Normal Maps

Next, we have the normal map. The normal map defines details on the surface of the material seperate to those of the mesh.

If you're reading this guide, you might not know what a normal is. Let me explain it for you! You see, meshes are made up of vertices, points in space, which make up polygons. And a polygon in a mesh is more than just three points in space. Stored throughout it is information regarding which direction the polygon is facing - the normal. This is what decides the lighting of meshes. Bad normals mean weird seams and warping that don't match our expectations. Good normals mean things look soft and round in all the right places, and hard and sharp elsewhere!

Think about meshes themselves. You can put a texture on a flat simple plane and cover it in colours, but it'll still be a simple plane underneath. So, someone had the great idea to use a texture to shift the normals around, covering it in details. A texture can be more abstractly detailed than a mesh, right? And that's what a normal map is.

In Unity, these are stored in a special format once imported, which is why you need to mark new normal map textures imported *as* normal map textures. Unity will normally show a warning if a texture is assigned as a normal map without being imported as a normal map.

If you have a normal map from another game engine, it might be oriented differently. You can tell because the ridges on one axis will look like bumps, and vice versa. In particular, normal maps made for Unreal Engine will always be inverted. In the current version of Unity, you can fix this once the texture is set to a normal map by checking the "Flip G channel" option.

If you're looking at assets without normal maps, there's no need to panic, because a normal map isn't necessary if you don't have one. It only adds lighting detail. You can generate them yourself if you want, see below.

- [More details on normal map orientation, and a handy table of which applications support and export what by default.](#)