

#rtcamp9

Vulkanレイトレのはなし

Nishiki @yknishidate

Vulkanレイトレ、頑張ってます

KHR

VK_KHR_acceleration_structure

VK_KHR_ray_query

VK_KHR_ray_tracing_pipeline

VK_KHR_ray_tracing_position_fetch

EXT

VK_EXT_opacity_micromap

NV

~~VK_NV_ray_tracing~~

VK_NV_ray_tracing_motion_blur

VK_NV_ray_tracing_invocation_reorder

VK_NV_displacement_micromap

基本の3つ

VK_KHR_acceleration_structure

ASをビルドする機能

VK_KHR_ray_tracing_pipeline

レイトレパイプライン。Indirectもいける

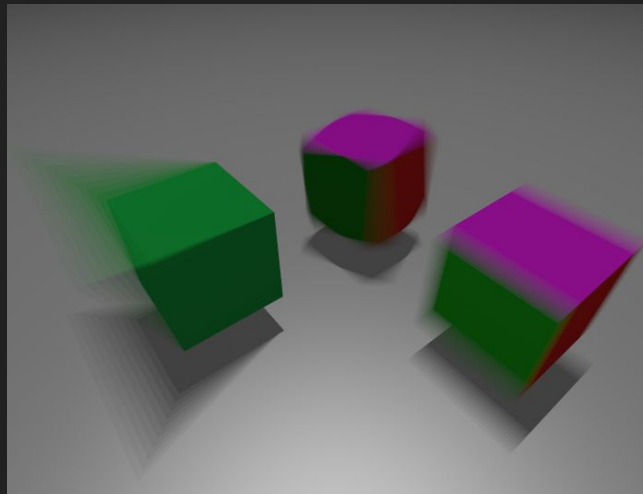
VK_KHR_ray_query

レイクエリ。任意のシェーダ内でレイを飛ばす

ブラーもあります

VK_NV_ray_tracing_motion_blur

```
traceRayMotionNV(  
    topLevelAS,  
    time,  
);
```



github.com/nvpro-samples/vk_raytracing_tutorial_KHR

Shader Execution Reorderingもあります

VK_NV_ray_tracing_invocation_reorder

RTX4000以降

// レイを飛ばす

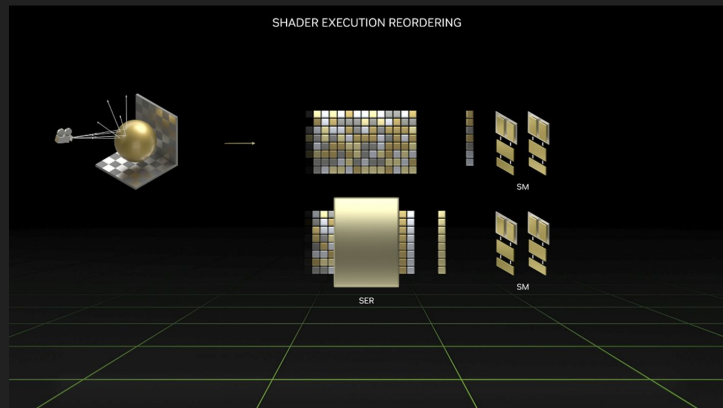
```
hitObjectTraceRayNV(hitObject, topLevelAS);
```

// リオーダーリング

```
reorderThreadNV(hitObject);
```

// シェーダーを実行する

```
hitObjectExecuteShaderNV(hitObject);
```



<https://developer.nvidia.com/blog/improve-shader-performance-and-in-game-frame-rates-with-shader-execution-reordering/>

Micromapが2つ

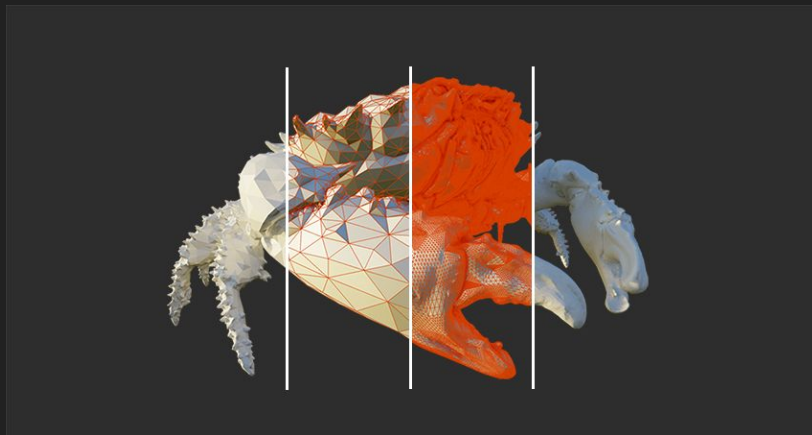
VK_EXT_opacity_micromap

EXT = NVIDIA + Intel (!)



VK_NV_displacement_micromap

micromeshという名前は消えたらしい



最近ちょっと地味な機能が追加

VK_KHR_ray_tracing_position_fetch

```
vec3 vertPos0 = gl_HitTriangleVertexPositionsEXT[0];  
vec3 vertPos1 = gl_HitTriangleVertexPositionsEXT[1];  
vec3 vertPos2 = gl_HitTriangleVertexPositionsEXT[2];
```