using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public enum WeaponAim

{

NONE,

SELF\_AIM,

AIM

}

public enum WeaponFireType

{

SINGLE,

MULTIPLE

}

public enum WeaponBulletType

{

BULLET,

ARROW,

SPEAR,

NONE

}

public class WeaponHandler : MonoBehaviour

{

private Animator anim;

public WeaponAim weapon\_Aim;

[SerializeField]

private GameObject muzzleFlash;

[SerializeField]

private AudioSource shootSound,reload\_Sound;

public WeaponFireType fireType;

public WeaponBulletType bulletType;

public GameObject attack\_Point;

void Awake()

{

anim = GetComponent<Animator>();

}

public void ShootAnimation()

{

anim.SetTrigger(AnimationTags.SHOOT\_TRIGGER);

}

public void Aim(bool canAim)

{

anim.SetBool(AnimationTags.AIM\_PARAMETER, canAim);

}

void Turn\_On\_MuzzleFlash()

{

muzzleFlash.SetActive(true);

}

void Turn\_Off\_MuzzleFlash()

{

muzzleFlash.SetActive(false);

}

void Play\_ShootSound()

{

shootSound.Play();

}

void Play\_ReloadSound()

{

reload\_Sound.Play();

}

void Turn\_On\_AttackPoint()

{

attack\_Point.SetActive(true);

}

void Turn\_Off\_AttackPoint()

{

if (attack\_Point.activeInHierarchy)

{

attack\_Point.SetActive(false);

}

}

// Start is called before the first frame update

void Start()

{

}

// Update is called once per frame

void Update()

{

}

}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public enum WeaponAim

{

NONE,

SELF\_AIM,

AIM

}

public enum WeaponFireType

{

SINGLE,

MULTIPLE

}

public enum WeaponBulletType

{

BULLET,

ARROW,

SPEAR,

NONE

}

Các enum này được sử dụng để đại diện cho các trạng thái khác nhau hoặc loại khác nhau liên quan đến vũ khí trong một hệ thống game hoặc phần mềm mô phỏng. Cụ thể:

1. **WeaponAim**: Đại diện cho cách thức nhắm bắn của vũ khí.
   * NONE: Không nhắm bắn.
   * SELF\_AIM: Tự nhắm (có thể là auto-aim hoặc một loại nhắm bắn tự động).
   * AIM: Nhắm bắn thủ công (người chơi hoặc người sử dụng sẽ nhắm bắn).
2. **WeaponFireType**: Đại diện cho loại bắn của vũ khí.
   * SINGLE: Bắn từng viên một.
   * MULTIPLE: Bắn nhiều viên (có thể là bắn loạt hoặc bắn nhiều viên cùng lúc).
3. **WeaponBulletType**: Đại diện cho loại đạn mà vũ khí sử dụng.
   * BULLET: Đạn.
   * ARROW: Mũi tên.
   * SPEAR: Giáo.
   * NONE: Không có đạn (có thể là vũ khí cận chiến hoặc trạng thái không có đạn).

Những enum này có thể được sử dụng trong các lớp hoặc phương thức để xác định cách một vũ khí sẽ hoạt động, hoặc để kiểm tra trạng thái hiện tại của vũ khí trong game. Ví dụ:

* Một lớp vũ khí có thể có các thuộc tính để lưu trữ loại nhắm bắn, loại bắn và loại đạn, và từ đó xác định hành vi của vũ khí khi người chơi sử dụng nó.
* Trong logic game, có thể sử dụng các enum này để thay đổi cách mà vũ khí hoạt động dựa trên loại nhắm bắn hoặc loại bắn hiện tại.

public class WeaponHandler : MonoBehaviour

{

private Animator anim;

public WeaponAim weapon\_Aim;

[SerializeField]

private GameObject muzzleFlash;

[SerializeField]

private AudioSource shootSound,reload\_Sound;

public WeaponFireType fireType;

public WeaponBulletType bulletType;

public GameObject attack\_Point;

Khai báo các biến sau để phục vụ cho việc gọi hàm,cho sound effect của súng và thêm điểm tấn công

void Awake()

{

anim = GetComponent<Animator>();

}

Khai báo anim = GetComponent<Animator>(); để nhận dữ liệu từ Animator

public void ShootAnimation()

{

anim.SetTrigger(AnimationTags.SHOOT\_TRIGGER);

}

Chức năng: Kích hoạt hoạt hình bắn súng.

Mô tả: Hàm này sử dụng anim.SetTrigger với một tham số AnimationTags.SHOOT\_TRIGGER để kích hoạt hoạt hình bắn súng.

public void Aim(bool canAim)

{

anim.SetBool(AnimationTags.AIM\_PARAMETER, canAim);

}

Chức năng: Điều khiển trạng thái nhắm bắn của vũ khí.

Mô tả: Hàm này thiết lập một giá trị boolean cho hoạt hình, dựa trên tham số canAim, để quyết định xem nhân vật có đang nhắm bắn hay không.

void Turn\_On\_MuzzleFlash()

{

muzzleFlash.SetActive(true);

}

Chức năng: Bật hiệu ứng ánh sáng nòng súng (muzzle flash).

Mô tả: Hàm này kích hoạt đối tượng muzzleFlash để hiển thị hiệu ứng ánh sáng khi bắn.

void Turn\_Off\_MuzzleFlash()

{

muzzleFlash.SetActive(false);

}

Chức năng: Tắt hiệu ứng ánh sáng nòng súng (muzzle flash).

Mô tả: Hàm này vô hiệu hóa đối tượng muzzleFlash để tắt hiệu ứng ánh sáng sau khi bắn.

void Play\_ShootSound()

{

shootSound.Play();

}

Chức năng: Phát âm thanh khi bắn súng.

Mô tả: Hàm này phát âm thanh bắn súng bằng cách gọi phương thức Play trên đối tượng shootSound.

void Play\_ReloadSound()

{

reload\_Sound.Play();

}

Chức năng: Phát âm thanh khi nạp đạn.

Mô tả: Hàm này phát âm thanh nạp đạn bằng cách gọi phương thức Play trên đối tượng reload\_Sound.

void Turn\_On\_AttackPoint()

{

attack\_Point.SetActive(true);

}

Chức năng: Bật điểm tấn công (có thể là vùng va chạm khi tấn công).

Mô tả: Hàm này kích hoạt đối tượng attack\_Point để bật vùng va chạm khi thực hiện tấn công.

void Turn\_Off\_AttackPoint()

{

if (attack\_Point.activeInHierarchy)

{

attack\_Point.SetActive(false);

}

}

Chức năng: Tắt điểm tấn công.

Mô tả: Hàm này kiểm tra xem đối tượng attack\_Point có đang được kích hoạt hay không, nếu có thì sẽ vô hiệu hóa nó.

Kết luận:

Nhìn chung, các hàm này quản lý các khía cạnh khác nhau của hoạt động bắn súng và tấn công trong trò chơi, bao gồm hoạt hình, hiệu ứng hình ảnh, và âm thanh, để tạo ra trải nghiệm thực tế và sinh động cho người chơi.