Developped on Unity 2020.3.35f 1 (LTS) - HDRP　　Théo PAUMARD

ThermalVision

Add this folder to the ‘Assets’ folder of your Unity Project.

This folder contains 5 scripts and a resources folder.

Step 1 : Copy all the files from the TATV-TrueAutomatedThermalVision/Resources folder and paste them to the Assets/Resources folder of your project.

(if the ‘Resources’ folder doesn’t exist in your Project/Assets/ folder, create it)

Step 2 : **Add a layer called ‘** Thermal **’ in the Unity Editor > Inspector > Layer > Add Layer.**
OR Unity Editor > Layers (on the top right corner of the window) > Edit Layers.

Step 3 : Add the script ‘ThermalVisionCam.cs’ to your current Camera. Adjust the output displays of your Thermal Vision and Basic Vision as you wish.

(Default parameters are Display 1 for Thermal Vision and Display 2 for Basic Vision)
(Choosing the same display will create a splitscreen, but you may need to adjust your editor game window resolution)

Step 4 : Add the script ‘ObjectSelector.cs’ to your game object/parent that you want to appear in thermal vision, and adjust the temperature of the object.

(Default temperature is 80)

Now you just have to run your scene and you will have a Thermal Vision and a Basic Vision on the displays you chose.

Specific situations

- Oceans : For the rendering of water in thermal vision, you must mot apply the ‘ObjectSelector’ script, but the ‘OceanThermalSelector’ script. It will create a planar Game Object with a simulation of moving waves as realisticaly seen in Thermal Vision.

- Terrains : Unity Terrains contain complexes forms of Objects, so you cannot use the ‘ObjectSelector’ script over it. So, you have to use the TerrainThermal script over it. It contains one public parameter called ‘*leaf_material_position*’ which references the position of the leaf material in the prefabs trees used by the terrain. In fact, the behaviour of the leaves must be considered differently as an opaque object. Most of the time, the material of the leaf will be in the 2nd position, so index 1.

Algorithm explanation

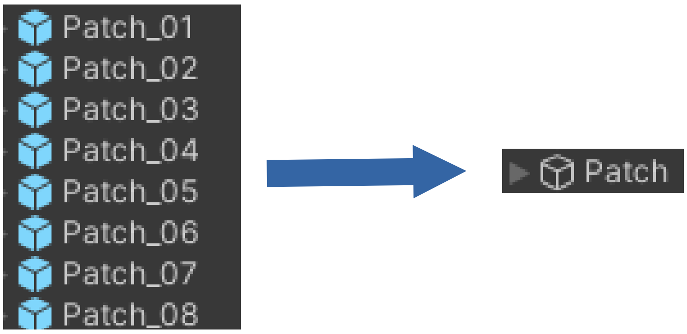
The ‘ThermalCamera’ script, once applied, will create a second camera displaying the real vision, while the current camera will be set to see only the objects in a Thermal layer.

The ‘ObjectSelector’ script will duplicate the current object so that we can still have access to the real vision. It will then apply to every cloned object the ‘ObjectThermal’ script. **So you don’t need to apply manually the ‘ObjectThermal’ script.**

The ‘ObjectThermal’ script will put the current object in the ‘Thermal’ layer, for the Thermal Camera to see it. Then it will get the Thermal Material from the Resources folder, and apply it to every materials of the Mesh, considering the temperature given for the object. If the current object doesn’t have a Mesh, the script will duplicate to every children of the GameObject.

Tips

As the ‘ObjectSelector’ and ‘ObjectThermal’ scripts work recursively through all the cildren of the object selected, you should first put all the objects of the same temperature in a same folder GameObject.



Then, you just have to apply the script once to the root GameObject to access each of its children.

Common Thermal Vision mistakes

The Object Selector script is applied on all the objects, but nothing happens :

- Check the console : Did you move the resources to the Assets/Resources folder ?
- In the resources folder, is the Material « HDRP_Thermal_Mat » linked to the « HDRP_Shader_Graph » shader ? However it could appear pink.
- Did you put the Thermal Camera script over your Camera ?

- If you use the 'OceanThermalSelector', check in the resources that the Animator and the AnimatorController are linked together (if you open the AnimatorController window, the Animator component should appear. However, drag it in into and adjust the animation speed).

Same with the Thermal_Water material, and with the water_normal_map.jpeg attached to the 'normal' parameter of it.

Your objects appear either all bright white or black :

- Check the HDRP_Shader_Graph and check if the 'gradient' node goes from black to white. If it goes all white (as it sometimes randomly does), modify it by adding a black point at the beginning.

Your trees in the distance belonging to a terrain appear blue :

- Increase the parameters : 'billboard start' and 'fade length' in your terrains parameters