

Trails of Totec

Scripts

Liam Pickering

The SFXManager Script handles the activating and fading of the environment SFX in Trials of Totec.

The script uses Boolean Logic and a series of IF and ELSE IF loops, along with a RNG to activate and fade between 2 SFX tracks at between random intervals that the users can predefine.

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

public class SFXManager : MonoBehaviour
{
    public AudioSource longHumAS;
    public AudioSource shortHumAS;

    public float fadeTime = 5f;

    //time off
    public float minTimeOff = 5f;
    public float maxTimeOff = 15f;
    float timeOff;

    //time on
    public float minTimeOn = 5f;
    public float maxTimeOn = 15f;
    float timeOn;

    //volume
    public float minVol = 0.5f;
    public float maxVol = 1.0f;

    float longHumVol;
    float shortHumVol;

    //timer
    public float timer;
    public float timeTillNextSound;

    //bools
    bool longHumPlaying;
    bool shortHumPlaying;

    void Start()
    {
        RNG();

        timeTillNextSound = timeOff;
```

Trails of Totec

Scripts

Liam Pickering

```
}  
  
void Update()  
{  
    timer += Time.deltaTime;  
  
    if(timer > timeTillNextSound)  
    {  
        if (longHumPlaying)  
        {  
            StartCoroutine(FadeLongHum(longHumAS.volume, 0, fadeTime));  
  
            longHumPlaying = false;  
  
            RNG();  
  
            timeTillNextSound = timeOff + fadeTime;  
        }  
  
        else if (!longHumPlaying)  
        {  
            StartCoroutine(FadeLongHum(0, longHumVol, fadeTime));  
  
            longHumPlaying = true;  
  
            RNG();  
  
            timeTillNextSound = timeOn + fadeTime;  
        }  
  
        if (shortHumPlaying)  
        {  
            StartCoroutine(FadeShortHum(shortHumAS.volume, 0, fadeTime));  
  
            shortHumPlaying = false;  
  
            RNG();  
  
            timeTillNextSound = timeOff + fadeTime;  
        }  
  
        else if (!shortHumPlaying)  
        {  
            StartCoroutine(FadeShortHum(0, longHumVol, fadeTime));
```

Trails of Totec

Scripts

Liam Pickering

```

        shortHumPlaying = true;

        RNG();

        timeTillNextSound = timeOn + fadeTime;
    }

    timer = 0;
}

void RNG()
{
    timeOff = Random.Range (minTimeOff, maxTimeOff);
    timeOn = Random.Range(minTimeOn, maxTimeOn);

    longHumVol = Random.Range(minVol, maxVol);
    shortHumVol = Random.Range(minVol, maxVol);
}

IEnumerator FadeLongHum(float startValue, float endValue, float duration)
{
    float currentTime = 0;

    while (currentTime <= duration)
    {
        longHumAS.volume = Mathf.Lerp(startValue, endValue, (currentTime /
duration));
        currentTime += Time.deltaTime;
        yield return null;
    }
}

IEnumerator FadeShortHum(float startValue, float endValue, float duration)
{
    float currentTime = 0;

    while (currentTime <= duration)
    {
        shortHumAS.volume = Mathf.Lerp(startValue, endValue, (currentTime /
duration));
        currentTime += Time.deltaTime;
        yield return null;
    }
}

```

Trails of Totec

Scripts

Liam Pickering