

```

// Hanabi_Firing.cs

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

public class Hanabi_Firing : MonoBehaviour
{
    [SerializeField]
    private GameObject[] m_HanabiPrefabs;           // 花火プレハブ
    [SerializeField]
    private HalfPoint2 m_HalfPoint2;               // HalfPoint2
    [SerializeField]
    private int m_InstantiateCounter;               // 生成した数
    [SerializeField]
    private float m_NextCoolTime;                  // 花火を生成するまでのクール
    タイム

    private bool m_RandFlag = false;               // ランダム係数を行うためのフラ
    グ
    private bool m_FireworksIsPlay;                // 花火が上がっているか確かめる
    ためのフラグ
    private bool m_InstantiateFlag;                // インスタンスフラグ

    private int m_PrefabsNumber;                   // 花火の種類
    private float m_Timer;                         // タイマー
    private float m_RandX, m_RandY, m_RandZ;       // X・Y・Z座標

    // Use this for initialization
    void Start ()
    {
        m_HalfPoint2 =
        GameObject.FindGameObjectWithTag("HalfPoint2").GetComponent<HalfPoint2>();

        m_InstantiateCounter = 0;
        m_FireworksIsPlay = true;
    }
}

```

```
}
```

```
// Update is called once per frame
```

```
void Update ()
```

```
{
```

```
    // フェードアウト/インが終わっていたら
```

```
    if (m_HalfPoint2.m_FireworksFadeFlag == true)
```

```
    {
```

```
        // 花火が上がっているどうか？
```

```
        if (m_FireworksIsPlay)
```

```
        {
```

```
            m_InstantiateFlag = true;
```

```
            // 生成を行う
```

```
            if (m_InstantiateFlag)
```

```
            {
```

```
                InstantiatePrefab();
```

```
                m_FireworksIsPlay = false;
```

```
            }
```

```
        }
```

```
    // HalfPoint2に到達した灯籠と生成された花火の数が一緒なら
```

```
    if (m_InstantiateCounter == m_HalfPoint2.m_TourouCount)
```

```
    {
```

```
        m_FireworksIsPlay = false;
```

```
        m_HalfPoint2.m_FireworksFadeFlag = false;
```

```
        m_Timer = 0.0f;
```

```
    }
```

```
    if (m_InstantiateCounter < m_HalfPoint2.m_TourouCount)
```

```
    {
```

```
        m_Timer += Time.deltaTime;
```

```
        if (m_Timer > 3.0f)
```

```
        {
```

```
            m_FireworksIsPlay = true;
```

```

        m_Timer = 0.0f;
    }
}

// 花火が全部あがったら
if (m_FireworksIsPlay == false)
{
    m_Timer += Time.deltaTime;

    if (m_Timer >= m_NextCoolTime)
    {
        // フェード演出を始めさせる
        m_HalfPoint2.m_FireworksEndingProduction = true;
        m_HalfPoint2.m_FireworksEndingStart = true;

        // フェード演出が終わっていたら
        if (m_HalfPoint2.m_EndFireworksFadeFlag == true)
        {
            // 灯籠スクリプトを起動させる
            foreach (GameObject tourou in
GameObject.FindGameObjectsWithTag("Tourou"))
            {
                // 灯籠スクリプトを取得
                Tourou tourouScript = tourou.GetComponent<Tourou>();
                tourouScript.enabled = true;
                // Rigidbodyのポジションとローテーションの制約を元に戻す
                Rigidbody rigidbody = tourou.GetComponent<Rigidbody>();
                rigidbody.isKinematic = false;
                rigidbody.constraints = RigidbodyConstraints.FreezePositionY
                    | RigidbodyConstraints.FreezeRotation;

                // 灯籠スクリプトが起動後HalfPoint2との当たり判定をしないように
                if (tourouScript.enabled == true)

```

```

        {
            tourouScript.m_RunOnceFlag = false;
        }
    }

    // Flowスクリプトを追加する (OnTriggerStayがスクリプトを停止させても
    // 動いてしまうため、一時的に削除したので付けなおす必要がある)
    foreach (GameObject river in
GameObject.FindGameObjectsWithTag("Flow"))
    {
        river.AddComponent<Flow>();
    }

    // BuoyancyスクリプトとBuoyancyQuataスクリプトを起動させる
    foreach (GameObject tourouQuatate in
GameObject.FindGameObjectsWithTag("TourouQuatate"))
    {
        Buoyancy buoyancy = tourouQuatate.GetComponent<Buoyancy>();
        BuoyancyQuata buoyancyQuata =
tourouQuatate.GetComponent<BuoyancyQuata>();
        buoyancy.enabled = true;
        buoyancyQuata.enabled = true;
    }
    // フェード演出を終了させる
    m_HalfPoint2.m_EndFireworksFadeFlag = false;
    // カメラの追従を戻す

GameObject.FindGameObjectWithTag("MainCamera").GetComponent<Chasing_Camera>().enabled = true;

    // HalfPoint2を止める

GameObject.FindGameObjectWithTag("HalfPoint2").GetComponent<HalfPoint2>().enabled =
false;

    // このスクリプトを停止させる
    this.enabled = false;
}

```

```

    }

}

/// <summary>
/// 乱数を関数で管理
/// </summary>
private void RandomRange()
{
    m_PrefabsNumber = Random.Range(0, 2);
    m_RandX = Random.Range(-30.0f, 30.0f);
    m_RandY = Random.Range( 0.0f, 40.0f);
    m_RandZ = Random.Range(-30.0f, 30.0f);
}

/// <summary>
/// 生成処理
/// </summary>
private void InstantiatePrefab()
{
    // 花火を 1 ～ 5 個ランダムに生成
    for (int i = 0; i < Random.Range(1, 5); i++)
    {
        // 乱数生成メソッド
        RandomRange();

        if (m_InstantiateCounter < m_HalfPoint2.m_TourouCount)
        {
            // 花火を生成する位置と花火の生成
            GameObject instanceObj = Instantiate(m_HanabiPrefabs[m_PrefabsNumber]);
            instanceObj.transform.position = transform.position;
            instanceObj.transform.position += new Vector3(m_RandX, m_RandY,
m_RandZ);

            // 花火を生成した数をカウントする

```

```
        m_InstantiateCounter = m_InstantiateCounter + 1;
    }

    if (m_InstantiateCounter > m_HalfPoint2.m_TourouCount)
    {
        GameObject.Instantiate(null);
    }
}
}
```