Animator animator;
    Rigidbody2D rb;
    [SerializeField] Transform groundcheckCollider;
    [SerializeField] LayerMask groundLayer;
    const float groundCheckRadius = 0.2f;
    [SerializeField] float speed = 3;
    [SerializeField] float jumpPower = 100;
    bool jump;
    float horizontalValue;
    [SerializeField] bool isGrounded; // +
    bool facingRight;
    private void Awake(){
        animator = GetComponent<Animator>();
        rb = GetComponent<Rigidbody2D>();
    }
    void Update (){
        horizontalValue = Input.GetAxisRaw("Horizontal");
        if ([Input.GetButtonDown("Jump")]){
            animator.SetBool("Jumping", true);
            jump = true;
        }else if (Input.GetButtonUp("Jump")){
            jump = false;
        }
    }

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

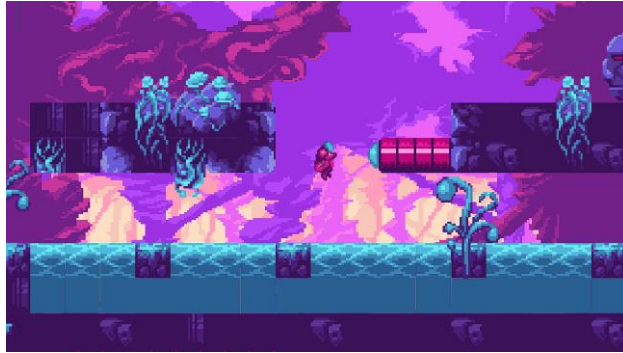
    void GroundCheck(){
        isGrounded = false;
        Collider2D[] colliders = Physics2D.OverlapCircleAll(groundcheckCollider.position, groundCheckRadius, groundLayer);
        if (colliders.Length > 0){
            isGrounded = true;
        }
        animator.SetBool("Jumping", !isGrounded);
   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

    void Move(float dir, bool jumpflag){
        if(isGrounded && jumpflag){
            isGrounded = false;
            jumpflag = false;
            rb.AddForce(new Vector2(0f, jumpPower));
        }
        #region gerak kanan kiri
        float xVal = dir * speed * 100 * Time.fixedDeltaTime;
        Vector2 targetVelocity = new Vector2(xVal, rb.velocity.y);
        rb.velocity = targetVelocity;

        if (facingRight && dir < 0){
            transform.localScale = new Vector3(-4, 4, 4);
            facingRight = false;
        }else if (!facingRight && dir > 0){
            transform.localScale = new Vector3(4, 4, 4);
            facingRight = true;
        }
        #endregion
    }
}
```



21. Lalu klik CTRL + SHIFT + B untuk melakukan *Build and Run*. Simpan sesuai folder yang di inginkan lalu coba mainkan.



Gambar 9.20 Menjalankan Game Setelah Di Build and Run

9.2 Kuis

```
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);
    }
}
```

Analisis:

Pada baris bagian “`animator.SetBool("isJumping",);`” ini tidak ada nilai *boolean* maka harus diberi nilai *boolean true* karena kondisi pada *if* nya itu saat tombol *keyboard* spasi ditekan nanti animasi *isJumping* harus *true* untuk karakter dapat meloncat dengan tombol spasi ditekan.



Pada baris `“animator.SetBool(“isJumping”,);”` ini juga sama harus diberi nilai *true* karena kondisi dibagian *else if* yaitu saat tombol spasi masih ditekan yang nantinya karakter juga harus tetap loncat bila tombol spasi masih ditekan terus-menerus.

Lalu pada bagian fungsi *HandleJumpInput* juga tambahkan bagian *else* agar karakter tidak loncat dengan memberi `“animator.SetBool(“isJumping”, false);”`.

Lalu pada bagian fungsi *HandleMovementInput* bagian kondisi *if* pertama tidak tepat karena kondisinya masih terlalu acak karena angka yang tidak sama dengan 1 itu ada banyak sekali. Mungkin dapat diubah pada bagian kondisinya menjadi `move == 0` yang artinya yaitu tidak ada gerakan. Lalu isi bloknnya juga ditambahkan dibawahnya `animator.SetBool(“isIdle”, true);` ditambahkan dengan `“animator.SetBool(“isWalking”, false);”` untuk memastikan bahwa karakter tidak sedang berjalan.

Lalu pada bagian `“transform.Translate(Vector3.left * move * Time.deltaTime);”` pindahkan ke bagian *else* agar karakter tidak bergerak ke arah kiri meskipun kondisi karakter sedang tidak bergerak.

Lalu didalam *else* yang bagian `“animator.SetBool(“isWalking”, false);”` bagian *false* diganti menjadi *true* agar karakternya dalam keadaan berjalan, jika tetap menggunakan *false* maka karakter tidak dapat berjalan. Lalu tambahkan pula `“animator.SetBool(“isIdle”, false);”` sebelum `“isWalking”, true` untuk memastikan agar karakter tidak dalam keadaan diam terlebih dulu setelah itu karakter dapat berjalan.

Lalu pada kondisi *if* kedua kondisinya kurang tepat, ganti menjadi `“move < 0”` untuk menandakan bahwa karakter bergerak ke arah kiri lalu atur skala karakter untuk menghadap ke kiri dengan parameternya yaitu `(-x, y, z)` agar karakter menghadap ke kiri.

Lalu atur agar nilai parameternya di bagian *if* `“(-4, 1, 1)”` dan *else if* `“(1, 2, 1)”` disesuaikan dengan ukuran lokal karakter, jika ukuran karakter lokalnya pada sumbu x adalah 4, sumbu y adalah 2 dan sumbu z adalah satu. Maka samakan saja nilainya menjadi `“-4, 2, 1”` untuk bagian *if* dan `“4, 2, 1”` untuk bagian *else if*.