Debugging the graphic performance of a level is vital to understand which parts are causing the bottleneck. It can also be used to find bugged/incorrect meshes or entities.

Start Depth.exe in Dev mode and then press F1 to bring up the Debug Toolbar. The two groups that are important are the "Debug texts" and "Graphics Debug".

Debug texts

The important checkboxes here are Show FPS, Show Memory Usage and Show Render Info. These will tell you how costly

the rendering of the scene is.

Show FPS

Shows how fast the level is running.

- **FrameTime**: Shows how long it takes to render a frame, this is the most important info.
- **FPS**: 1000 / FrameTime. It is what you usually see when talking about performance. It is easier to grasp but not as useful as FrameTime.

Show Memory Usage

- **Renderer Memory Usage**: The amount of VRAM used by the renderer. This contains frame buffers, shadows, terrain and post-effects. It is greatly affected by the resolution of the window.
- **Texture Memory Usage**: The texture memory used by the level. This contains all the textures used for the materials of the objects. Texture memory is reduced by a lot with just a few changes in the config file.
- **Vertex Memory Usage**: The amount of VRAM used by meshes and decals. The imporant difference between Vertex and Texture memory is that Vertex memory can't be reduced by changing a setting. It takes up as much space on a good and a bad computer.
- **Total Memory Usage**: The sum of the above memory usage.

Show Rendering Info

- **Draw Calls**: The number of objects rendered in the current scene.
- **Rendered Triangles / Vertices**: The total number of triangles and vertices rendered per frame.
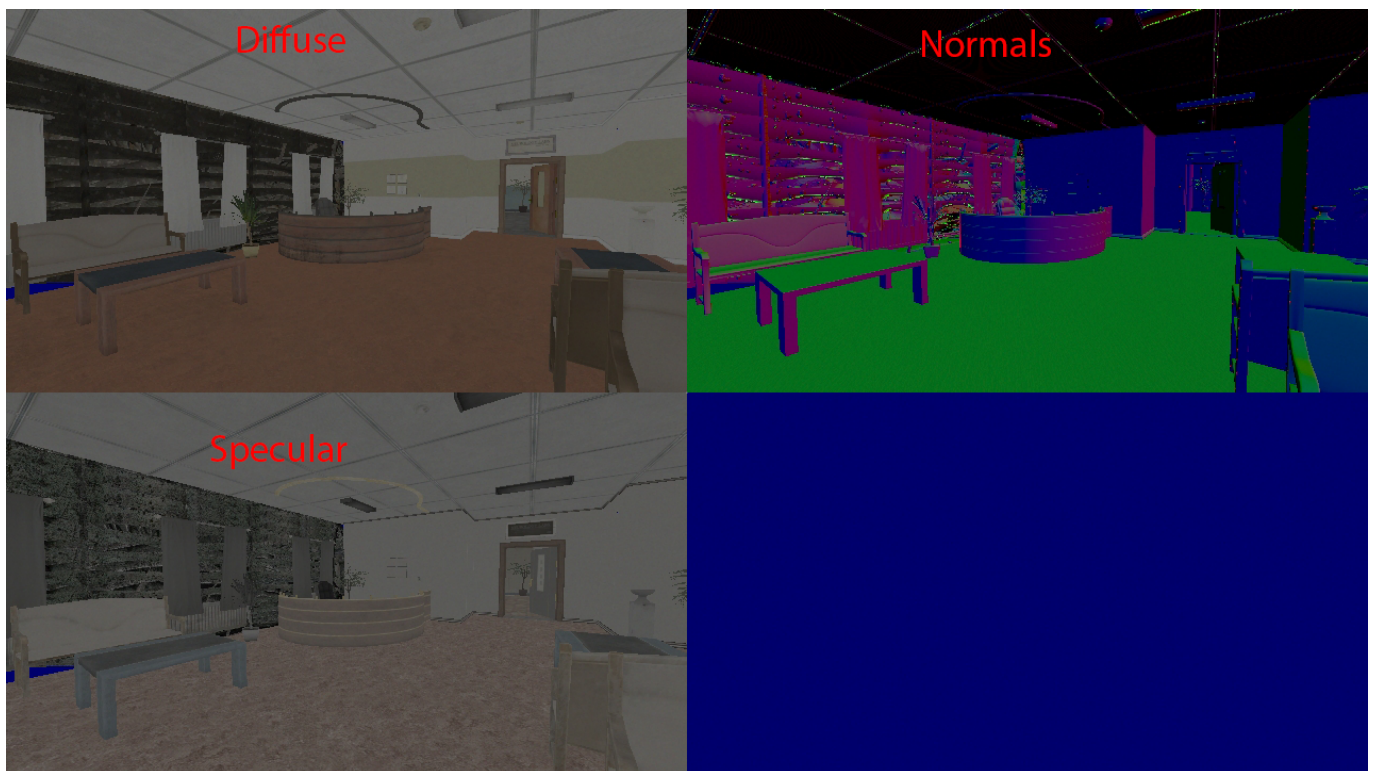- **Queries**: A special draw call used to check if an object is occluded. It is more expensive than a draw call.

Graphics Debug

To be able to find which parts of the scene that takes up the most performance you can use these special render modes.
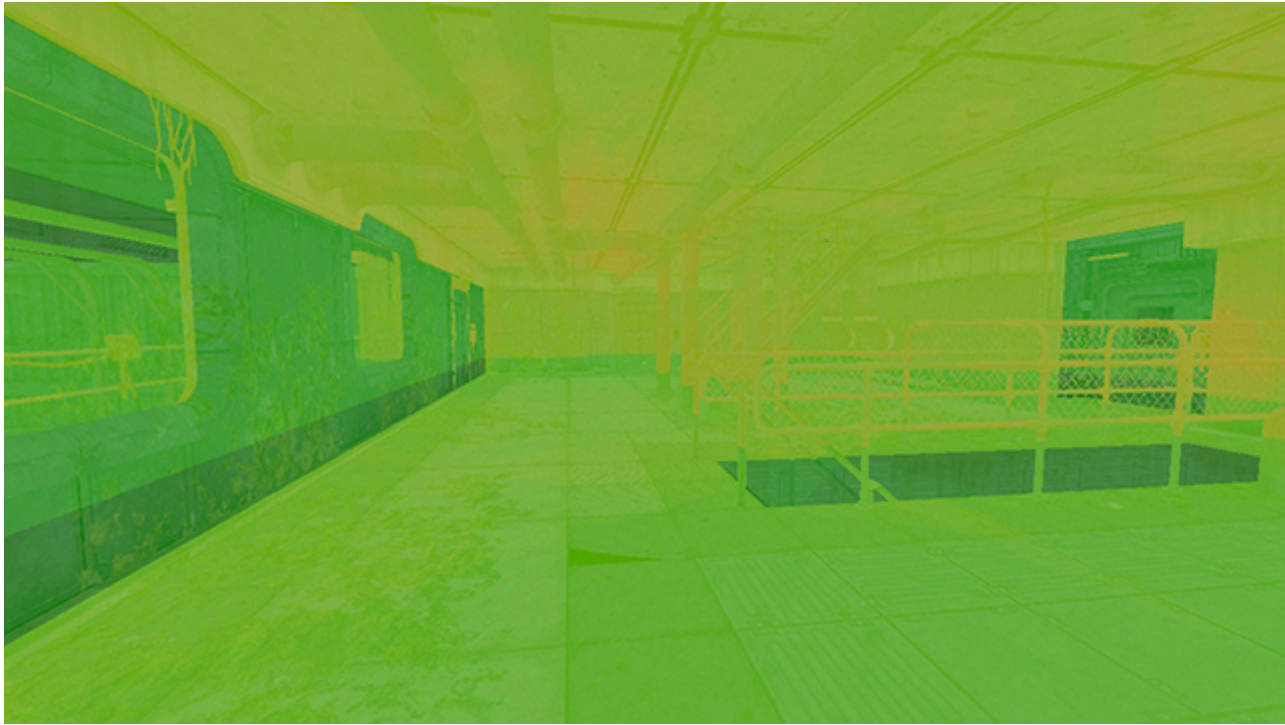
Previous Frame Occlusion

GBuffer

Light Complexity

Overdraw