

# Vulkan Meetup Japan

学生/研究者からみたVulkan

September 9, 2022 / VIRTUAL

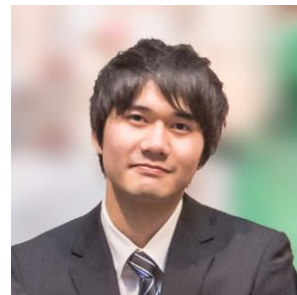
The Vulkan logo is displayed in white on a red background. It features a stylized red swoosh above the word "Vulkan" in a bold, sans-serif font, followed by a registered trademark symbol (®).The Khronos Group logo consists of the word "KHRONOS" in a bold, sans-serif font with a red swoosh under the "O", and the word "GROUP" in a smaller font below it.

WEBINARS  
& MEETUPS

# 自己紹介

西舘 祐樹 | Yuki Nishidate

- 慶應義塾大学大学院
- Vulkanビギナー
- [Vulkan Guide](#)の日本語訳担当
- [Zenn](#)でたまに記事執筆



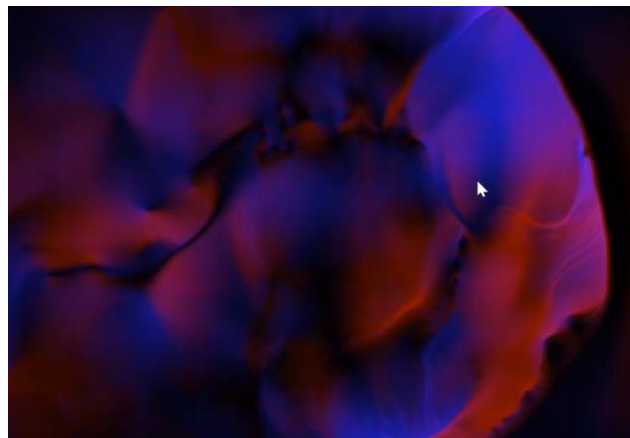
# 発表内容

これから入門する方に向けて  
学生 / 研究の面から経験を共有する

- 個人として「入門してみた」系の内容
- 個人的な感想が含まれる

# Vulkanを何に使っているか

- GPUレンダラーの開発
- GPUシミュレーターの開発
- レンダリングの研究



# なぜVulkanを選んだのか

- GPU自体の勉強になる
- マルチプラットフォーム
- 新しい技術で面白そう！
- レイトレーシング拡張機能！
  - ハイブリッドレンダリングも

# Vulkanの学習方法

- Vulkanの基本的な使い方  
[Vulkan Tutorial](#)
- さらにその先へ  
[Vulkan Samples](#)
- レイトレーシング拡張機能の使い方  
[NVIDIA Vulkan Ray Tracing Tutorial](#)

# レイトレ拡張機能とは

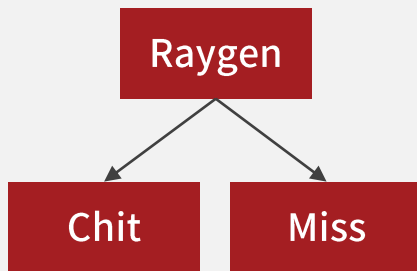
レイトレHWを利用するための機能 [\[詳細\]](#)

- DXRやOptiXと似たような機能
  - レイトレ用のデータ構造のビルド
  - レイとデータ構造の交差判定
- Vulkan 1.2.162以降で利用可能
  - VK\_KHR\_acceleration\_structure
  - VK\_KHR\_ray\_tracing\_pipeline
  - VK\_KHR\_ray\_query

# レイトレ拡張機能とは

2種類の使い方ができる

Ray Tracing Pipeline



複雑な制御が可能

Ray Query

Frag etc.

手軽に使える



# 実装Tips

## [vulkan.hpp](#)のすすめ

- 公式C++ラッパー
- Vulkan SDKに含まれている

```
std::vector<const char*> extensions = ...;
std::vector<const char*> layers = ...;

VkInstanceCreateInfo createInfo{};
createInfo.sType = VK_STRUCTURE_TYPE_INSTANCE_CREATE_INFO;
createInfo.enabledLayerCount = layers.size();
createInfo.ppEnabledLayerNames = layers.data();
createInfo.enabledExtensionCount = extensions.size();
createInfo.ppEnabledExtensionNames = extensions.data();

VkInstance instance;
if (vkCreateInstance(&createInfo, nullptr, &instance) != VK_SUCCESS)
{
    throw std::runtime_error("failed to create instance!");
}

vkDestroyInstance(instance, nullptr);
```

vulkan.h

```
std::vector<const char*> extensions = ...;
std::vector<const char*> layers = ...;

vk::InstanceCreateInfo createInfo{};
createInfo.setPEnabledLayerNames(layers);
createInfo.setPEnabledExtensionNames(extensions);

vk::UniqueInstance instance = vk::createInstanceUnique(createInfo);
```

vulkan.hpp

# 実装Tips

必要なことだけを書く

- あらゆる環境で動く必要はあるか？
- 最高のパフォーマンスが必要か？

# Vulkan入門してみても

技術以外の面について

良い点

- 古い情報が少ない
- コードが綺麗に書ける
  - 研究モチベが保てる！

悪い点

- 日本語情報は少ない
- 学習に時間がかかる
  - 学生の時間は大切に…

# Vulkanやってみる？

- 新しい技術が好き
  - Vulkan試してみましよう！
- OpenGLから移行しないといけない？
  - OpenGLでも問題ないなら無理に移行しなくてOK
- マルチプラットフォームHWレイトレしたい
  - Vulkan一択！

# ありがとうございました

本セッションに関すること以外でも  
Twitterでいつでも話かけてください！

[@yknishidate](https://twitter.com/yknishidate)