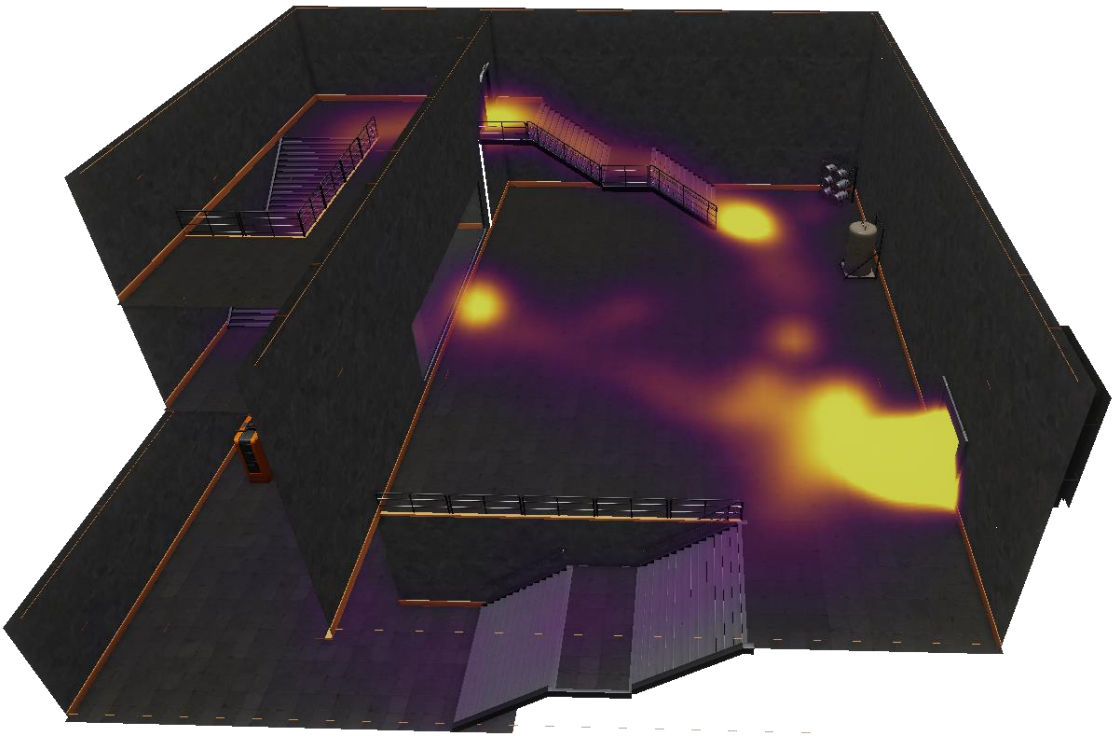
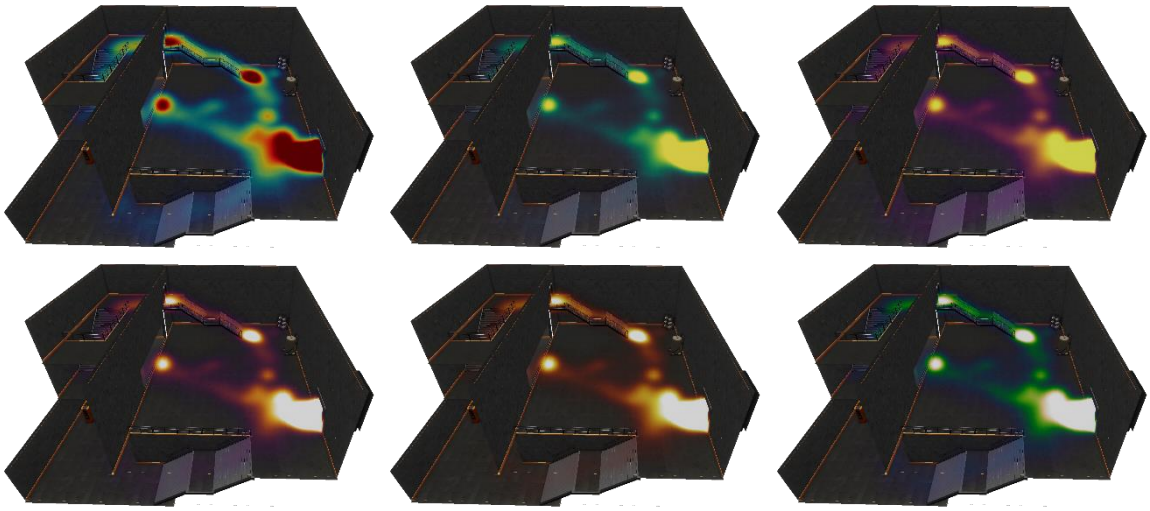
Heatmap Visualization Documentation





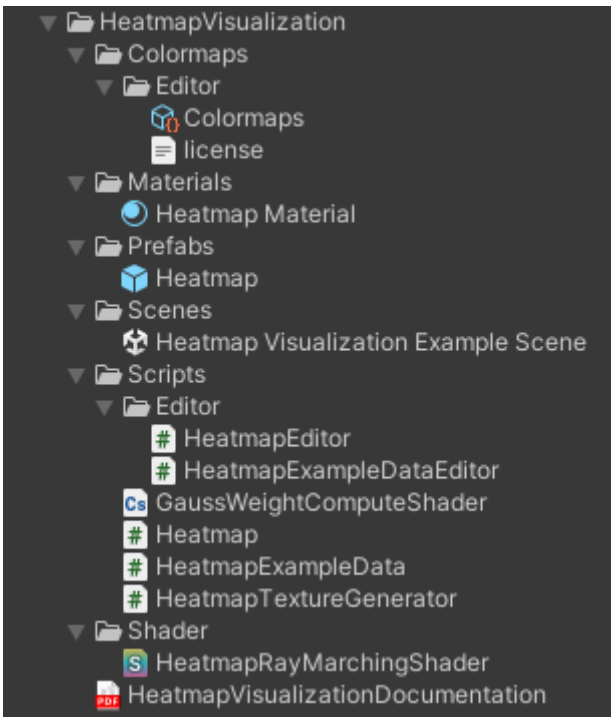
Version: 1.0.0

Contents

1. General Description

The Heatmap Visualization Package can be used to analyze Data gathered from play testers, beta testers or just players. It works entirely in edit mode, so no need to enter play mode.

2. Important Files



- `Prefabs/Heatmap.prefab`
  This is the main Prefab, use it to generate and show your heatmap.

- `Scripts/Heatmap.cs`
  This is the main Component on the main prefab.

- `Scripts/ HeatmapTextureGenerator.cs`
  In here is the class, which generates the `Texture3D` used for rendering the Heatmap.

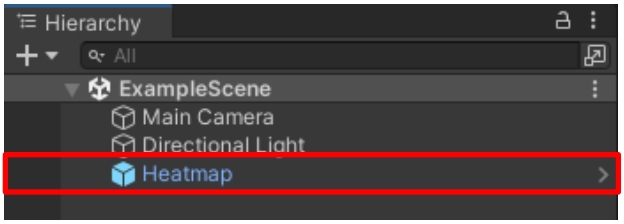
- `Scripts/Editor/HeatmapEditor.cs`
  The Editor Script for `Heatmap.cs`.

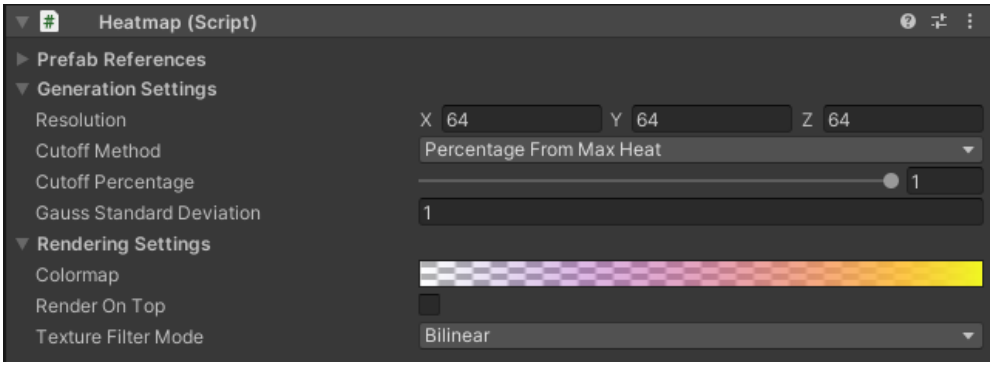
- `Colormaps/Editor/ Colormaps.gradients`
  This file holds some gradients you can use.

3. How to use

3.1 Heatmap Prefab

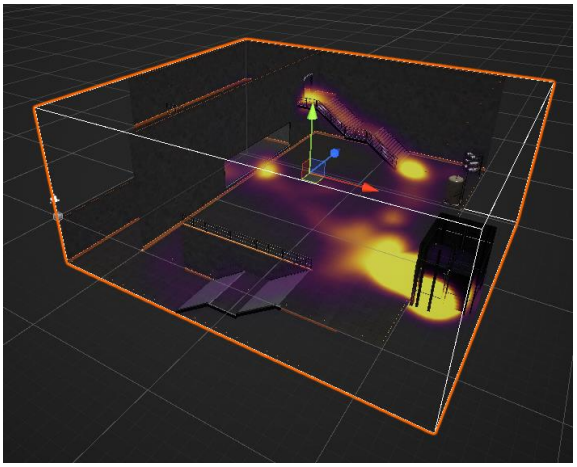
1.1 Place the heatmap prefab in your scene.



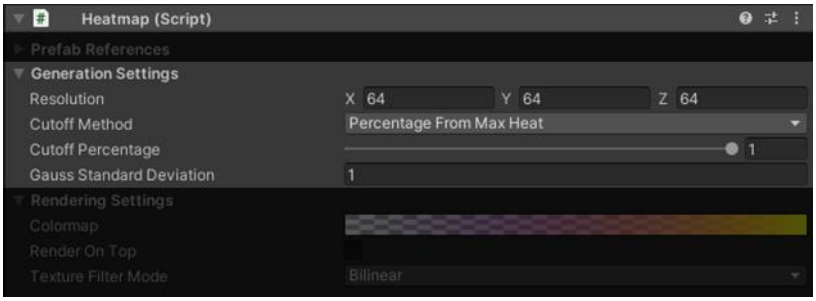


3.2 Heatmap Area

Adjust the position and scale of the prefabs transform, so it matches your map or world.



3.3 Heatmap Generation Settings



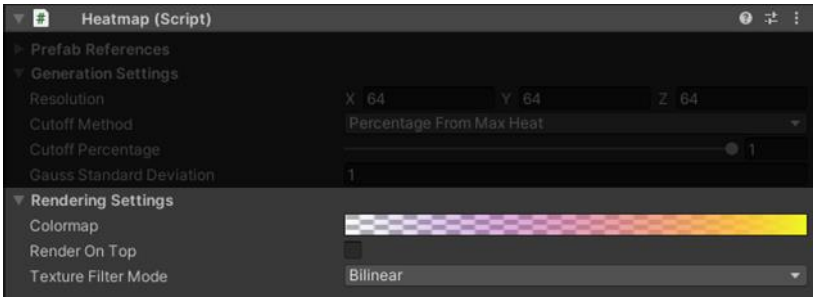
- **Resolution**
  The Resolution of the generated Texture3D. The Higher the Resolution the longer it takes to generate. On my System I can get it up to 512x512x512. I recommend starting with lower resolutions like 64x64x64 and increase from there.

- **Cutoff Method**
  Usually there is a lot of Datapoints at the Spawn point. This can prevent other areas from being visible at all. Therefore, you need to set a threshold to cut off outliers.
  - Percentage From Max Heat
    The maximum heat is at the set percentage of the highest value.
    This is very fast to calculate, therefore recommended.
  - Percentage From Datapoints
    The hottest percentage of datapoints will be ignored. Then the maximum heat is set to the highest value. (Zero values do not contribute to the percentage)
    This method is more reliable but very slow on higher resolutions.

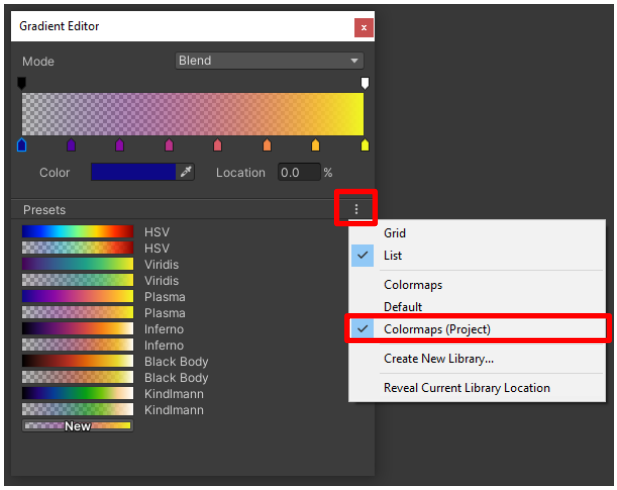
- **Cutoff Percentage**
  The percentage used for the chosen cutoff method.

- **Gauss Standard Deviation**
  The Influence range of each datapoint. The Influence is in the form of a normal distribution function.

3.4 Heatmap Rendering Settings



- **Colormap**
  The gradient used as colormap from cold to hot. Six different colormaps are included in the Package.



- **Render On Top**
  If false, heat values behind other object will be ignored.

- **Texture Filter Mode**
  point, bilinear or trilinear

3.5 Generate Heatmap

Call the `GenerateHeatmap(List<Vector3> points)` Function of the Heatmap Component on the prefab in your scene. It only expects a List of points. Any points out of bounds will be ignored.

Part of the package is the `HeatmapExampleData` Script, which generates random datapoints.

4. Performance and Restrictions

The Cost of the heatmap generation is dependent on the texture resolution and the data point count. I was able to generate Heatmaps up to 256x256x256 and 1 000 000 data points in about a second. This can be achieved due to a compute shader taking over the most expensive part of the pipeline. On higher resolutions, the compute shader might reach the graphic drivers timeout (the timeout can be changed in a setting usually).

5. Colormaps

More Information about the used colormaps can be read on this post:
https://www.kennethmoreland.com/color-advice/