

课程四：导入街道

内容

部分一：导入 **DXF** 街道数据

部分二：导入 **OSM** 街道数据

部分三：导入 **Shapefile** 街道数据（**GDB**）

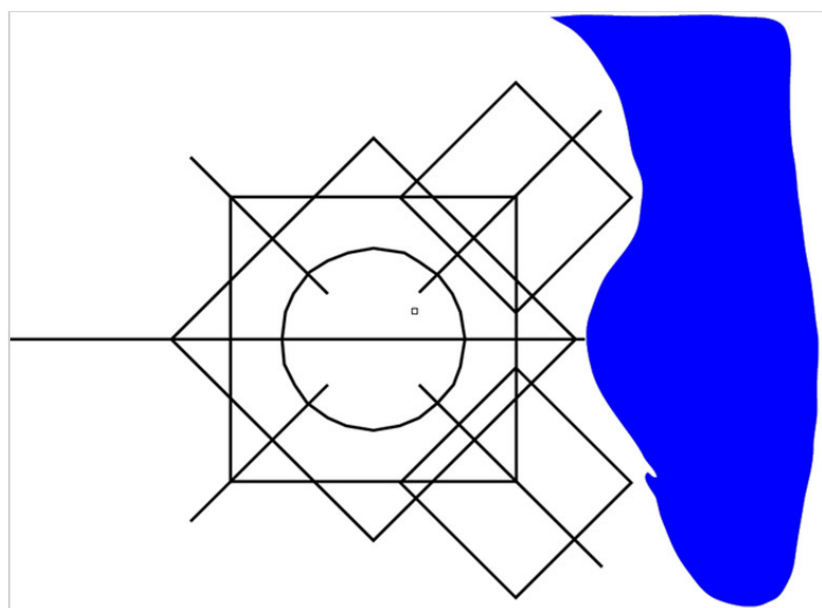
第一部分：导入 **DXF** 数据

CityEngine 街道网络

CityEngine 街道网络图，由图形节点（十字路口）和图形边缘（街段）组成。它们可以在 CityEngine 内部生成，或者通过外部文件导入，例如 DXF。

为 CityEngine 准备数据

在这个例子中，一座沿海城市的主要道路已经在 Illustrator 中被设计完了。



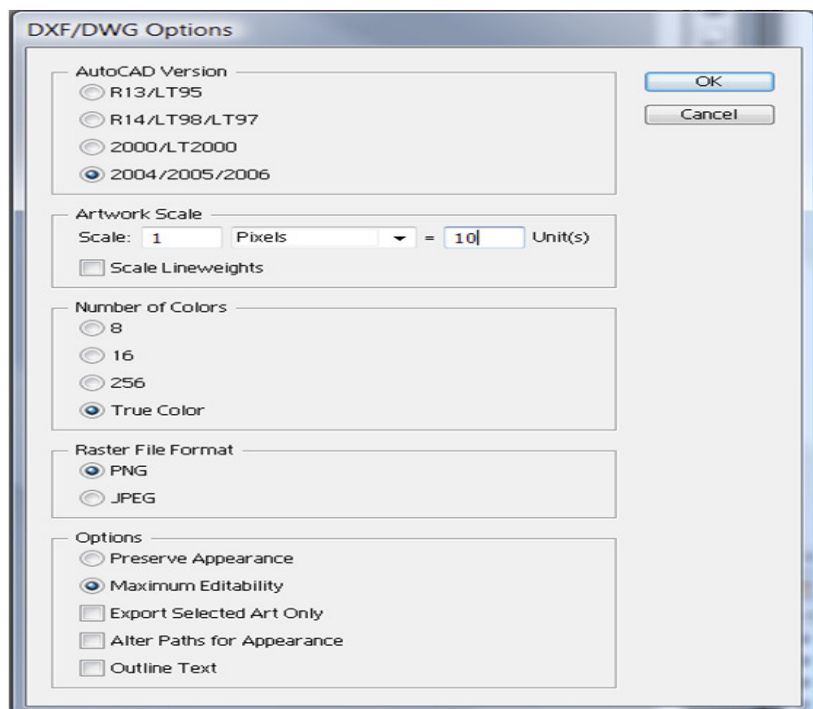
从 CAD 的外部应用程序输出.dxf 文件

当输出街道网络时，确保单位适应 CityEngine 的单位系统，通常以米作为单位。你也许希望在文本编辑器中打开.dxf 文件并且查找节点数据以查看街区的尺寸被写入到这个文件里。在本例中，我们设置一个像素等于十个基本单位为了获得我们需要的尺寸[看输出设置快照（export options screenshot）]

一些由 DXF 输出文件生成的节点数据：

```
...
AcDb2dVertex
10
1244.99951171875
20
234.998046875
30
0.0
0
VERTEX
...
```

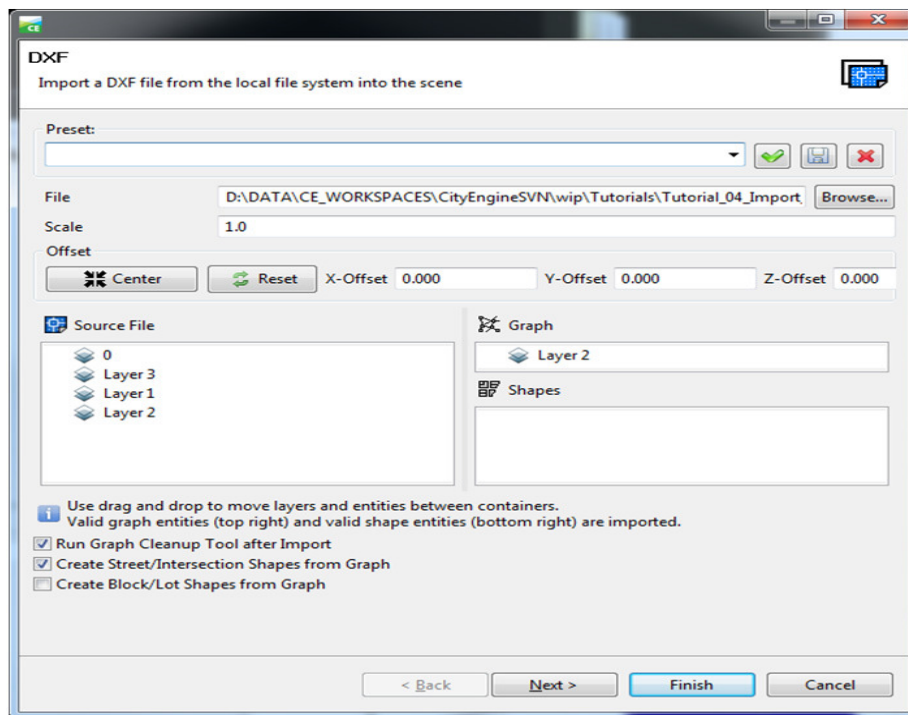
在 Illustrator 的 DXF 输出对话框的建议设置：



往 CityEngine 中导入.dxf 文件

- 打开场景文件 “sesame_01.cej”
- 在'data'文件夹中找到'sesame_streetsketch.dxf'
- 在文件上右击，选择'Import...'

数据层(Layer2)已经被导入进来添加为图形层



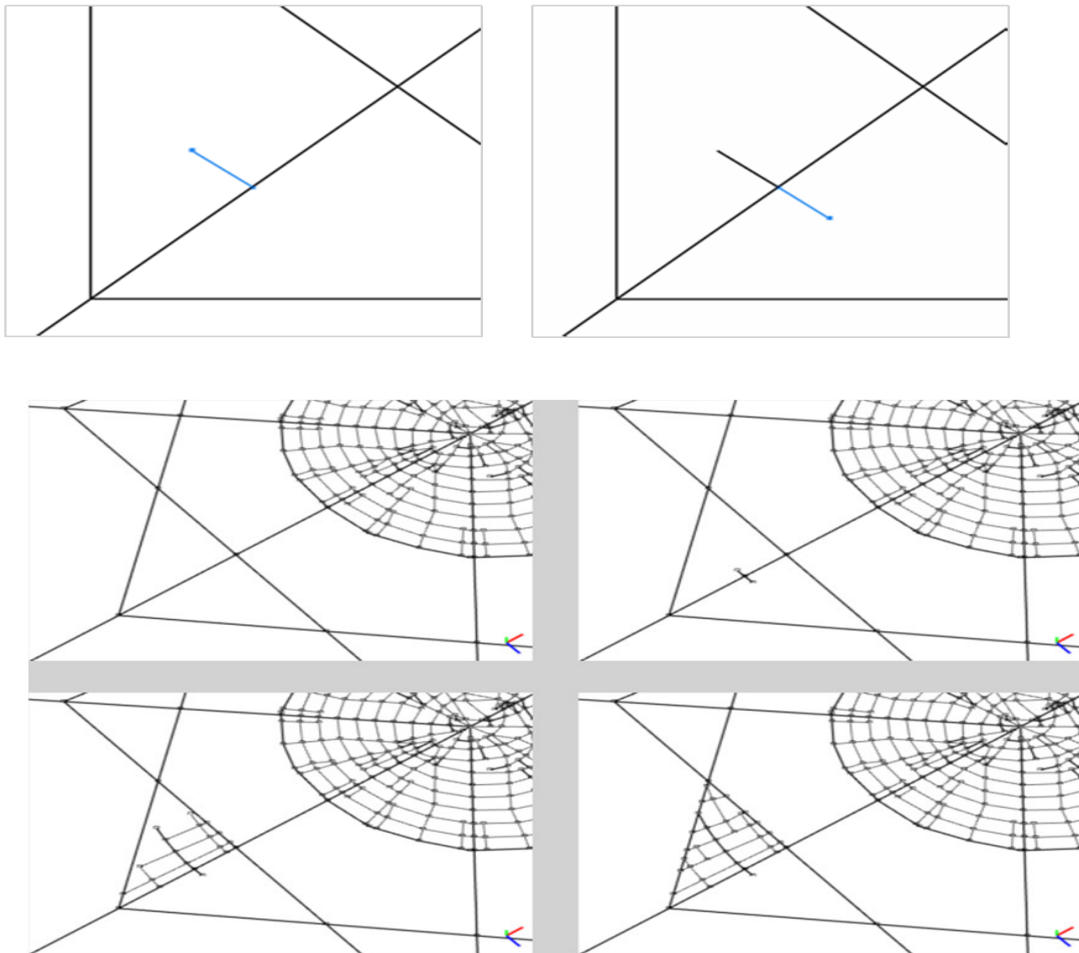
- 确保 “Run Graph Cleanup Tool after Import” 被勾选上
- 点击 “Next”
- 确保相交图段，将点捕捉到图段上以及合并节点。
- 设置捕捉距离为 1，合并距离为 5.
- 点击“Finish”

一个新的图形层叫做 “sesame_streetsketch” 出现在场景编辑器中。或者，你也可以直接打开 “sesame_02.cej”，获得的场景与之前你导入数据后的场景是一样的。

创建小街道

主要街道被导入后，你需要完善街区路网，这期间就需要创建小街道。CityEngine 街道生长算法具有填补现有封闭块的功能，因此你可以用迭代的方式在建筑群周围生成街道。

- 聚焦到你想创造小街道的建筑群区域
- 利用图形编辑工具创建两个简单的小街道（此操作一方面可以明确建筑群区域的街道方向，另一方面为利用算法生成街道定义了起始节点。）
- 手工的创造一些街道区段
- 选择区段（利用键盘“Q”键改变选择方式），然后通过“Grow Streets”对话框来填充空白区域。

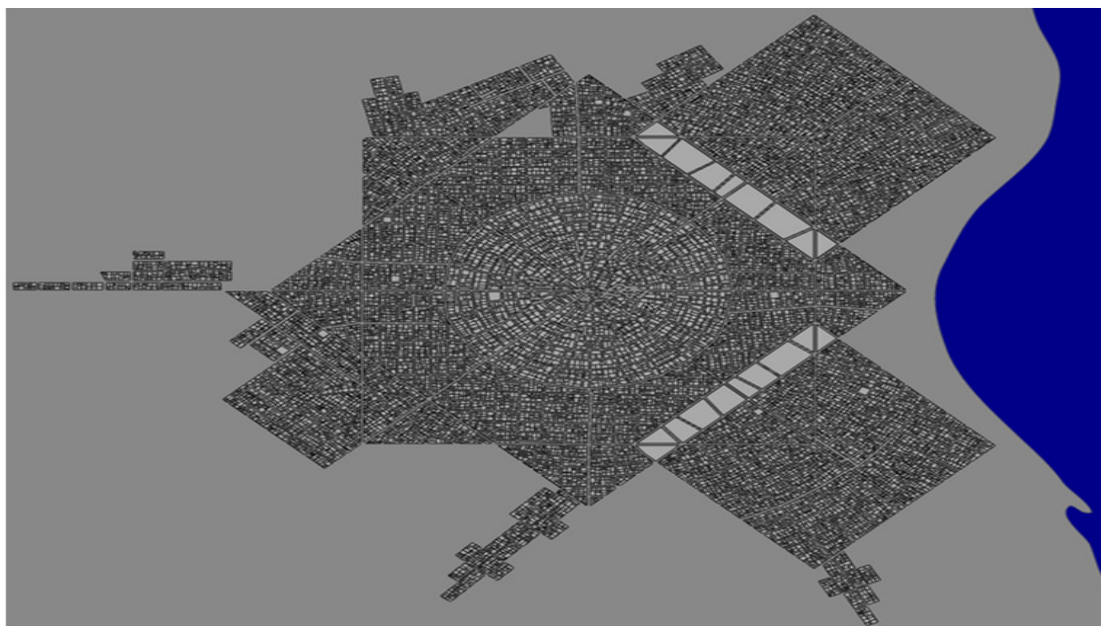


“sesame_03.cej” 包含了以上操作的结果。

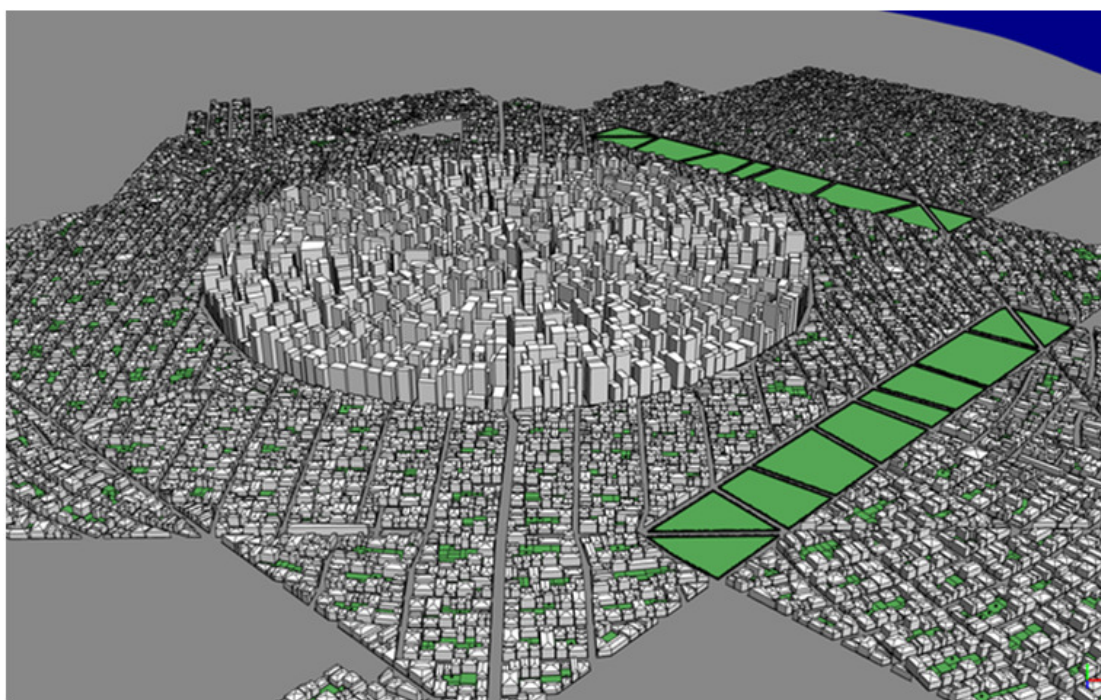
最终结果

场景文件 “sesame_12.cej” 包含了在影像中可以分辨出来的路网及地块的模型。

Sesame City 中包含的小街道及拔地而起建筑的分布形式：



Sesame City 的简单建筑及植被模型：



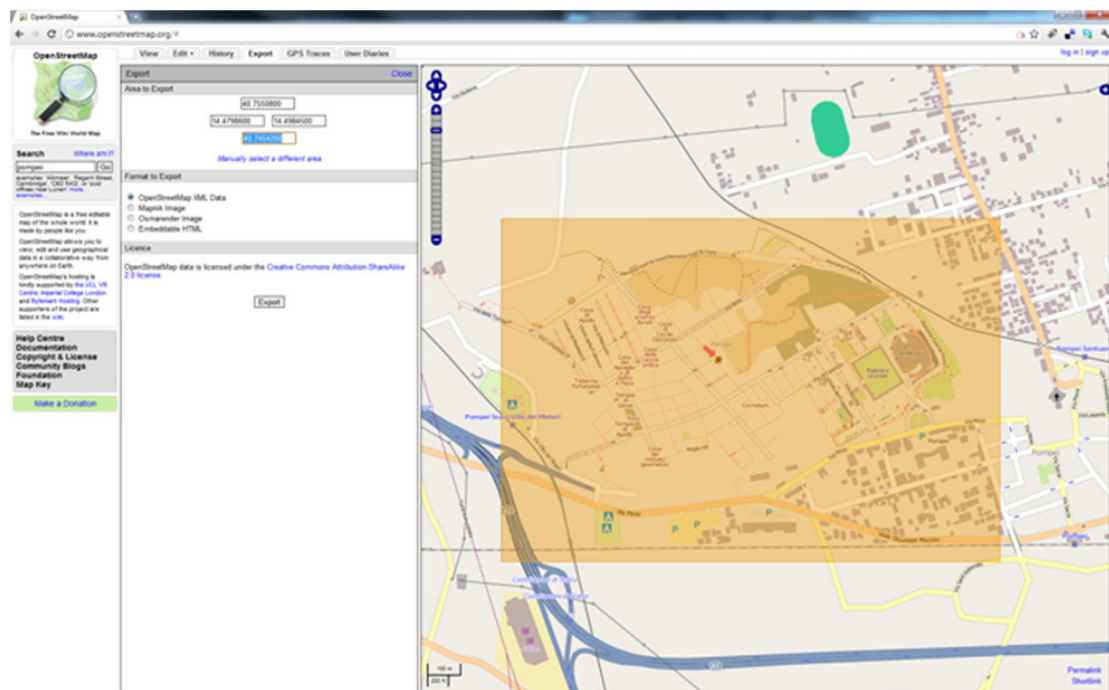
第二部分：导入 OSM 街道数据

“OpenStreetMap 是一个以创建及提供免费地理数据为目的的工程，比如人们非常想使用的街道底图数据”
—openstreetmap.org

OSM 是用来在地图中描述矢量数据以 XML 为基本格式的文件。它定义了三种基本数据类型，如节点、道路和闭合路，这些类型被用来描述其他的元素。

- 1) 节点 (Nodes): 两个图段之间的点。
- 2) 道路(Ways): 一系列有序的节点，在编辑器中显示为连接的线段。
- 3) 闭合路 (Closed Ways): 闭合路的样式是完全封闭的一个环。它们通常被用来表示公园、湖泊及岛那样的区域。

在 OpenStreetMap 中输出 OSM 文件



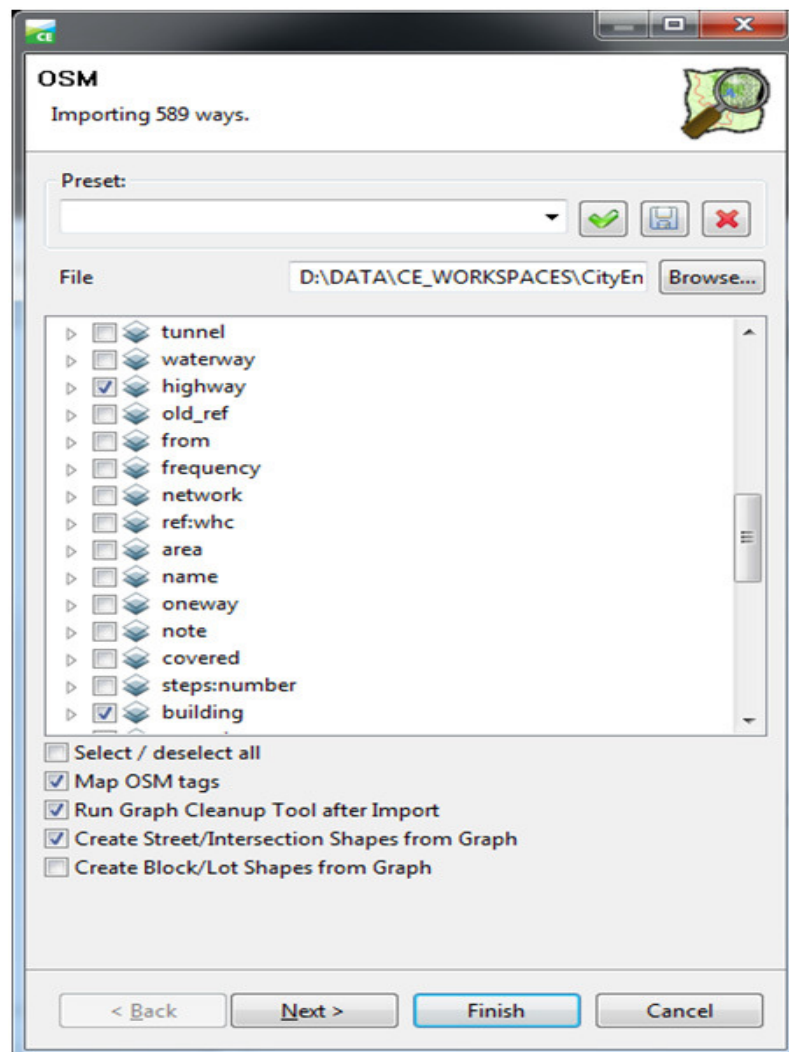
- 本课程所需的 OSM 数据（庞贝古城）从下列地址下载：

api.openstreetmap.org/api/0.6/map?bbox=14.4798600,40.7454200,14.4984500,40.7550800

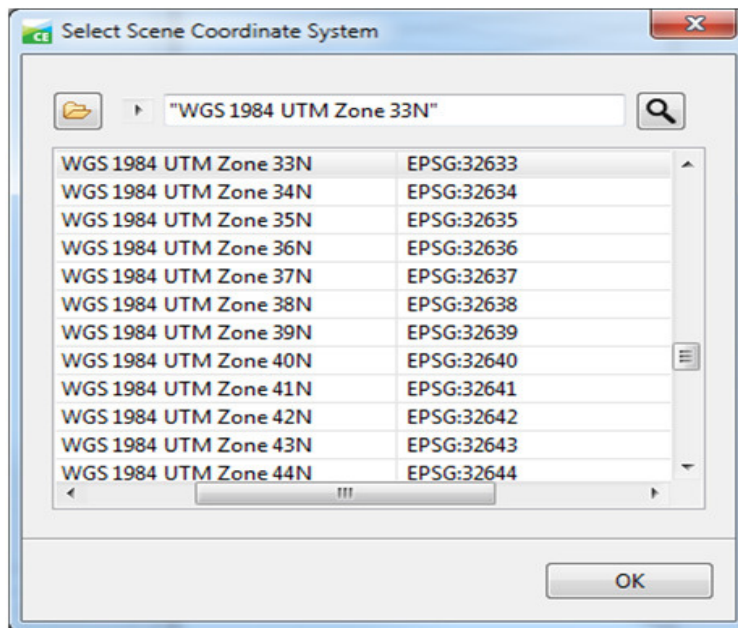
- 在本工程的“data”文件夹中查找 OSM 文件

导入一个 OSM 文件到 CityEngine 中

- 打开一个新场景 File>New...>CityEngine>CityEngine scene。
- 将文件拖拽到 3D Viewport 窗口中
- 在 OSM 文件导入对话框中，选择要导入的图层。通常情况，街道数据被导入到 highway 层，建筑物被导入到 building 层。使用“Select/deselect all”选项选择是否全选图层作为所需图层。
- 点击“Finish”



- 使用推荐的坐标系统 “WGS 1984 UTM Zone 33N”
- 点击 “OK”。



两个新图层（路网和模型）出现在场景编辑器中（Scene Editor），OSM数据在 3D Viewport 窗口中显示出来，街道模型会沿着导入的道路中心线自动的创建出来。

- 重命名图形图层为 “OSM_Streets”。
- 重命名模型图层为 “OSM_Shapes”。

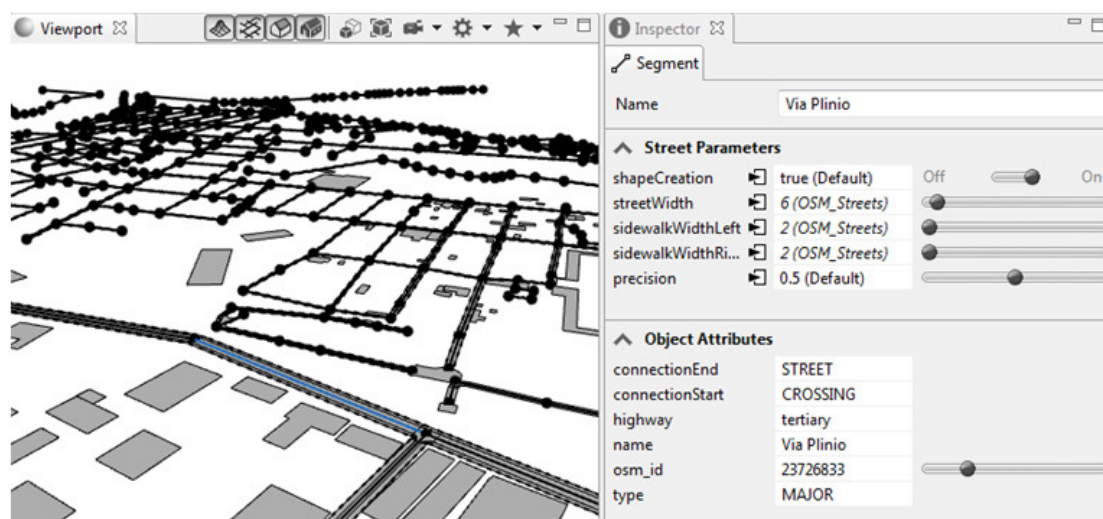


街道宽度

如果你仔细观察一下生成的街道模型，你会发现他们在宽度上会有所不同。当导入 OSM 标记信息选项可用的 OSM 地图数据时，CityEngine 创建出与 OSM 街道类型宽度一致的图层。

- 选择一段街道并在监视器(Inspector)窗口查看它的参数

在下面展示的例子中，拥有“tertiary”类型的街道已经将主干道与人行道分配了宽度参数，参数来自于导入的“OSM_Streets”图层。



你也可以手动的修改宽度参数来改宽度，或者使用工具条上的 Street Edit Tool 也能达到目的。

OSM 属性制图

- 在场景编辑器（Scene Editor）中选择新的图形网络图层并在监视器（Inspector）中查看图层的属性编码。

在影像之上的街道是“tertiary”类型的，通过 `streetWidthByClass` 函数定义了宽度为 6 米。你也可以根据自己的喜爱而改变制图的样式。

```
#2012.1
```

```
//-----
```

```
// Example OSM Tag Mapping
```

```

streetscale = 1 // street width scale factor
width = getObjectAttr("width")
lanes = getObjectAttr("lanes")
attr streetWidth = // street width dependeding on available attributes
case width > 0 : width * streetscale
case lanes > 0 : lanes * 3 * streetscale
else : streetWidthByClass * streetscale * oneway
class = getObjectAttr("highway")
12
streetWidthByClass =
case class == "primary" : 8
case class == "secondary" : 7
case class == "tertiary" : 6
case class == "motorway" : 12
case class == "trunk" : 11
case class == "road" : 6
case class == "residential" : 5
case class == "footway" : 2
case class == "cycleway" : 2
case class == "steps" : 2
else : 4
oneway = // oneway width correction
case getObjectAttr("oneway") == "yes" : 0.5
else : 1
sidewalkscale = 1 // sidewalk width scale factor
sidewalkWidth =
case class == "primary" : 2
case class == "secondary" : 2
case class == "tertiary" : 2
case class == "residential" : 2
else : 0
attr sidewalkWidthLeft = sidewalkWidth * sidewalkscale
attr sidewalkWidthRight = sidewalkWidth * sidewalkscale

```

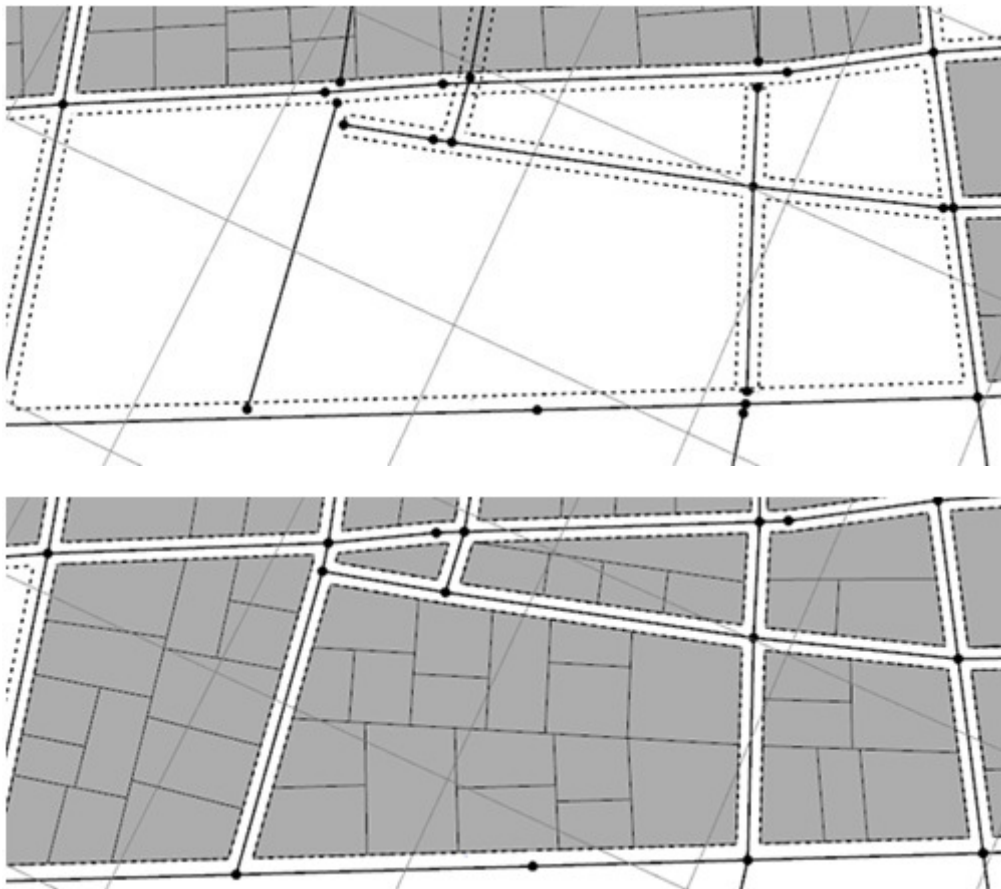
冲突数据

导入的 OSM 数据通常不是那么的完美，经常会有一些冲突（例如：街道压盖）。打开图块或红色的虚线表示图段未连接上或者互相压盖的街道通常会带来无效的建筑群区域。这里有不同的方式去解决下面的冲突：

- 使用 Cleanup Graph Tool

- 在导入对话框中为 **highway** 图层选择较少的街道
- 为街道与人行道设置较小的宽度值（减少街道间的重叠）。
- 在导入数据后通过合并或移除较近的十字路口，平行道路以实现手动规整街道网络的目的。

一个图形节点未连接，清理前后对比情况：

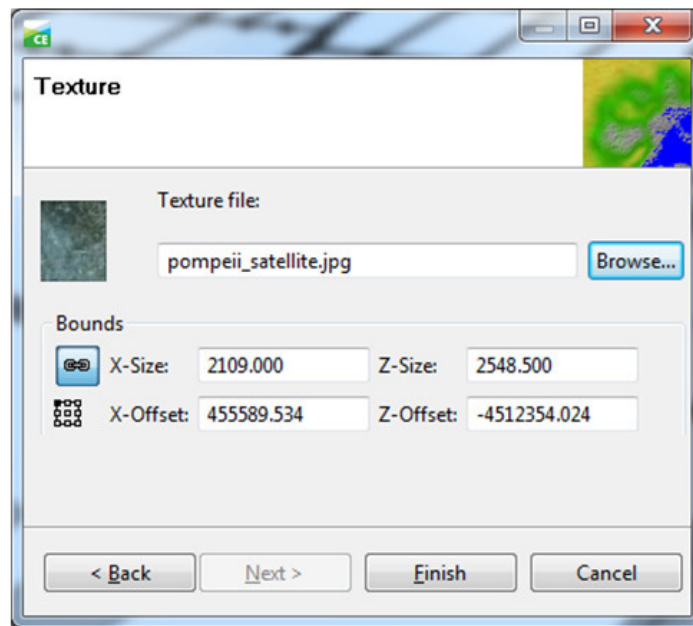


注意：OSM 数据包含的街道图段有可能会受到不同不同高度的影响。也许这个信息不适合 OSM 的数据信息，CityEngine 在导入所有数据时高程值都是一致的。在相交的图段处使用 Graph cleanup tool 也许会让相交处不正确。

添加卫星影像数据

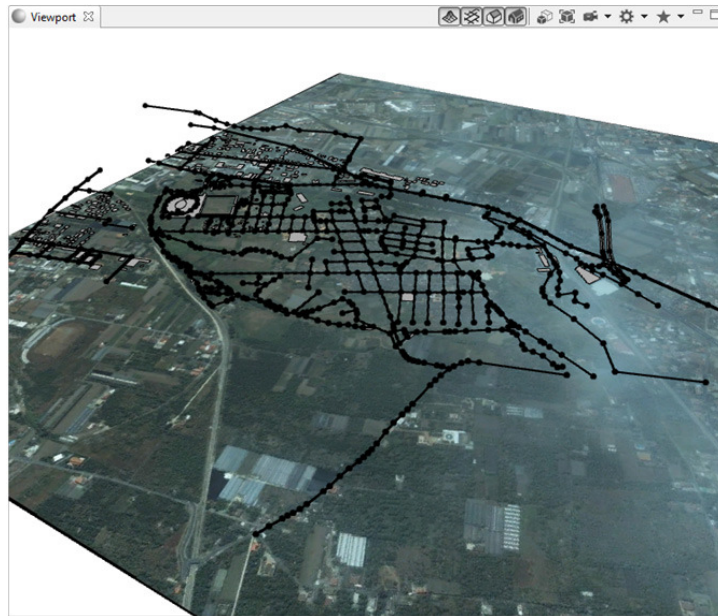
- File>Import...>CityEngine Layers>Texture Import

- 在 maps 文件夹下找到 “*pompeii_satellite.jpg*” 数据
- 为卫星影像定义 “WGS 1984 UTM Zone 33N” 的投影
- 尺寸[Size(bounds)]与偏移[Offset(Location)]被设置成自动的适应
“*Pompeii_satellite.jgw*”
- 点击 Finish 键完成创建新的纹理底图图层



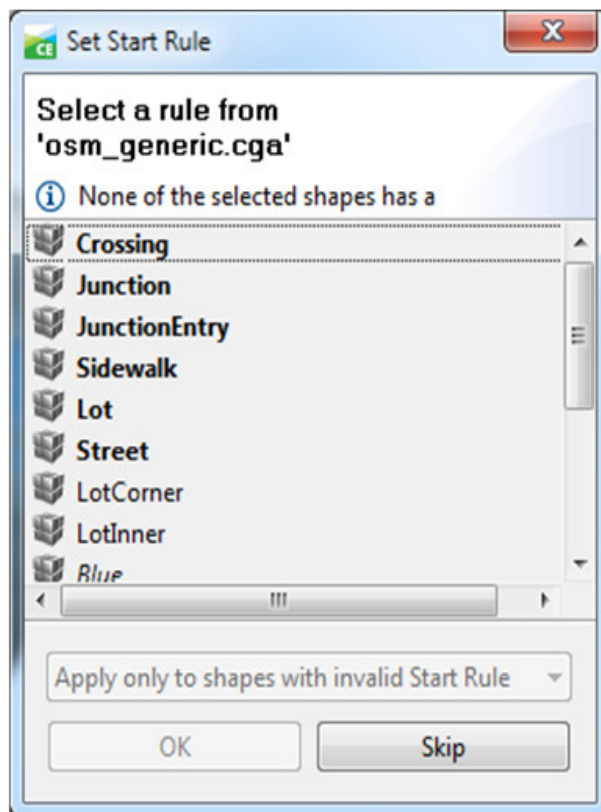
注意：CityEngine 不会为导入的影像重新投影。选择投影文件是为了让文件的位置正确。因此在 CityEngine 的场景中准备已经做过投影的数据是至关重要的[在这个例子之前,Pompeii 的卫星影像数据已经在 ArcMap 中被定义了 WGS 1984 UTM Zone 33N(本课程场景中使用的投影坐标)]。

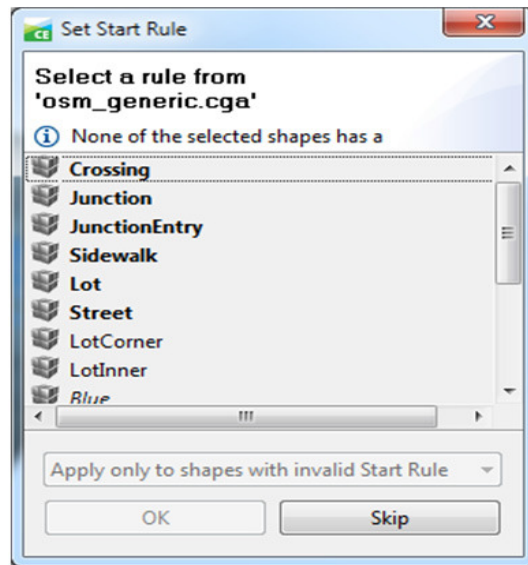
被加载进来的卫星影像数据：



生成模型

- 选择所有的二维图形
- 分配“osm_generic.cga”的规则文件。有可能会弹出起始规则文件为空的提示，按“Skip”选择忽略。





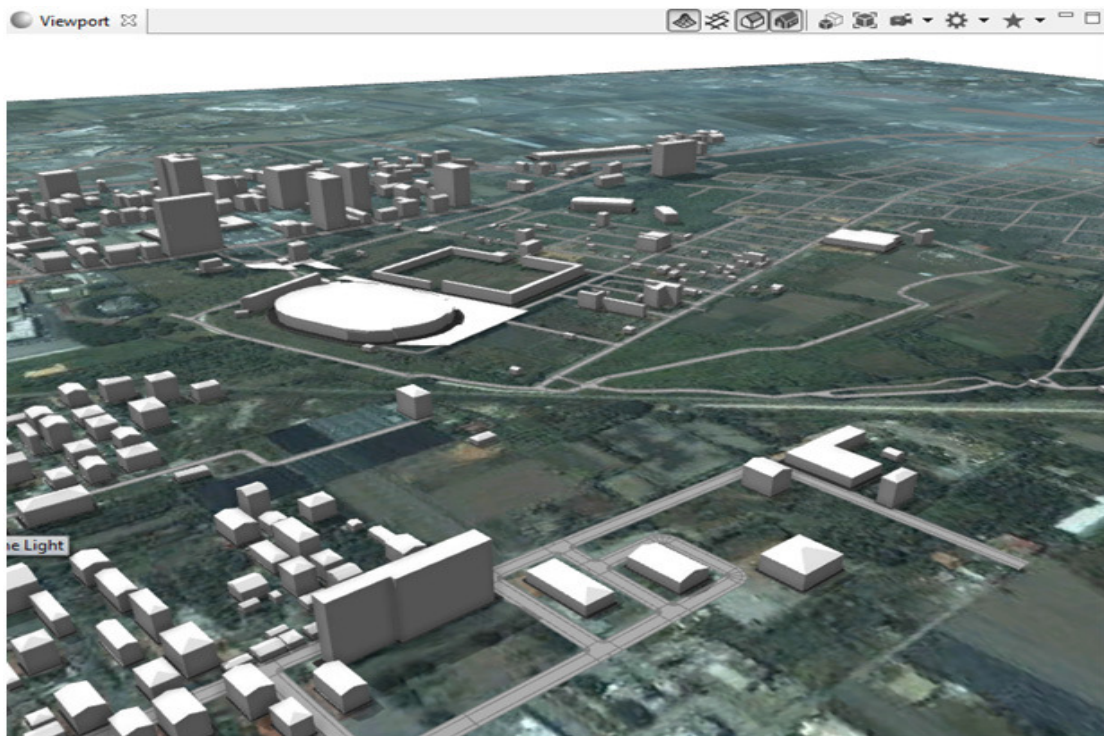
- 生成模型

也许你会注意到，仅有街道模型生成出来。建筑模型看起来好像没有被创建。选择模型中的一个并在 Inspector 窗口中观察它们，并发现它们没有被分配规则。

- 修复它，在 viewport 窗口中右击模型并进行 select→select Objects with Same Start Rule(选择相同规则文件的对象)操作。这样没有规则的模型都会被挑选出来。

- 在 Inspector 窗口中的 Start Rule（起始规则）点击“select...”,并将“Lot”作为 Start Rule（起始规则）

在 OSM 数据上生成简单的模型：



第三部分：导入 Shapefile 格式的街道数据（GDB）

导入 Shapefile 与 GDB 格式的街道数据几乎是一样的，因此这里就不单独讲解 GDB 格式的导入了。

Shapefile 格式的数据

导入 Shapefile 与 GDB 格式的街道数据几乎是一样的，因此这里就不单独讲解 GDB 格式的导入了。

- 定位到课程数据文件夹中的 streets.shp 文件

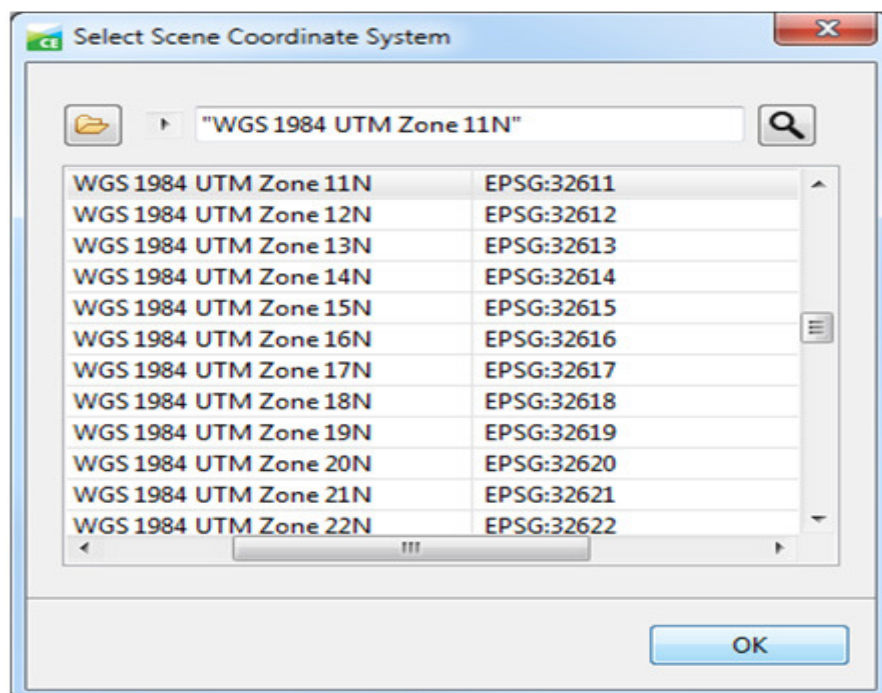
Shp 文件通常会有一个保证导入数据有一个正确配准信息的*.prj 文件。数据中包含了多段线，它会作为街段（street segments）被导入到 CityEngine 中。

ArcMap 中的 shp 文件属性表：

FID	Shape	highway	name	width
0	Polyline	footway		2
1	Polyline	residential	North Camde	7
2	Polyline	residential	Benedict Can	7
3	Polyline	footway		2
4	Polyline	secondary	Beverly Boul	11
5	Polyline	footway		2
6	Polyline	footway		2
7	Polyline	residential	Oxford Way	9
8	Polyline	residential	Carmelita Av	7
9	Polyline	residential	Cove Way	7
10	Polyline	residential	Rexford Driv	7

导入数据

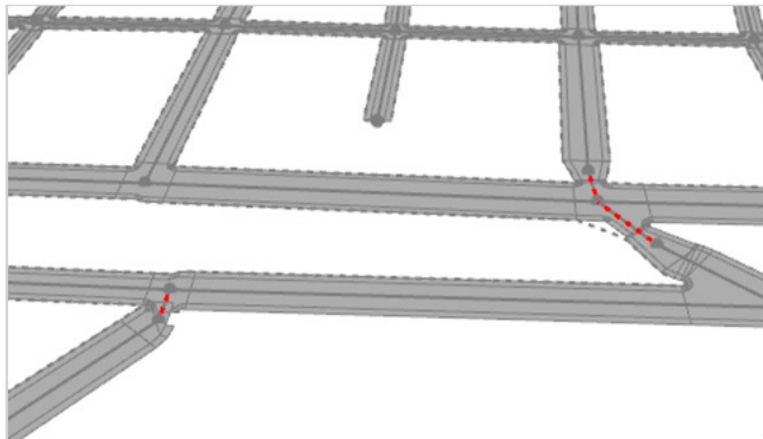
- 创建一个空的 CityEngine 场景。
- 将 shp 文件从 Navigator 窗口拖拽到 3D Viewport 窗口。这个过程将会使用默认的设置，在本次课程中是比较合适的。
- 在导入的过程中，CityEngine 会让你选择一个场景坐标系统。建议选择默认的，如果不正确可在下面的对话框中进行调整。



注意：在本例中的 shp 文件包含着 “width” 这个属性为了控制街道的宽度。没有这个属性的数据宽度会被设置为默认值。街道宽度也可以在数据导入后再进行调整。为了让模型有更多的属性显现出来，我们也可以使用其他的属性映射，参考 CityEngine 的说明书以了解更多的内容。

Cleanup

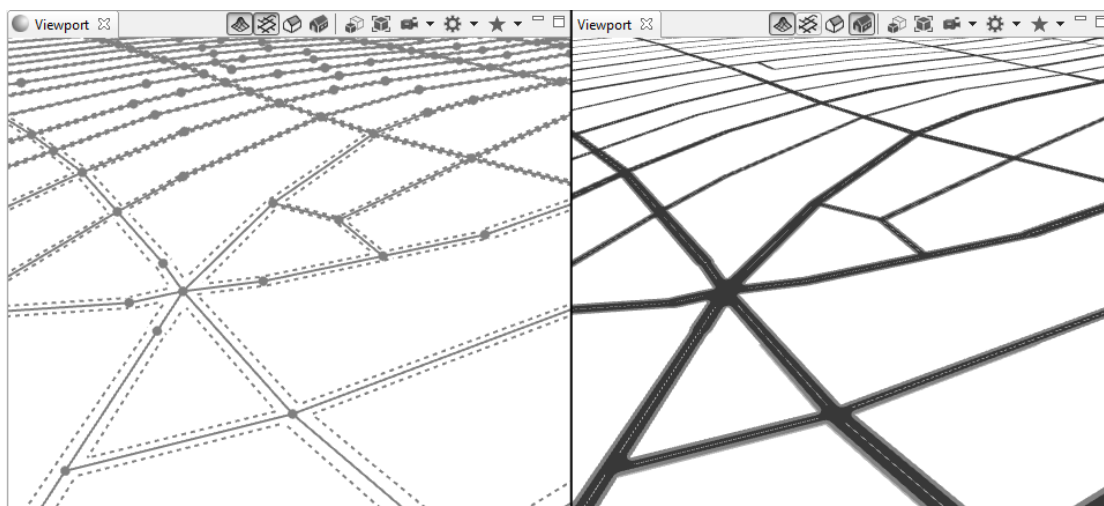
导入的数据会显示出一些冲突的模型，红色的虚线表示的即是。使用 Cleanup Graph 这个工具即可自动修复这些冲突。



分配规则

- 选择所有的街道模型
- 在 Navigator 窗口中，将 “sesame_01.cga” 文件拖拽到 3D Viewport 中已选择的文件上。
- 简单的街道模型被创建出来

潜在的街道网与生成的模型：



(Lejenson 作品，如有不足，欢迎指导)