Getting Information Back from the Graphics System

Getting Information Back from the Graphics System

• Don’t draw the colors – just draw the depths (especially if the fragment shader is time-consuming)
• Don’t draw the whole bounding volume – cull away the back faces (two reasons: time and correctness)
• Don’t draw the whole object – just draw a simple bounding volume at least as big as the object
• Don’t draw the whole scene – just draw the object you are interested in

Some hints:

• This is commonly used to see what level-of-detail should be used when drawing a complicated object
• This indicates that Vulkan thinks that Queries are time-consuming (relatively) to setup, and thus better to set them up in program-setup than in program-runtime

### Setting up Query Pools

- There are 3 types of Queries: Occlusion, Pipeline Statistics, and Timestamp
- Vulkan requires you to first setup “Query Pools”, one for each specific type

### Pipeline Statistics Query

Pipeline Statistics Queries count how many of various things get done between the `vkCmdBeginQuery` and the `vkCmdEndQuery`

```c
uint32_t counts[NUM_STATS];
result = vkGetQueryPoolResults(LogicalDevice, statisticsQueryPool, 0, 1, sizeof(uint32_t), counts, 0, VK_QUERY_RESULT_WAIT_BIT);
```

### Occlusion Query

Occlusion Queries count the number of fragments drawn between the `vkCmdBeginQuery` and the `vkCmdEndQuery` that pass both the Depth and Stencil tests.

This is commonly used to see what level-of-detail should be used when drawing a complicated object

Some hints:

- Don’t draw the whole scene – just draw the object you are interested in
- Don’t draw the whole object – just draw a simple bounding volume at least as big as the object
- Don’t draw the whole bounding volume – cut away the back faces (two reasons: time and correctness)
- Don’t draw the colors – just draw the depths (especially if the fragment shader is time-consuming)

```c
uint32_t fragmentCount;
result = vkGetQueryPoolResults(LogicalDevice, occlusionQueryPool, 0, 1, sizeof(uint32_t), &fragmentCount, 0, VK_QUERY_RESULT_WAIT_BIT);
```
```
uint64_t nanosecondsCount;
result = vkGetQueryPoolResults(
    LogicalDevice, timestampQueryPool, 0, 1,
    sizeof(uint64_t), &nanosecondsCount, 0,
    VK_QUERY_RESULT_64_BIT | VK_QUERY_RESULT_WAIT_BIT);
```

**Timestamp Query**

Timestamp Queries count how many nanoseconds of time elapsed between the `vkCmdBeginQuery` and the `vkCmdEndQuery`.

```
vkCmdWriteTimeStamp(CommandBuffer, pipelineStages, timestampQueryPool, 0);
```

Even though the stages are “bits”, you are supposed to only specify one of them, not “or” multiple ones together.