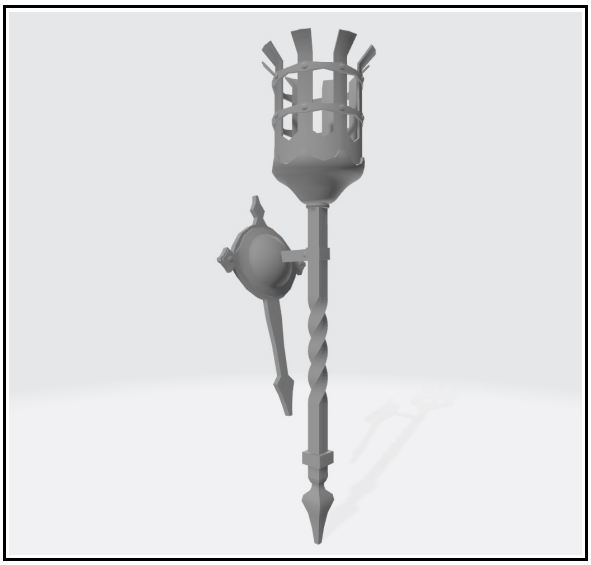
# Old Torch

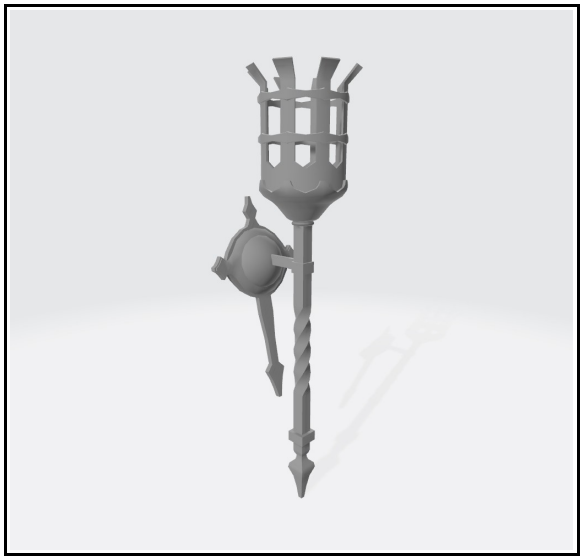
## Models

This little asset contains two models of old torch; Low poly with 1187 faces and High poly with 2115 faces. Each model has two sets of textures that can be used in ***HD Render Pipeline*** (BaseMap, MaskMap and Normal map) and ***Universal Render Pipeline*** (Albedo, Metallic and Normal map).

The textures size are 1024x1024 for low poly torch and 2048x2048 for the high poly torch.



*(mesh Torch highPoly)*



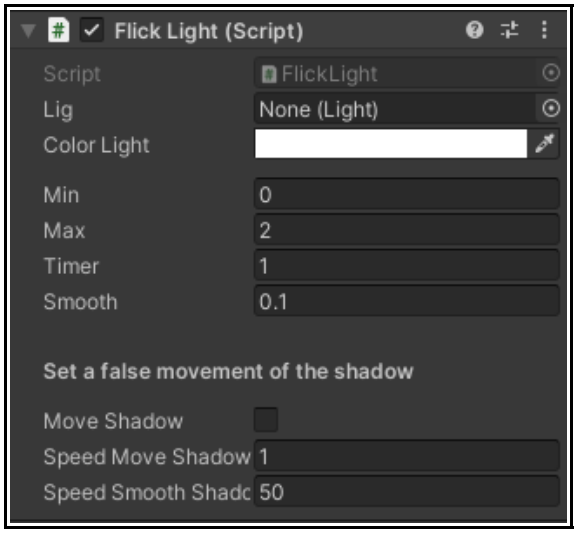
*(mesh Torch lowPoly)*

These two models contemplate a particle system for the flame (containing an animation based on sprite images) already inserted where it's possible modify its properties.

# Script

For more realism it's possible attach a game object Light (not included in the prefab) as a child to the Particle System to simulate the light emitted by flames. In this light can be added a component script called "Flick Light" which allows us to change ramdomly the intensity of the light recreating the typical flicker effect of the flames.

The concept behind this script is very simple; Determine two values to find a random value of intensity of light over time (Update method) simulating a "flicker" light typical of a flame. A random movement of the light inside a "sphere" has been added to simulate the movement of the shadow cast by the light on the surfaces




**Flick Light script**

**Lig:** the light game object where the script is attached.

**Color Light:** Here you can change the color of the light.

**Min** and **Max:** The minimum and maximum values of the light intensity determined randomly.

**Timer:** The timing of the speed for flick Intensity of the light.

**Smooth:** The waiting time for the light to flicker.

**Move Shadow:** Toggle the option to make a random movement of the light inside a "sphere".

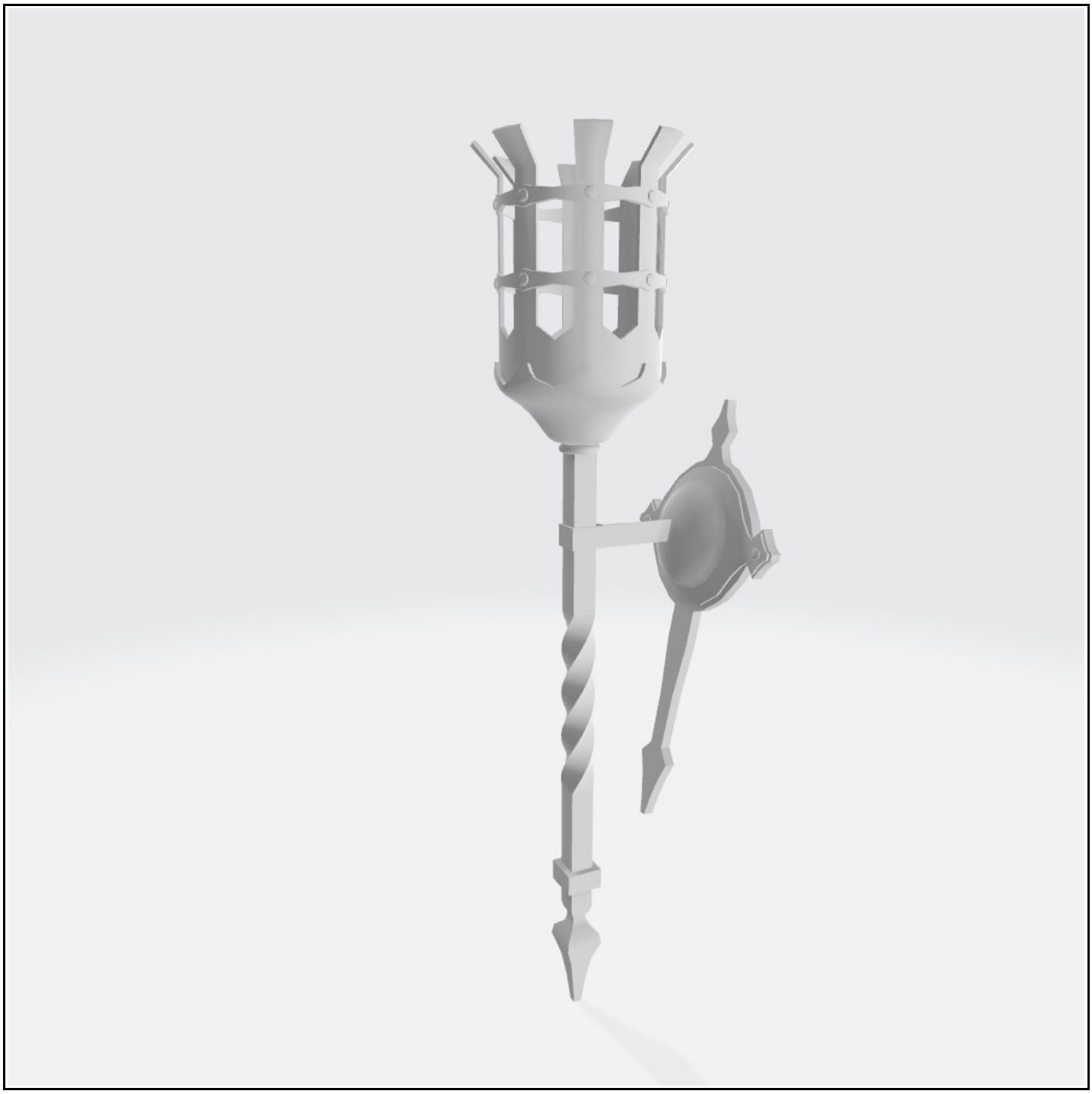
**Speed Move Shadow:** The speed of the movement of light.

**Speed Smooth Shadow**: The speed smooth of the movement of light.

# Script in HDRI

If you want to use the script in **HDRP** you must change the type of light in “HDAdditionalLightData” and add the collection UnityEngine.Rendering.HighDefinition because otherwise in **URP** it doesn't work returning an error. How to do? In Unity create a new script called “HdriFlickLight” and copy/paste the script below:

```
using System.Collections;
using UnityEngine;
using UnityEngine.Rendering.HighDefinition;

public class HdriFlickLight : MonoBehaviour
{
    Vector3 _startPosLight;
    public HDAdditionalLightData lig;
    public Color colorLight = Color.white;
    public float min = 0.0f;
    public float max = 2.0f;
    private float _flickIntensity;
    public float timer = 1.0f;
    public float smooth = 0.1f;
    public bool moveShadow = false;
    public float speedMoveShadow = 1f;
    public float speedSmoothShadow = 50f;

    // Start is called before the first frame update
    void Start()
    {
        if (lig == null)
        {
            lig = GetComponent<HDAdditionalLightData>();
            _startPosLight = lig.transform.position;
        }

        StartCoroutine(SmoothFLick());
    }

    // Update is called once per frameIn Unity create a new script “”
    void Update()
    {
        if (lig == null)
            return;

        StartCoroutine(SmoothFLick());
        MoveShadowLight();
    }

    private IEnumerator SmoothFLick()
    {
        _flickIntensity = Mathf.Lerp (_flickIntensity, (Random.Range(min, max)), timer *
Time.smoothDeltaTime);
        lig.intensity = _flickIntensity;
        lig.color = colorLight;
        yield return new WaitForSeconds(smooth);
    }

    void MoveShadowLight()
    {
        if (moveShadow)
        {
            lig.transform.position = _startPosLight + (Random.insideUnitSphere * speedMoveShadow /
speedSmoothShadow);
        }
        else
        {
            moveShadow = false;
            lig.transform.position = _startPosLight;
        }
    }
```



*All models are recreated from real references. Modeled by Marcelli Michele in Blender® and textured in Substance Painter® (Steam version)*