using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class EnemyAnimator : MonoBehaviour

{

private Animator anim;

void Awake()

{

anim = GetComponent<Animator>();

}

public void Walk(bool walk) {

anim.SetBool(AnimationTags.WALK\_PARAMETER, walk);

}

public void Run(bool run)

{

anim.SetBool(AnimationTags.RUN\_PARAMETER, run);

}

public void Attack()

{

anim.SetTrigger(AnimationTags.ATTACK\_TRIGGER);

}

public void Dead()

{

anim.SetTrigger(AnimationTags.DEAD\_TRIGGER);

}

// Start is called before the first frame update

void Start()

{

}

// Update is called once per frame

void Update()

{

}

}

private Animator anim;

**private Animator anim;**: Dòng này khai báo một biến riêng tư tên là anim với kiểu Animator. Thành phần Animator trong Unity được sử dụng để kiểm soát hệ thống hoạt ảnh cho một GameObject, cho phép bạn phát, pha trộn và chuyển đổi giữa các hoạt ảnh khác nhau.

void Awake()

{

anim = GetComponent<Animator>();

}

Hàm Awake trong đoạn mã này được sử dụng để khởi tạo biến anim khi đối tượng script được tải lên. Cụ thể, nó lấy thành phần Animator gắn với cùng một GameObject và gán nó cho biến anim. Đây là cách đảm bảo rằng script có thể sử dụng Animator để kiểm soát hoạt ảnh của nhân vật.

Dòng này tìm và lấy thành phần Animator từ GameObject mà script EnemyAnimator đang được gắn vào và gán nó cho biến anim. Điều này cho phép script sử dụng biến anim để truy cập và điều khiển các hoạt ảnh của nhân vật thông qua Animator.

Hàm Awake trong đoạn code này được sử dụng để khởi tạo biến anim bằng cách lấy thành phần Animator từ cùng GameObject. Điều này cho phép script EnemyAnimator sử dụng anim để điều khiển các hoạt ảnh của nhân vật. Khởi tạo trong Awake đảm bảo rằng Animator đã sẵn sàng để sử dụng trước khi bất kỳ logic nào khác được thực hiện.

public void Walk(bool walk) {

anim.SetBool(AnimationTags.WALK\_PARAMETER, walk);

}

 anim.SetBool là một phương thức của Animator để thiết lập giá trị của một tham số boolean trong Animator.

 AnimationTags.WALK\_PARAMETER là một biến chứa tên của tham số boolean trong Animator liên quan đến hoạt ảnh đi bộ. Thông thường, giá trị này được định nghĩa ở nơi khác trong mã, có thể là một hằng số hoặc biến tĩnh để dễ quản lý.

 walk là giá trị boolean được truyền vào phương thức Walk. Nếu walk là true, tham số hoạt ảnh sẽ được đặt thành true, kích hoạt hoạt ảnh đi bộ. Ngược lại, nếu walk là false, tham số sẽ được đặt thành false, dừng hoạt ảnh đi bộ.

 **Điều khiển hoạt ảnh động**: Phương thức Walk cung cấp một cách đơn giản để điều khiển hoạt ảnh đi bộ của kẻ thù. Bằng cách gọi phương thức này và truyền vào giá trị true hoặc false, bạn có thể bắt đầu hoặc dừng hoạt ảnh đi bộ.

 **Tương tác với Animator**: Bằng cách sử dụng Animator, hàm này cho phép sự chuyển đổi mượt mà giữa các trạng thái hoạt ảnh khác nhau dựa trên các tham số được đặt trong Animator Controller.

 **Chức năng**: Hàm Run được sử dụng để kích hoạt hoặc dừng hoạt ảnh chạy của nhân vật kẻ thù.

 **Chi tiết**:

* anim.SetBool(AnimationTags.RUN\_PARAMETER, run);: Dòng này sử dụng phương thức SetBool của Animator để đặt giá trị cho một tham số boolean trong Animator.
* AnimationTags.RUN\_PARAMETER: Là một hằng số (hoặc biến tĩnh) chứa tên của tham số boolean trong Animator liên quan đến hoạt ảnh chạy.
* run: Tham số boolean được truyền vào hàm Run. Nếu run là true, thì hoạt ảnh chạy sẽ được kích hoạt. Nếu run là false, hoạt ảnh chạy sẽ dừng lại.

public void Attack()

{

anim.SetTrigger(AnimationTags.ATTACK\_TRIGGER);

}

 **Chức năng**: Hàm Attack được sử dụng để kích hoạt hoạt ảnh tấn công của nhân vật kẻ thù.

 **Chi tiết**:

* anim.SetTrigger(AnimationTags.ATTACK\_TRIGGER);: Dòng này sử dụng phương thức SetTrigger của Animator để kích hoạt một trigger trong Animator.
* AnimationTags.ATTACK\_TRIGGER: Là một hằng số (hoặc biến tĩnh) chứa tên của trigger trong Animator liên quan đến hoạt ảnh tấn công. Khi trigger này được kích hoạt, Animator sẽ chuyển sang trạng thái hoạt ảnh tấn công.

public void Dead()

{

anim.SetTrigger(AnimationTags.DEAD\_TRIGGER);

}

 **Chức năng**: Hàm Dead được sử dụng để kích hoạt hoạt ảnh khi nhân vật kẻ thù chết.

 **Chi tiết**:

* anim.SetTrigger(AnimationTags.DEAD\_TRIGGER);: Dòng này sử dụng phương thức SetTrigger của Animator để kích hoạt một trigger trong Animator.
* AnimationTags.DEAD\_TRIGGER: Là một hằng số (hoặc biến tĩnh) chứa tên của trigger trong Animator liên quan đến hoạt ảnh khi nhân vật chết. Khi trigger này được kích hoạt, Animator sẽ chuyển sang trạng thái hoạt ảnh khi nhân vật chết.

Các hàm trên có tác dụng sau:

 **Điều khiển hoạt ảnh dễ dàng**: Các hàm này cung cấp một giao diện dễ sử dụng để điều khiển các hoạt ảnh của nhân vật kẻ thù. Thay vì phải gọi trực tiếp các phương thức của Animator, có thể gọi các hàm này với các tham số đơn giản như true, false, hoặc không cần tham số để kích hoạt các trạng thái hoạt ảnh khác nhau.

 **Phản hồi nhanh chóng**: Bằng cách sử dụng SetBool và SetTrigger, có thể nhanh chóng thay đổi trạng thái hoạt ảnh của nhân vật kẻ thù dựa trên các sự kiện trong trò chơi như tấn công, chạy, hoặc chết.