

Vulkan API Concepts

Krisztián Szabó

June 26, 2024

Contents

1	Swapchain	2
---	-----------	---

1 Swapchain

A swapchain object (a.k.a. swapchain) provides the ability to present rendering results to a surface. Swapchain objects are represented by `VkSwapchainKHR` handles. A swapchain is an abstraction for an array of presentable images that are associated with a surface. The presentable images are represented by `VkImage` objects created by the platform. One image (which can be an array image for multiview/stereoscopic-3D surfaces) is displayed at a time, but multiple images can be queued for presentation. An application renders to the image, and then queues the image for presentation to the surface.

When creating a swapchain we have to specify a lot of pieces of data. One of the most important option is the present mode. To describe the functionality of these options, first let us introduce some definitions.

- vertical blank: typically refers to the period during which the display device (such as a monitor) refreshes its image. This process is often referred to as Vertical Blank (VBlank) or Vertical Sync (VSync).
- tearing: Tearing is a visual artifact in video display where a frame is displayed on the screen before the previous frame has finished being drawn, causing parts of multiple frames to be shown at once.

There are four options that are important for us.

1. `VK_PRESENT_MODE_IMMEDIATE_KHR`: specifies that the presentation engine does not wait for a vertical blanking period to update the current image, meaning this mode may result in visible tearing. No internal queuing of presentation requests is needed, as the requests are applied immediately.
2. `VK_PRESENT_MODE_MAILBOX_KHR`: specifies that the presentation engine waits for the next vertical blanking period to update the current image. Tearing cannot be observed. An internal single-entry queue is used to hold pending presentation requests. If the queue is full when a new presentation request is received, the new request replaces the existing entry, and any images associated with the prior entry become available for reuse by the application. One request is removed from the queue and processed during each vertical blanking period in which the queue is non-empty.
3. `VK_PRESENT_MODE_FIFO_KHR`: specifies that the presentation engine waits for the next vertical blanking period to update the current image. Tearing cannot be observed. An internal queue is used to hold pending presentation requests. New requests are appended to the end of the queue, and one request is removed from the beginning of the queue and processed during each vertical blanking period in which the queue is non-empty. This is the only value of `presentMode` that is required to be supported.
4. `VK_PRESENT_MODE_FIFO_RELAXED_KHR`: specifies that the presentation engine generally waits for the next vertical blanking period to update the current image. If a vertical blanking period has already passed since the last update of the current image then the

presentation engine does not wait for another vertical blanking period for the update, meaning this mode may result in visible tearing in this case. This mode is useful for reducing visual stutter with an application that will mostly present a new image before the next vertical blanking period, but may occasionally be late, and present a new image just after the next vertical blanking period. An internal queue is used to hold pending presentation requests. New requests are appended to the end of the queue, and one request is removed from the beginning of the queue and processed during or after each vertical blanking period in which the queue is non-empty.