```
ObjectBegin "pCubeShape1"  # {
 SubdivisionMesh "catmull-clark" [ 4 4 4 4 4 4 ]
 [ 2 3 1 0 4 5 3 2 6 7 5 4 0 1 7 6 3 5
    7 1 4 2 0 6 ]
   [ "interpolateboundary" ] [ 0 0 ] [ ] [ ]
   "vertex point P" [ -0.5 -0.5 0.5 0.5 -0.5 0.5 -0.5 0.5 0.5 0.5 0.5 0.5
     -0.5 0.5 -0.5 0.5 0.5 -0.5 -0.5 -0.5 -0.5 0.5 -0.5 -0.5 ]
   "facevertex float[2] st" [ 0 0 1 0 1 1 0 1 0 -1 1 -1
     1 0 0 0 0 -2 1 -2 1 -1 0 -1
     0 -3 1 -3 1 -2 0 -2 1 0 2 0
     2 1 1 1 -1 0 0 0 0 1 -1 1 ]
ObjectEnd  # }

ObjectBegin "pCubeShape2"  # {
 PointsGeneralPolygons
   [ 1 1 1 1 1 1 ]
   [ 4 4 4 4 4 4 ]
   [ 2 3 1 0 4 5 3 2 6 7 5 4 0 1 7 6 3 5
     7 1 4 2 0 6 ]
   "vertex point P" [ -0.5 -0.5 0.5 0.5 -0.5 0.5 -0.5 0.5 0.5 0.5 0.5 0.5
     -0.5 0.5 -0.5 0.5 0.5 -0.5 -0.5 -0.5 -0.5 0.5 -0.5 -0.5 ]
   "facevarying normal N" [ 0 0 1 0 0 1 0 0 1 0 0 1
     0 1 0 0 1 0 0 1 0 0 1 0
     0 0 -1 0 0 -1 0 0 -1 0 0 -1
     0 -1 0 0 -1 0 0 -1 0 0 -1 0
     1 0 0 1 0 0 1 0 0 1 0 0
     -1 0 0 -1 0 0 -1 0 0 -1 0 0 ]
   "facevarying float[2] st" [ 0 0 1 0 1 1 0 1 0 -1 1 -1
     1 0 0 0 0 -2 1 -2 1 -1 0 -1
     0 -3 1 -3 1 -2 0 -2 1 0 2 0
     2 1 1 1 -1 0 0 0 0 1 -1 1 ]
ObjectEnd  # }
```

Above are the two different representations of geometric types. The first one is polygon and second one is subDivision mesh. This is the format in which we have to feed the data to the renderer. There can be other geometric types also. For example particle, patch, curves etc. A single type of geometry should be having fixed set of arrays as well as couple of other arrays. Most of the cases, we do have arrays of integer and double type.

The geometries that are mentioned above is a simple cube of 8 vertices, where in real production we have much bigger objects contains thousand and in some cases million of vertices or points.

The idea is to store this data in HD5 format from our 3d application. For that we need to have an exporter and once the file has been created an another application that will read this data back and feed to renderer at render time.

Feeding back this data to the renderer is simple. For example, we'll retrieve the arrays stored for particular objects back in integer or floating point arrays and pass these arrays to renderer's library function. The library function to pass data for polygon type of objects is :

```
RiPointsGeneralPolygonsV(   RtInt npolys,
                            RtInt nloops[],
                            RtInt nvertices[],
                            RtInt vertices[],
                            RtInt n,
                            RtToken tokens[],
                            RtPointer parms[]);
```

To pass all the arrays required for this function, we'll read from HD5 file into separate arrays and feed to this function.

I gave a quick look on HD5 site and it's looking quite promising. I personally feel that introduction of HD5 in our production pipeline can play a very important role. Before we actually start the development we would like to know you expert advise for the concern. I would really appreciate if you can show us the right way to go ahead.

Best Regards

Prashant saxena
Founder
Pantheon Studios

www.pantheon-studios.in