Maat Graphics
Design Goal

Lilith Wynter

7th September, 2018

Contents

1 Introduction

- "User" refers to the programer using the Maat-Graphics.

2 Graphics API

2.1 Window and Context Handling

2.1.1 Window.rs

The purpose of this is to handle everything to do with the window, holds both Vulkan and OpenGL versions. It is one of the lowest files in the program. This shouldn't only update with Winit and Glutin crate updates and when currently unused window functions are needed in a higher context.

- Handles window creation.

- Handles window resize.

- Gets drawing context.

- Gets the event loop.

- Show/Hide Mouse.

Dependencies

- None

2.1.2 Graphics.rs - CoreRender

The purpose of this is to simply hold the trait that is shared between openGL and Vulkan that the User has access to all of the trait functions and only difference for the User will be calling the new function from either openGL or Vulkan, where the new functions are defined as the impl of the respective structs.
Elaborated In ResourceManager and respective Graphics API sections.
This will rarely be updated unless there is a fundemental change or a new function is needed to be exposed to the User.

2.2 Vulkan

2.2.1 VkMaat.rs

The purpose of this file is to impl the trait CoreRender from Graphics.rs. the veins of Maat-Graphics.

- Creates VkWindow from Window.rs, passing parameters from Settings.rs

- Creates TextureShader, ForwardShader and FinalShader using the new() from their respective files.

Functions

- clear_screen()
  Does as it says, clears the current framebuffer.

- pre_draw()
  This is used to do anything that needs to be completed before drawing is attempted. In this case recreating the swapchain when nessarily, usually only happens after window creation and screen resizes.

- draw()
  Takes in all the draw_calls, Frame buffers are composited in this order, $TextureShader -> ForwardShader -> FinalShader$.
  Both TextureShader and Forward shader output a single image each, they need to be passed to the FinalShader for drawing.

- post_draw()
  Not used here, but is used for anything that needs to be disabled or unbounded after the entire draw.

- swap_buffers()
  Not used here, is used for openGL to swap current drawing image.

- screen_resized()
  This function should be called everytime the screen is resized and should put in motion nessesary functions for handling screen resizes.

Dependencies

- VkWindow

- ResourceManager

- TextureShader

- ForwardShader

- FinalShader

- Camera

2.2.2 resource.rs

The purpose of this file is to load and handle texture and model resources, most of these functions will be called directly from whatever impl CoreRender.

- Creates a threadpool that will be used to load resources in.

- Holds Vec of LoadableObjects which contains location of file and the reference name, as well as wether or not it has been loaded in or not.

Functions

- new()
  Creates empty Vec of loadableObjects and a ThreadPool

- TODO: add_texture()
  Will add a loadableObject to the Vec without loading it into memory.

- get_texture
  Returns Texture image for use elsewhere if it is loaded in

- load_texture()
  Initates a new LoadableObject and creates a new thread to load the new textures into memory.

- load_texture_into_memory()
  Loads textures into immutableImage format onto the GPU, usually called from a thread.

Dependencies

- None

2.2.3 finalshader.rs

The purpose of this file is to hold the FinalShader struct used in VkMaat.

- Deals with the final frame buffer to draw "directly" to the screen.

- Handles passing uniform variables from cpu to shader, data comes from parameters.

- Loads in the Shader and any image attachments related to this specific framebuffer. (excluding attachments from other Framebuffers).

- Loads in the Vertices and Indices for drawing the single 2D quad used in this framebuffer.

Functions

- create()
  This function is to inits all the data the TextureShader holds, including the framebuffer, uniform buffer, vertex and indices, renderpass (Attachments), and Pipeline (Shader).

- empty_framebuffer()
  discards all avalible images for drawing, thus triggering recreate_framebuffer.

- recreate_framebuffer()
  Creates fresh drawing contexts.

- begin_renderpass()
  Sets the frame buffer to be drawn to

- draw()
  Takes in Both images from TextureShader and ForwardShader and passes them to the shader along with uniform variables and draws quad over the entire screen.

- end_renderpass()
  Finishes the current framebuffer

Dependencies

- VkFinal.vert

- VkFinal.frag

2.3 OpenGl

2.3.1 GlMaat.rs - (Currently rawgl.rs needs renaming)

The purpose of this file is to impl the trait CoreRender from Graphics.rs. the veins of Maat-Graphics.