VDK Unity

This project demonstrates the use of Euclideon Vault developer Kit (VDK) with the Unity game engine. There are two projects included:

- The basic package, including a basic flight camera and collider, this demonstrates the basic concepts required to get simulations and games using VDK running in unity. This is ideal for developers wishing to use VDK with their own projects
- The advanced project: includes a driving simulator and third person ragdoll. This demonstrates use of VDK with Unity physics and is good for exploring and testing interactions of unity systems with Unlimited Detail point clouds.

Both examples require a valid license for Euclideon Vault SDK, trial licenses can be obtained here

Installation

Both examples are tested with Unity 2019.3.0f6

1. Download and extract VDK 0.5.0 package from here using your license credentials (if you do not have one, free trials are available from here)
2. Download the Unity VDK example from here
3. Copy the files from Euclideon_vdk0.5.0/lib/(*your operating system here*)/to Assets/VDK in your Unity project
4. Replace the username and password lines from *Assets/VDK/vdkLogin.cs* to your vault login details **YOUR UNITY EDITOR MAY CRASH ON LOAD IF THIS STEP IS NOT COMPLETED**
5. Open Unity Hub, Select Add and specify the folder
6.

Basic Example

This is an example demonstrating how to use VDK with unity, it includes a minimalist example of a flight camera and an attached collider. Unlimited detail rendering is implemented as a postprocessing effect that can be applied to cameras in order to display UDS objects.

Sample Scene Structure

Main Camera

The main view used in the example. It has a flight camera script attached to it to enable user interation. Of importance to this object is the implemented post process layer and volume properties which must be included for the camera to view UDS files.

UDS Model

Each of UDS to be loaded in unity is represented as a one of these models

vdkLogin

This file contains the login logic for the unity example, including login credentials. GlobalVDKContext contains a vdkContext for managing licensing and login information between objects, and a vdkRenderContext, enabling the rendering of

VDKPPER

VDKPPER.cs contains the implemention of VDK in Unity as a post processing effect. This is applied to a camera as a texture

VDK Collider

This object demonstrates how to achieve physical collisions between Euclideon UDS models and native Unity colliders. Because of the potential scale of UDS objects it is not practical to construct mesh colliders of UDS objects (espeially if these objects are being streamed externally) The approach taken is to construct a mesh of the UDS object local to a point of interest (for example a player or the potential colliding object using information available from an instance of vdkRenderView.

Because the information contained in UDS files (especially unfiltered point clouds) can be noisy, we have included functionality to smooth the generated surfaces.

VDKCollider.cs contains the majority of the logic associated with the example collider system.