Sop处理对象object

detail最高层的数据，对应整个object

Primitive集合对象中的组成部分，如三角面片，nurbs曲面，或者volume

point几何对象上的点

Vertex一个point可能同时存在对个相邻的primitive，因此共享同一个point的多个primitive会各自存一个对point的引用

Geometry Attribute属性

New pane tab type --- geometry spreadsheet

Attribute wrangle

删除 attribute delete

读取/修改 同层级属性 @变量名

以下3种方式获取数据：第二个输入数据的垫层及位置属性@P

point(1,”p”,@ptnum)

Point(@OpInput2,”p”,@ptnum)

v@opinput1\_P

注意事项：

一

[V表示类型简写@opinputN(N表示输入序号0,,1,2...)\_属性名称](mailto:V表示类型简写@opinputN(N表示输入序号0,,1,2...)_属性名称)：这种方法只能获取其他输入中对应序号对象的属性

Point(N,”属性名称”,点序号)：可以读取任意输入中任意序号对象的属性

二

如果使用的属性是非浮点数类型，如vector型向量，第二种方法必须@之前写上属性数据类型的简写。

|  |  |
| --- | --- |
| vex type | Syntax |
| Float | f@name |
| Vector2(2 floats) | u@name |
| Vector(3 floats) | v@name |
| Vector4(4 floats) | p@name |
| Int | i@name |
| Matrix2(2x2 floats) | 2@name |
| Matrix3(3x3 floats) | 3@name |
| Matrix4(4x4 floats) | 4@name |
| String | s@name |

不同层级之间的属性读取可以使用point(...),prim(...),detail(...)函数完成

Detail(geometry,属性名称)

备注：geometry表示输入的几何对象，分两种情况

1该数据来自输入的节点，那么直接写输入节点序号(0,1,2,3...)

2来自其他节点，那么则填入它的路径

属性名称表示P的属性，此处填写”P”(加双引号即可)

Point(geometry,属性名称,点的序号)

Prim(geometry,属性名称,对象序号)

增加属性：数据类型@变量名

例如，代码编辑器中输入如下

Float @myval=0;

@myval+=1;

Float @mass=1;

Vector @up={0,1,0};

属性在整个节点网络中，一旦创建，在认为手动删除之前，一直存在且全局可见。

属性非常消耗空间和性能，不需要的时候，使用attribute delete删除

变量variable

使用包含变量的一组值来定义局部变量，使用set(...)

例如：

X=set(0,@y,0);

参数parameter

·在窗口上，对参数进行自动链接；

右键点击源参数，选择copy parameter

右击要粘贴的参数，选择paste relative reference

·在窗口上，对参数进行手动链接；

使用hscirpt表达式，

在目标参数处输入函数ch(“源参数路径”)，源参数路径三种情况：

在同节点内的参数: ch(“size”)

在同一节点网络内的参数，用简写”../” : ch(“../BoxSize/BoxSizex”)

不同节点网络内的参数，使用完整的路径，第一个斜杆”/”不要忘记：ch(“/obj/BoxSize/BoxSizey”)

·在窗口上，手动创建参数；

Parameter description一栏，新建的变量有2个名称：Name和Label，name是参数整整的名字，使用ch(“参数名”)或者vex内引用该参数是，用的都是name，注意name不能包含空格

但label可以包含空格，作为参数在用户界面上显示的名称

·在wrangle节点里，用vex代码创建窗口上的参数；

使用chi，chf，chv，chs函数在vex中定义，随后在界面上点击一个按钮即可，或者还可以用chramp来定义ramp

·在wrangle节点里面，用vex代码读取窗口上的参数；

使用chi，chf，chv，chs即可

·在窗口上，读取vex代码创建的属性作为参数使用

制作城市道路：

1. 建立道路曲线
2. 对曲线进行交点合并，细分，切割等操作
3. 算出曲线的四个端点和中心交点，并进行排序
4. 基于每个点生成临近点，并完成连接操作
5. 沿着特定方向，向外一定点，完成4的操作，最后连接成面，形成路面

一建立道侣曲线以及对曲线进行合并，细分，切割操作；

·建议道路曲线；（节点curve）

·只保留曲线点的位置属性，一面导入引擎出现属性不一的问题（节点attributedelete）

·合并曲线的交点；（节点fuse）

·细分曲线；（节点resample）

·创建路面属性，路面的宽度；（节点attribcreate）

·翻转曲线为下一步从两边分割曲线，从中点重新分布点的序号，分成两个基元准备；（节点reverse）

·从中点分割曲线，分成两部分，路面交接与路面延伸的曲线（节点carve）

·合并分割后的点；（节点fuse）

·找到曲线的交点，并标记其属性值为1，其余为0；（节点pointwrangle）

·（节点pointwrangle分支）只保留曲线的交点（节点delete）

·（节点pointwrangle分支）转移属性，把上一步交点属性转移到基元属性；（节点attribpromote）

·保留与曲线交点相链接的线段（保留路面交界处曲线），通过上一簇的属性进行删除；（节点delete）

Pointwrangle节点代码

//neighbourcount()返回连接到指定点的点数

Int nc=neighbourcount(0,@ptnum);

If (n>2)

{

I@intersectionid=1;

}

1. 算出曲线的四个端点和中心点，并进行点序号的排序

·为曲线上的点分布坐标属性，富裕发现属性；（节点polyframe）

·分割曲线，可以调节路面交接处长度（节点carve）

·合并风格后的节点（节点fuse）

·算尺曲线的各个端点并为发现属性赋值为临近点发现只，并移除其他所有点；（节点pointwrangle1）

·计算曲线端点的atan2()的值，主要用于排序点的序号；（节点pointwrangle2）

·为曲线端点进行飘絮，按照上一步计算出的属性值；（节点sort）

·属性转换，转换宽度属性从基元到点级别；（节点attribpromote）

Pointwrangle1节点代码

//neighbourcount()返回连接到指定点的点数

Int nc=neighbourcount(0,@ptnum)

If (nc>1)

{

Removepoint(0,@ptnum);

}

//neighbour() 函数 返回连接到给定的下一个节点的序号

//赋予端点法线

Else{

v@N=point(0,”N”,neighbour(0,@ptnum,0));

}

Pointwrangle2节点代码