Render Passes

This document merely describes a render pass, to see where it fits in with the rendering system, see the render system document.

A render pass should have a single framebuffer the renderers would draw to. The frame buffer **must** be bound before submitting to the renderer as the renderer will not bind the framebuffer for you.

An entity list will be passed to the render pass, this contains the entire entity list which has a renderable component. Check the entity’s display variable here.

The scene contains the order of the passes, so if a pass is reliant on another having been rendered, check the order in the scenes config file.

The render pass also has an enabled bool which determines whether the render pass is to be processed, this can be useful to have a pass in the pass list which sometimes you want to use and sometimes you don’t. Set the value of this with the **setEnabled** function.

**Create new render pass**

1. Create a new subclass which inherits from render pass
2. Declare the framebuffer in subclass or list of framebuffers
3. In the onRender, define the behaviours, clear the buffers, set the OpenGL states, etc
4. Update the function **createRenderPass** in the sceneLoader class to load the render pass into the scene
5. Update the scene’s config file to load the renderpass for the scene, order is important