glTF-Tutorials

Previous: Animations | Table of Contents | Next: Meshes

Simple Meshes

A `mesh` represents a geometric object that appears in a scene. An example of a `mesh` has already been shown in the minimal glTF file. This example had a single `mesh` attached to a single `node`, and the mesh consisted of a single `mesh.primitive` that contained only a single attribute—namely, the attribute for the vertex positions. But usually, the mesh primitives will contain more attributes. These attributes may, for example, be the vertex normals or texture coordinates.

The following is a glTF asset that contains a simple mesh with multiple attributes, which will serve as the basis for explaining the related concepts:

```
{
  "scene": 0,
  "scenes" : [
    {
      "nodes" : [ 0, 1]
    }
  ],
  "nodes" : [
    {
      "mesh" : 0
    },
    {
      "mesh" : 0,
      "translation" : [ 1.0, 0.0, 0.0 ]
    }
  ],

  "meshes" : [
    {
      "primitives" : [ {
        "attributes" : {
          "POSITION" : 1,
          "NORMAL" : 2
        },
        "indices" : 0
      } ]
    }
  ],

  "buffers" : [
```

```
    {
      "uri" : "data:application/octet-stream;base64,AAABAAIAAAAAAAAAAAAAAAAAAAAAIA/AAAAA
      "byteLength" : 80
    }
  ],
  "bufferViews" : [
    {
      "buffer" : 0,
      "byteOffset" : 0,
      "byteLength" : 6,
      "target" : 34963
    },
    {
      "buffer" : 0,
      "byteOffset" : 8,
      "byteLength" : 72,
      "target" : 34962
    }
  ],
  "accessors" : [
    {
      "bufferView" : 0,
      "byteOffset" : 0,
      "componentType" : 5123,
      "count" : 3,
      "type" : "SCALAR",
      "max" : [ 2 ],
      "min" : [ 0 ]
    },
    {
      "bufferView" : 1,
      "byteOffset" : 0,
      "componentType" : 5126,
      "count" : 3,
      "type" : "VEC3",
      "max" : [ 1.0, 1.0, 0.0 ],
      "min" : [ 0.0, 0.0, 0.0 ]
    },
    {
      "bufferView" : 1,
      "byteOffset" : 36,
      "componentType" : 5126,
      "count" : 3,
      "type" : "VEC3",
      "max" : [ 0.0, 0.0, 1.0 ],
      "min" : [ 0.0, 0.0, 1.0 ]
    }
  ],

  "asset" : {
    "version" : "2.0"
```

```
    }
  }
```

Image 8a shows the rendered glTF asset.

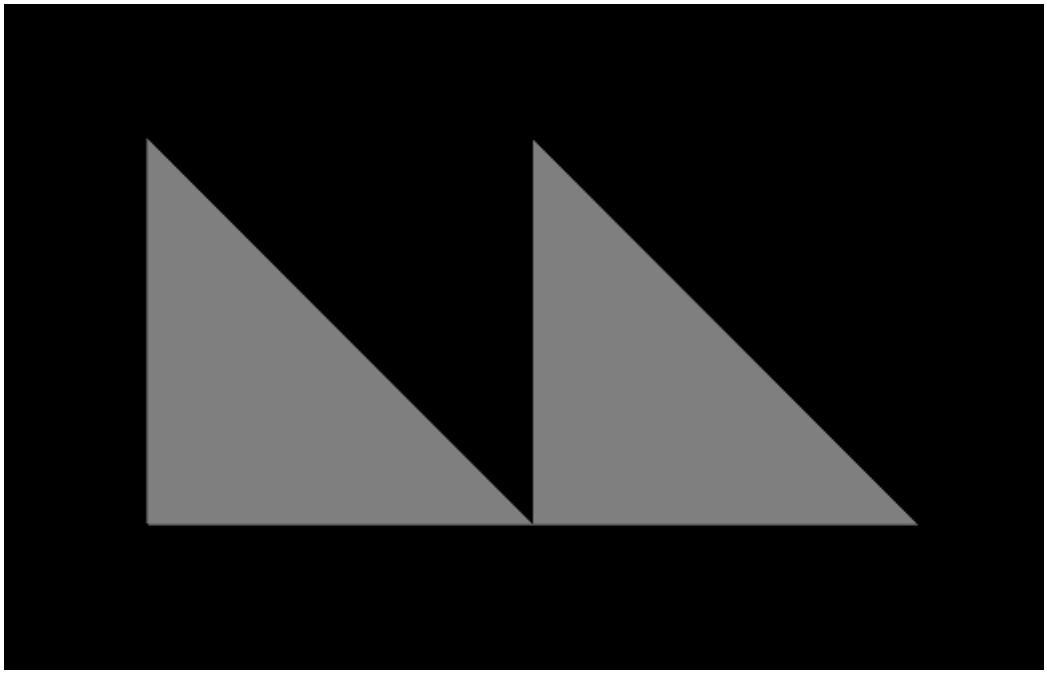

Image 8a: A simple mesh, attached to two nodes.

The mesh definition

The given example still contains a single mesh that has a single mesh primitive. But this mesh primitive contains multiple attributes:

```
"meshes" : [
  {
    "primitives" : [ {
      "attributes" : {
        "POSITION" : 1,
        "NORMAL" : 2
      },
      "indices" : 0
    } ]
  }
],
```

In addition to the `"POSITION"` attribute, it has a `"NORMAL"` attribute. This refers to the `accessor` object that provides the vertex normals, as described in the Buffers, BufferViews, and Accessors section.

The rendered mesh instances

As can be seen in Image 8a, the mesh is rendered *twice*. This is accomplished by attaching the mesh to two different nodes:

```
"nodes" : [
  {
    "mesh" : 0
  },
  {
    "mesh" : 0,
    "translation" : [ 1.0, 0.0, 0.0 ]
  }
],
```

The `mesh` property of each node refers to the mesh that is attached to the node, using the index of the mesh. One of the nodes has a `translation` that causes the attached mesh to be rendered at a different position.

The next section will explain meshes and mesh primitives in more detail.

Previous: Animations	Table of Contents	Next: Meshes
