

Review

- Define the rendering pipeline
- Name components of the typical rendering pipeline and their role
- Illustrate the rendering pipeline using a figure or diagram
- Provide examples of what each component of the rendering pipeline does
- Provide examples that connect directly to the topics seen in class

(2-5 were answered using class notes)

1. The rendering pipeline is the way/path in which visuals and graphics are computed before being displayed on the screen of the computer
2. CPU, API, GPU vert, GPU Assembly, GPU Rasterize, GPU Frag, GPU Framebuffer, Screen
3. CPU → API → GPU vert → GPU Assembly → GPU Rasterize → GPU Frag → GPU Framebuffer → Screen
4. The CPU collects vertex and element data, then through the help of the API it sends this data to the GPU which then puts this data into world space, then GPU uses this data to form triangles, then these triangles are used to determine which pixels on screen need to be used for the render, then the GPU processes each fragment and determines final colours e.t.c., then the framebuffer handles depth and blends them with final colours onto the screen
5. Collects the vertex data of a cube, this data gets sent to the gpu, this vertex data is then computed to determine the vertex positions in world space, this data is then formed into triangles which make up the shape of the object, then these fragments are processed to determine final colours and blended with depth in the framebuffer, this data is then put onto the screen

Fundamentals of deferred and forward rendering

- What are forward and deferred rendering
- What are the typical application of forward and deferred rendering
- How would you explain forward and deferred rendering using a diagram
- Provide and explain examples of forward and deferred rendering

- 1.
- 2.
- 3.
- 4.

Fundamentals of vertex and fragment shaders

- What are vertex and fragment shaders
- What are typical applications of vertex and fragment shaders

- How would you explain vertex and fragment shaders using a diagram
- Provide examples of vertex and fragment shaders

(unable to answer 2/4 questions without assistance)

1. Vertex shaders are shaders that make it appear as though the topology of an object is being changed when in reality it only appears that way as a result of the rendering process. Fragment shaders affect the way in which the object appears in regards to colouring, and lighting.
2. Typical applications of vertex shaders would be something like making the surface of water appear to move in the form of waves/ripples, and an application of fragment shaders would be for controlling lighting and texture.
- 3.
4. Waves or moving grass,