Low Poly FIRE & SMOKE

Guide

1. Project structure

Editor folder includes an editor script. It is used to create a new layer and automatically assign it to all the prefabs. Will be discussed further.

Mesh folder contains all the meshes used in this asset.

Prefabs folder has all the prefabs in it.

In ScriptsAndShaders you, as the name suggests, might find all scripts and shaders used for creation of this asset.

And, finally, the textures folder contains all the materials. Further updates may contain textures.

2. Lighting

The best look can be achieved using the real-time lighting. This is especially important for smoke. Texture-baked light does not allow for particles rotation and looks much less impressive in general.

However, for optimization purposes some games do not use dynamic lighting or use it selectively. Or during the night there may not be a directional light at all. In such case layers come in handy. A layer LP_Fire is created automatically and assigned to all prefabs. Night_Fire_Light fire prefab then can be used to make particles look more volumetric. Of course, it is not necessary to use it only during the night.

And you can setup your own light that affects only fire and smoke by setting culling mask to LP_Fire.

Of course, the use of these extra directional lights is not necessary and great look can be achieved with point lights.

If any sort of real-time lighting is unacceptable for your game please contact me.

3. Layers

As was mentioned before there is an editor script designed to assign layer automatically. However, if you imported this asset not to the root folder, the script wouldn’t be able to locate it. In this case you can either modify the asset path (lines 16 and 17 of LayerController script) or do it manually:

- Create a layer called “LP_Fire”
- Assign this layer to all prefabs (and children)
- Set culling mask of Night_Fire_light children to “LP_Fire”

4. Performance

The performance was stresstested on a 3 years old Chinese mobile phone (1920x1080 screen, MT6592, 2Gb RAM, MALI-450 MP4, Android 4.2). No problems or FPS drops were observed.

However, if you notice any performance issues or you have a very challenging scene with many particle systems please contact me and I will do my best to help you and if necessary provide you with a simplified, high performance shader.

5. Scaling script

To scale a particle system just drag and drop a “scaler” script on a particle system on the cene (only on a parent, children are scaled automatically), click play and use a slider to adjust a size based on your need.

6. Mesh Randomizer

This package includes a handy low poly mesh randomization script. Simply throw it on an object. I believe public variables are self-explanatory. Just choose if a certain parameter should be randomized for the object or\and for children and set the randomization value.

If you have any questions\problems, feel free to contact me: stanislavdol@gmail.com

Thank you for purchasing this asset.