```
// GoalSmoothMove.cs

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

public class GoalSmoothMove : MonoBehaviour
{
    [SerializeField]
    private Transform m_GoalTargetTransform;            //  ゴールオブジェクト
    [SerializeField]
    private float m_Duration;                           //  遷移するまでの時間
    [SerializeField]
    private float m_RotateSpeed;                        //  回転スピード
    [SerializeField]
    private float m_MaxSpeed;                           //  最大速度

    private Vector3 m_MoveVelocity;                     //  移動速度
    private float m_XVelocity;                          // X軸を移動する際の速度
    private float m_YVelocity;                          // Y軸を移動する際の速度
    private float m_ZVelocity;                          // Z軸を移動する際の速度
    private float m_StartTime;                          //  遷移開始時間

    public bool m_SmoothFlag;                           //  遷移フラグ
    public PlayerAnime m_PlayerAnime;                   // PlayerAnimeスクリプト
    public bool m_FlagFlag = false;

    // Use this for initialization
    void Start()
    {
        m_PlayerAnime = GameObject.Find("Player").GetComponent<PlayerAnime>();

        m_StartTime = Time.time;
        this.enabled = false;
    }

    // Update is called once per frame
    void Update()
    {
        TargetSmoothlate();
    }
    /// <summary>
    /// カメラがターゲットの位置と角度に遷移にする
    /// </summary>
    public void TargetSmoothlate()
    {
        float time = (Time.time - m_StartTime) / m_Duration;
        // ターゲットの位置を指定し遷移
        float xPos = Mathf.SmoothStep(transform.position.x, m_GoalTargetTransform.position.x, time);
        float yPos = Mathf.SmoothStep(transform.position.y, m_GoalTargetTransform.position.y, time);
        float zPos = Mathf.SmoothStep(transform.position.z, m_GoalTargetTransform.position.z, time);
        // ターゲットの角度を指定
        float xRotate = Mathf.SmoothDampAngle(transform.eulerAngles.x, m_GoalTargetTransform.eulerAngles.x,
ref m_XVelocity, m_RotateSpeed, m_MaxSpeed, time * Time.deltaTime);
        float yRotate = Mathf.SmoothDampAngle(transform.eulerAngles.y, m_GoalTargetTransform.eulerAngles.y,
ref m_YVelocity, m_RotateSpeed, m_MaxSpeed, time * Time.deltaTime);
        float zRotate = Mathf.SmoothDampAngle(transform.eulerAngles.z, m_GoalTargetTransform.eulerAngles.z,
ref m_ZVelocity, m_RotateSpeed, m_MaxSpeed, time * Time.deltaTime);


        if (m_PlayerAnime.m_SmoothFlag == true)
        {
            m_SmoothFlag = true;

            if (m_SmoothFlag)
            {
                // ターゲットの位置に遷移させる
                transform.position = new Vector3(xPos, yPos, zPos);
                // ターゲットの角度に合わせる
                transform.eulerAngles = new Vector3(xRotate, yRotate, zRotate);
            }
        }

        // ターゲットの場所に遷移が出来ていたら
        if (transform.position == m_GoalTargetTransform.position && transform.rotation ==
m_GoalTargetTransform.rotation)
        {
            m_SmoothFlag = false;

            if (m_SmoothFlag == false)
            {
                m_PlayerAnime.m_GoalAnime = false;
                m_FlagFlag = true;
```

```
                        }
                    }
                }
            }
```