Date: February 26, 2020

Subject: RPK Testing

Technical notes: 4.24 plugin doesn't install, PRT runtime does not seem to be loading.

Initial testing of RPKs in both City Engine and Unreal Engine created an understanding of the usability of the rule packages in either program. City Engine parses the RPK natively and shows relevant parameters, hiding the individual texture and color settings per element. The PRT plugin for Unreal Engine on the other hand displays all the parameters for the RPKs. In the PRT plugin, it is essential to usability that the parameters are filtered, with the understanding that there are hidden options that allow the procedural generator to work. There currently is built in options to send certain parameters to the VR page, but for proper usability of the plugin, and for creating presentation ready applications with RPKs, there needs to be a measure of filtering of parameters.

RPKs have a number of inputs or starting geometries, that affect how context is generated. The RPKs listed below use the lot boundary as an input:
Building_FromFootprint_Textured_ByLandUse.rpk
Building_FromFootprint_Textured_ByBuildingType(1a).rpk
rule.rpk
rule v2.rpk
rule v3.rpk

The RPKs listed below use a mass as an input:
Building_FromMass_Textured_ByBuildingType.rpk
ColorMassByOrientation.rpk
BuildingShell_with_Detail(1a).rpk

In City Engine, generated lots cannot be extruded into a mass and thus a desirable workflow such as, lot boundary => lot boundary RPK => mass => mass RPK => finished context, is not supported. Instead, to use the RPKs with mass input, one must create the volume using City Engine's limited modeling tools before applying the rule package. The prototype RPKs that begin to generate mass from the lot boundary, taking into account setbacks and other zoning requirements, are somewhat successful, yet they can't be transformed into geometry. Unreal Engine handles these RPKs similarly. Rule packages build for lots utilize a flat .obj mesh to create the geometry. The problem comes with the mass RPKs. Importing solid .obj meshes allows for these RPKs to become functional, but there are problems with how textures and colors are applied. In many cases, the building uses the default black Unreal material, making things unreadable. Importing .obj meshes representative of a massing of a building becomes a good starting point for context building for architects. Using the BuildingShell_with_Detail.rpk, massings are enhanced by the detail of 3D windows, balconies, and doors. As stated above, this workflow can be enhanced by an exchange between lot and mass RPKs.

Notes on individual RPKs:

Building_FromFootprint_Textured_ByLandUse.rpk
-Land Use, Roof Type, Floor Height, Roof Height, ?Levels?

Building_FromFootprint_Textured_ByBuildingType.rpk
-BuildingType, Roof Type, Floor Height, Roof Height, ?Levels?

Rule.rpk
Rule v2.rpk
Rule v3.rpk
-When used on lots based on terrain, used normal vector rather than standard z direction.

Attachments. N/A

Copy to:
Henry Richardson