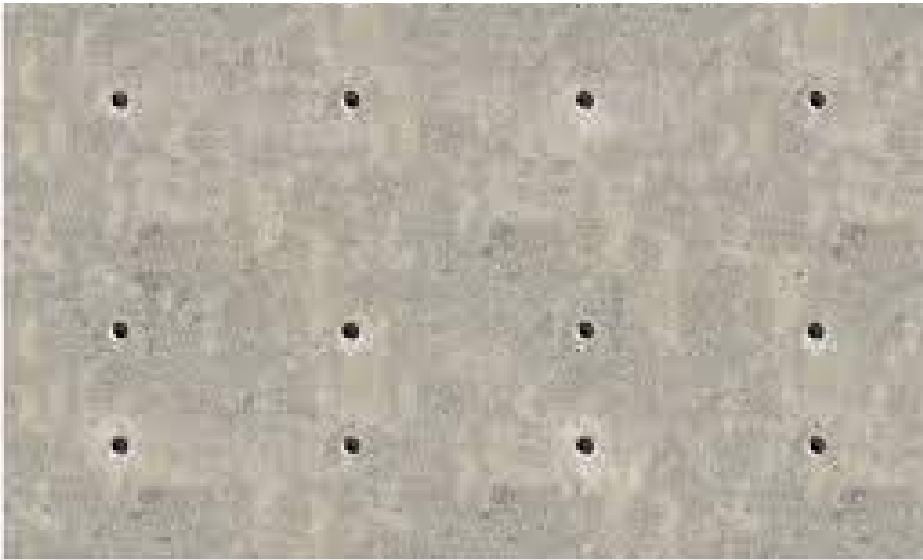# Creating your own textures for materials

About

This section involves creating your own textures for materials used in engines such as Unity or Unreal Engine. Often we use materials from Quixel Megascans, AmbientCG, Adobe Substance, or some asset pack, but it is worth knowing how to make your own textures.

The guides listed focus mostly on building what's called **seamless textures**, or textures that *can* tile ov_____he seams that connect the textur

Non seamless... Ouch!

Seamless (much better)

Getting Started

Begin with gathering a high quality photo of your soon to be texture. There are many, many guides on how to take the photo, but if you do not have a DSLR camera, an iPhone camera will still do great. Look out for things such as good lighting, no shadows being cast, and keeping your photo shot straight and perpendicular. Take many photos if possible for backups and references (e.g. find other tufts of grass, or turn over the wood plank, etc.)

Photoshop

These YouTube videos that will explain the process. Feel free to search more as well, or find one specific to the texture you are creating.

Base color textures:

- https://www.youtube.com/watch?v=owm0R2uDMtY
- https://www.youtube.com/watch?v=vE0dR-TWVg4

Normal Maps:

- https://www.youtube.com/watch?v=YJqWHsllczY

Normal Map (generator)

- https://cpetry.github.io/NormalMap-Online/

> Be careful of using generators. They are approximations based off the image received as input and the algorithms can't determine that your wood photo is actually wood.

Where to go from here?

This guide covers taking photos, editing photos into seamless textures, and creating normal maps. Consider researching other photoshop tutorials that cover maps such as roughness and metallic.

Other maps, such as shadows, ambient occlusion, cavity, can be generated in modelling tools like Blender or Maya. I recommend assigning these base color and normal maps to your modelling software's materials and producing these additional maps.

When completed, always test your maps and models in their final rendering destination. For SCiL, it is mostly Unreal Engine or Unity.

Revision #3
Created 11 July 2024 18:46:39 by Wes
Updated 17 January 2025 14:56:23 by Wes