

Введение в компьютерную графику

Быковских Дмитрий Александрович

07.09.2024

2024-10-18

Введение в КГ

Введение в компьютерную графику

Быковских Дмитрий Александрович

07.09.2024

- Общая информация о компьютерной графике
- Некоторые факты из истории компьютерной графики
- Аппаратные средства, связанные с выводом изображения
- Библиотеки визуализации

- Общая информация о компьютерной графике
- Некоторые факты из истории компьютерной графики
- Аппаратные средства, связанные с выводом изображения
- Библиотеки визуализации

swapchain? Vulkan API has two types of resource: buffers and images
Vulkan images represent contiguous texture data in 1D/2D/3D form. These images are primarily used as either an attachment or texture: Attachment: The image is attached to the pipeline for the framebuffer's color or depth attachment and can also be used as an auxiliary surface for multipass processing purposes Texture: The image is used as a descriptor interface and shared at the shader stage (fragment shader) in the form of samplers the image is created by specifying a number of bitwise fields indicating the kind of image usage, such as color attachment, depth/stencil attachment, a sampled image in a shader, image load/store, and so on. In addition, you need to specify the tiling information (linear or optimal) for the image. This specifies the tiling or swizzling layout for the image data in memory. The notion of texture Image: An image represents the texture object in Vulkan. contains metadata for memory requirements such as the format, size, and type (sparse map, cube map, and so on). multiple images, based on the mipmap level or array layers
An image layout is an implementation-specific way to store image texture information in a grid coordinate representation in the image memory.

2024-10-18

Введение в КГ

swapchain? Vulkan API has two types of resource: buffers and images
Vulkan images represent contiguous texture data in 1D/2D/3D form. These images are primarily used as either an attachment or texture: Attachment: The image is attached to the pipeline for the framebuffer's color or depth attachment and can also be used as an auxiliary surface for multipass processing purposes Texture: The image is used as a descriptor interface and shared at the shader stage (fragment shader) in the form of samplers the image is created by specifying a number of bitwise fields indicating the kind of image usage, such as color attachment, depth/stencil attachment, a sampled image in a shader, image load/store, and so on. In addition, you need to specify the tiling information (linear or optimal) for the image. This specifies the tiling or swizzling layout for the image data in memory. The notion of texture Image: An image represents the texture object in Vulkan. contains metadata for memory requirements such as the format, size, and type (sparse map, cube map, and so on). multiple images, based on the mipmap level or array layers
An image layout is an implementation-specific way to store image texture information in a grid coordinate representation in the image memory. Image view: Images cannot be used directly for reading and writing