API Walkthrough Data

Direct3D 12

- Fencing between frames was a subject of confusion for the participant. Why was it necessary and how did it function?
- Confusion about numbered type names such as IDXGIFactory4.
- Pointer arithmetics, such as when creating a view to a render target, was difficult to understand.
- Parameter Binding through D3D12RootSignature and descriptors was difficult to understand. What was a descriptor range and how does it relate to a descriptor table?
- Why do resources like vertex buffers need to be defined as subresources, when they contain only one flat resource?
- Confusion about DXGI_SAMPLE_DESC and D3D12_STATIC_SAMPLER_DESC having similar names, while the first is used to define multisampling and the other sampling from texture resources.
- Why do we manually have to serialize the data of a D3D12RootSignature before initializing the object?
- It was confusing that resources could reside on an upload heap or a default heap. Participant expected resources just to be uploaded to one type of memory.

Vulkan

- There was some confusion as to the difference between images and image views. Why was the functionality not implemented as a single class?
- How to enable and use different software layers, such as validation layers, was difficult to understand.
- The semantics of field values was sometimes surprising, such as in the case of the fieldmaxImageCount on SurfaceCapabilitiesKHR. The field stores an upper limit to how many buffers can be in a swapchain, but if it is 0, this indicates that there is no limit.
- Dependencies between subpasses within a renderpass become confusing. The participant did not expect to have to define at which pipeline stages the subpasses should wait for their dependencies. It was also surprising that there is always an implicit subpass taking place, before any of the ones defined by the developer.
- The relationship between DescriptorSetLayout, DescriptorSet and DescriptorPool was not immediately obvious. This also made it difficult to understand how parts of a DescritorSet could be updated by using a dynamic uniform buffer descriptor.
- Pointer arithmetic is difficult to keep track of, such as when when updating descriptors between frames.
- It took a long time to walk through the code, which ensures that the correct graphics device is picked, and that the correct queues, extensions and features are selected and enabled. It was surprising that the swapchain had to be enabled explicitly.