VRAM Upload Culling

Description

This chapter briefly describes how to use scope masks for uploading only dedicated content to VRAM.

VRAM Upload using Scope Masks

content to VRAM in succeeding steps.

VRAM Upload Policies

Refer to Startup and Asset Management for how to load scenes from assets and upload to VRAM with the default upload policy Candera::ContentLoader::DefaultUploadPolicy.

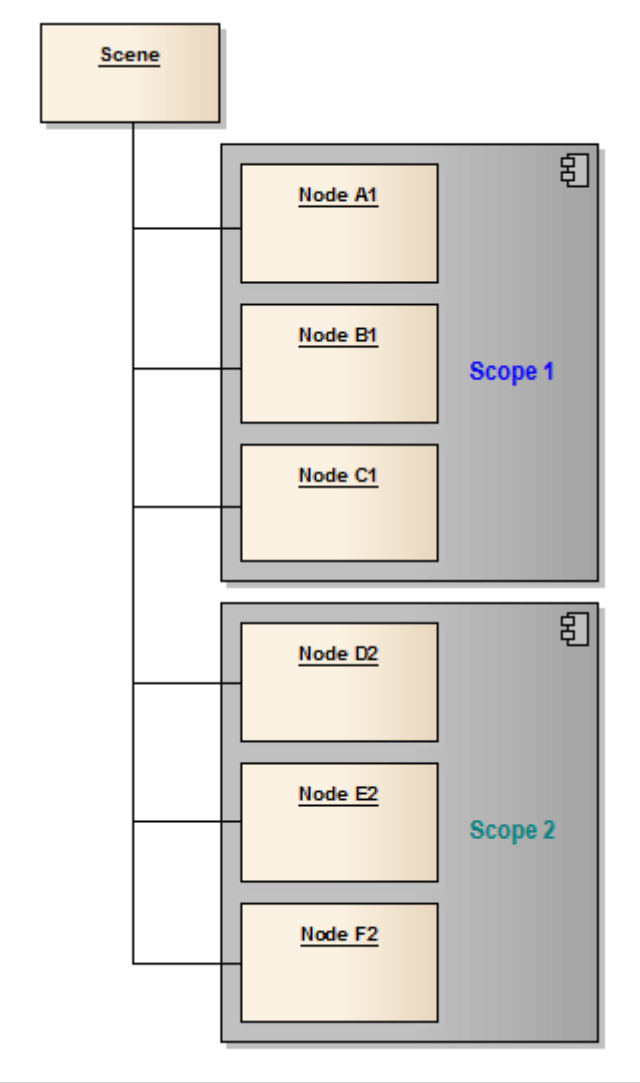
To implement VRAM upload culling with scopes, use Candera::ContentLoader::NoVramUploadPolicy and upload dedicated content based on scope masks later.

Scoping Concept

Refer to Scoping how to use and configure scope masks in SceneComposer.

Code Example

The following scene graph is a simple example where two parts of the scene have different scopes. Think about a use case where one part of the scene has to be displayed immediately after loading the scene and the second part has to be displayed only later in some special situations. To save time for uploading the content to VRAM not all of the content has to be uploaded immediately.



Using NoVramUploadPolicy

Loading the scene from the asset has to be triggered without loading the content to VRAM.

```
using namespace Candera;

ContentLoader& contentLoader = ContentLoader::GetInstance();
ContentLoader::BuildState buildState = contentLoader.BuildSceneContext("Scene",
ContentLoader::NoVramUploadPolicy);
if (buildState == ContentLoader::Completed) {
    SceneContext* sceneContext = contentLoader.GetSceneContext("Scene");
    Scene* scene = sceneContext->GetScene();
}
```

Upload Nodes with Scope 1

```
using namespace Candera;

ScopeMask scopeMask1(false);            // all scopes are disabled...
scopeMask1.SetScopeEnabled(1, true);    // ...and only scope 1 gets enabled.
scene->Upload(scopeMask1);
```

The scene can be rendered now as long as those nodes with scope 2 are not needed (e.g. rendering is disabled).

Deferred Upload of Nodes with Scope 2

```
using namespace Candera;

ScopeMask scopeMask2(false);
scopeMask2.SetScopeEnabled(2, true);
scene->Upload(scopeMask2);
```

Now the whole scene is uploaded to VRAM and can be rendered completely.

> In that simple use case the scopes are already set to the nodes e.g. during design phase in SceneComposer. The scopes can also be set during runtime by calling Candera::Node::SetScopeMask() or Candera::Node::SetScopeEnabled().

> This example is written for Candera 3D, for 2D please refer to Candera::Scene2D::Upload(), Candera::Node2D::SetScopeMask() and Candera::Node2D::SetScopeEnabled().

Revision #8
Created 2023-02-17 08:09:15 UTC by Tsuyoshi.Kato
Updated 2025-01-24 08:49:40 UTC by Elisabeth Zauner