

# 使用 CityEngine 创建酷炫场景

技术部 慕晓燕



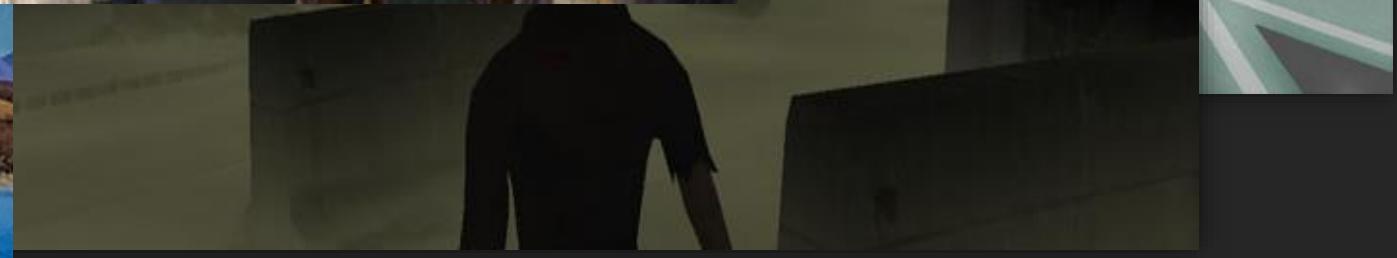
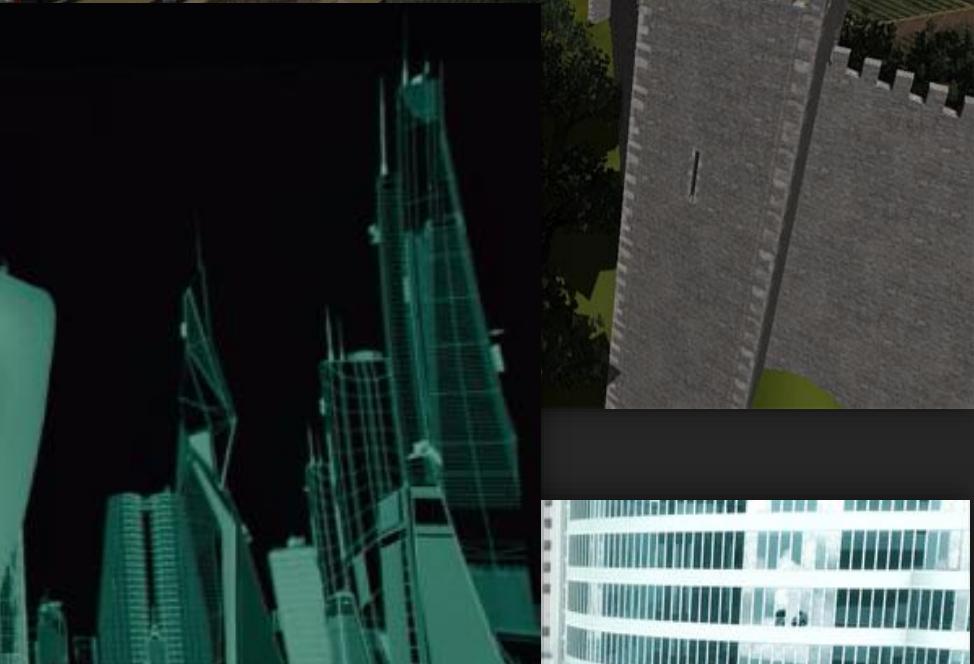
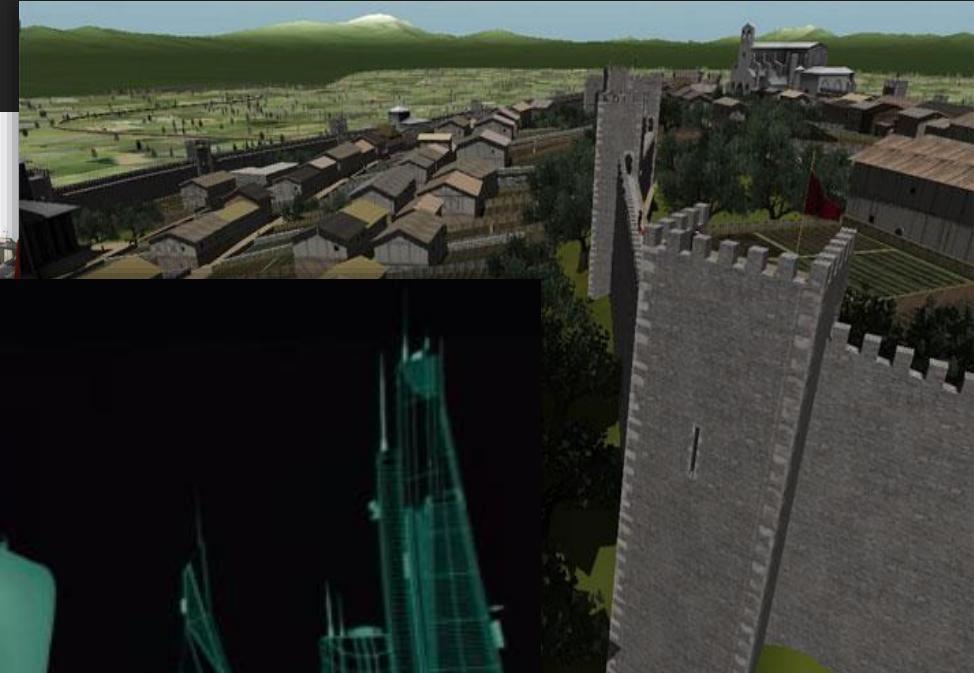
# 内容

- CityEngine 新技术速递
- 深入浅出CGA
- 使用python增强3D建模
- 3D内容导出与分享

# What is CityEngine?



# 应用领域



电影视觉

游戏场景

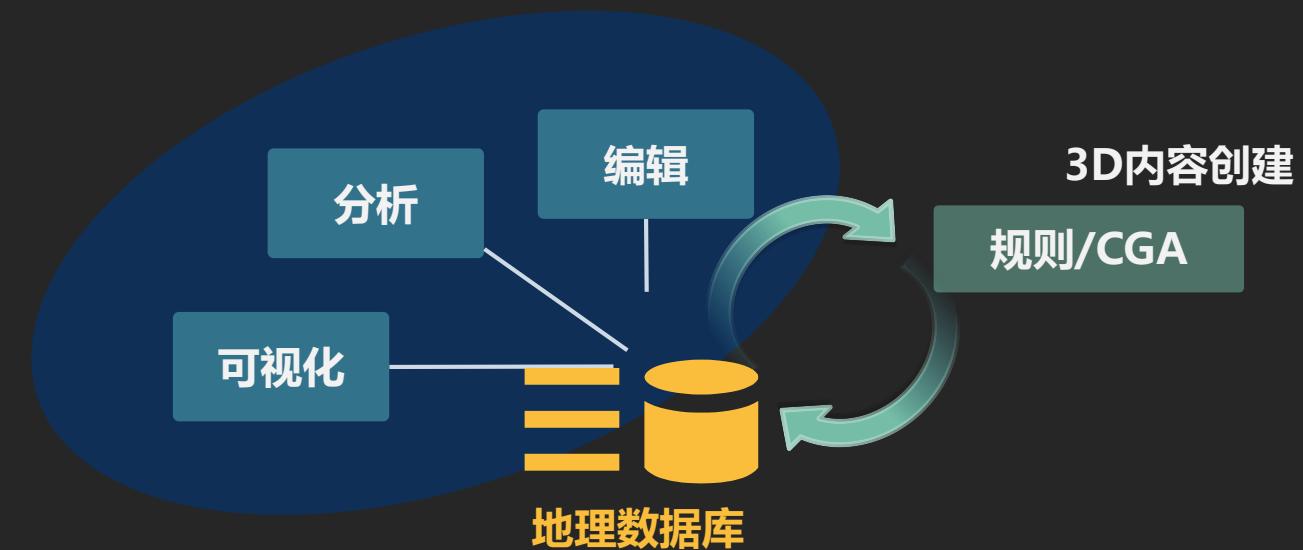
# 程序建模



# 核心技术

- GIS数据作为建模基础
  - 几何+属性

- 规则驱动创造三维模型
  - CGA

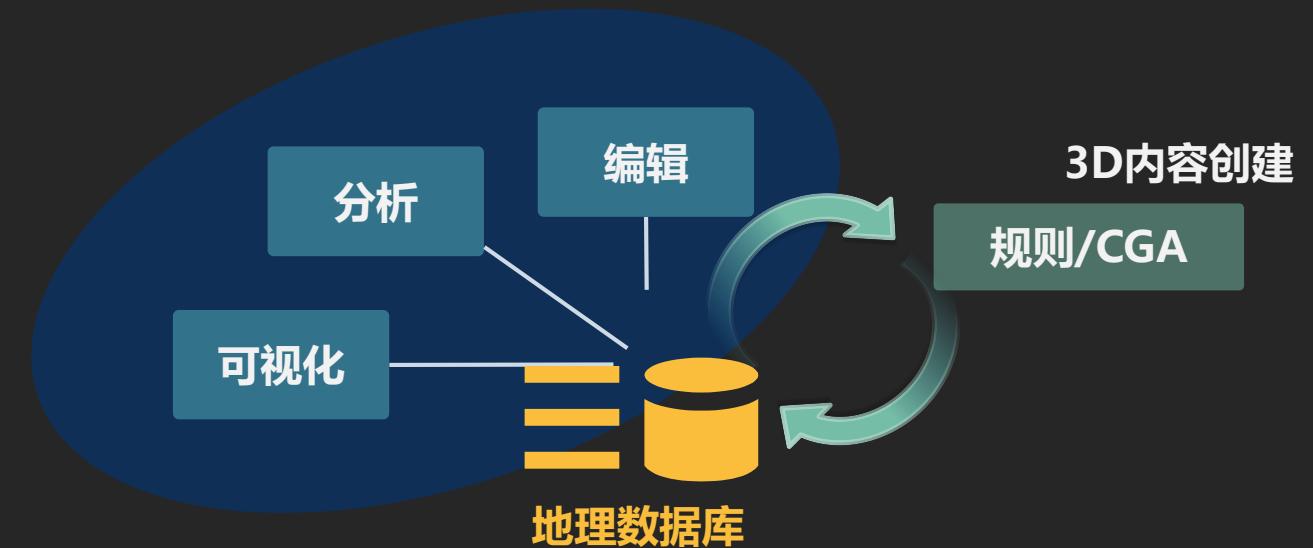


ArcGIS Pro - FeatureDataforCE											
Field:		Add	Delete	Calculate	Selection:	Zoom To	Switch	Clear	Delete		
		Building2_footprints3D									
OBJECTID	SHAPE	BuildingNa	Address	Type	Source	Z_Min	Z_Max	Z	Z	SHAPE_Lengt	
1	Polygon Z	GlaxoSmithKline	200 N 16th St, Philadelphia, PA 19102-1225	Commercial	Pictometry	8.2317	370.6847	37.31742	37.31742	725.516	
2	Polygon Z	GlaxoSmithKline	3 Franklin Plaza, Philadelphia, PA 19102	Commercial	Pictometry	8.1607	171.2461	34.97473	34.97474	1173.4150	
3	Polygon Z	Albanian Orthodox Church	237 N 17th St, Philadelphia, PA 19103-1201	Religious	Pictometry	8.1496	86.4337	34.40195	34.40195	298.8702	
4	Polygon Z	GlaxoSmithKline	3 Franklin Plaza, Philadelphia, PA 19102	Commercial	Pictometry	8.1823	46.726	34.31102	34.31102	46.050	
5	Polygon Z	Sheraton Philadelphia City...	237 N 17th St, Philadelphia, PA 19103	Commercial	Pictometry	8.1548	350.9131	36.79203	36.79202	1231.3947	
6	Polygon Z	Three Parkway	3 Benjamin Franklin Pkwy, Philadelphia, PA 1...	Commercial	Pictometry	8.1761	303.6733	35.36127	35.36127	1477.6642	
7	Polygon Z	Three Parkway	3 Benjamin Franklin Pkwy, Philadelphia, PA 1...	Commercial	Pictometry	8.2859	47.8489	34.5019	34.50189	73.10	
8	Polygon Z	Verizon	1631 Arch St, Philadelphia, PA 19103	Commercial	Pictometry	8.2155	308.5041	31.34247	31.34248	466.1722	
9	Polygon Z	Insurance Company of Nor...	1601 Chestnut St, Philadelphia, PA	Commercial	Pictometry	8.2367	421.323	32.3525	32.35246	1267.4648	
10	Polygon Z	Kennedy House	1901 JFK Blvd, Philadelphia, PA 19103	Commercial	Pictometry	30.4411	343.8876	34.55488	33.17279	1007.8662	
11	Polygon Z	CMS Building	1926 Arch St, Philadelphia, PA 19103	Commercial	Pictometry	24.4281	92.5583	25.77169	25.77169	643.2268	
12	Polygon Z	(demolished)	1910 Arch St, Philadelphia, PA 19103	Commercial	Pictometry	25.8244	58.7273	26.85561	26.85561	451.6970	
13	Polygon Z	(demolished)	1900 Arch St, Philadelphia, PA 19103	Commercial	Pictometry	25.3032	77.4199	28.77448	28.77447	426.9163	
14	Polygon Z	The Sterling	1815 JFK Blvd, Philadelphia, PA 19103	Commercial	Pictometry	31.2589	333.4618	34.56912	32.758	972.7860	
15	Polygon Z	Comcast Center	1701 JFK Blvd, Philadelphia, PA 19103	Commercial	Pictometry	28.5818	61.6377	35.78684	35.78685	462.2614	
16	Polygon Z	Bell Atlantic Tower	1717 Arch St, Philadelphia, PA 19103	Commercial	Pictometry	27.9202	758.9978	29.9143	29.9143	709.2614	
17	Polygon Z	Windor Suites	1700 Benjamin Franklin Pkwy, Philadelphia, P...	Commercial	Pictometry	21.1156	274.1688	30.65942	30.24113	1168.6173	
18	Polygon Z	Windor Suites	1700 Benjamin Franklin Pkwy, Philadelphia, P...	Commercial	Pictometry	29.8158	56.4651	30.29556	29.71733	121.5822	
19	Polygon Z	Embassy Suites Hotel	1776 Benjamin Franklin Pkwy, Philadelphia, P...	Commercial	Pictometry	26.855	314.3565	28.96533	28.96536	612.2926	
20	Polygon Z	Two Logan Square	100 N 18th St, Philadelphia, PA 19103	Commercial	Pictometry	25.1524	494.0972	27.4446	27.4446	863.8967	
21	Polygon Z	Logan Square Public Parking	1815 Cherry St, Philadelphia, PA 19103	Commercial	Pictometry	23.1221	105.5893	25.78696	25.78697	1153.2297	
22	Polygon Z	The Lofts at 1835 Arch	1835 Arch St, Philadelphia, PA 19103	Commercial	Pictometry	23.6395	261.1023	24.24143	24.24142	585.2169	
23	Polygon Z	Public Parking	113 N 19th St, Philadelphia, PA 19103	Commercial	Pictometry	21.5293	63.5771	23.88087	23.88087	524.3265	
24	Polygon Z	Four Seasons Hotel Philadel...	130 N 18th St, Philadelphia, PA 19103-2724	Commercial	Pictometry	20.6862	425.0168	24.24989	24.2499	1766.526	

# 核心技术

- GIS数据作为建模基础
  - 几何+属性

- 规则驱动创造三维模型
  - CGA



```
Complex_facade.cga X
@StartRule
Lot --> extrude(height) Building

Building --> comp(f){front : FrontfacadeTex}

FrontfacadeTex -->
    setupProjection(0, scope.xy, 2.25, 1.5, 1)
    setupProjection(2, scope.xy, '1, '1)
    Frontfacade

Frontfacade -->
    split(y){ groundfloor_height : Floor(split.index) // Groundfloor
    | floor_height : Floor(split.index) // First Floor
    | floor_height : Floor(split.index) // Second Floor
    | (~floor_height : Floor(split.index))* // Mid Floors
    | floor_height : Floor(999) // Top Floor, index
    | 0.5 : s('1,'1,0.3) LedgeAsset} // The top ledge just

Floor(floorindex) -->
    case floorindex == 0 :
        Subfloor(floorindex)
    case floorindex == 2 :
        split(y){~1 : Subfloor(floorindex) Balcony | 0.5 : TopLedge} # Add Balcony
    else :
        split(y){1 : BottomLedge(floorindex)
        | ~1 : Subfloor(floorindex) | 0.5 : TopLedge}

Subfloor(floorindex) -->
    split(x){ 0.5 : Wall(1)
    | { ~tile_width : Tile(floorindex) }*
    | 0.5 : Wall(1) }

Wall(walltype) -->
    // dark bricks with dirt
    case walltype == 1 :
        color(wallColor)
        texture(wall_tex)
        set(material.dirtmap, dirt_tex)
        projectUV(0) projectUV(2)
    // bright bricks with dirt
    case walltype == 2 :
        color(wallColor)
        texture(wall2_tex)
        set(material.dirtmap, dirt_tex)
        projectUV(0) projectUV(2)
    // dirt only
```



# 核心技术

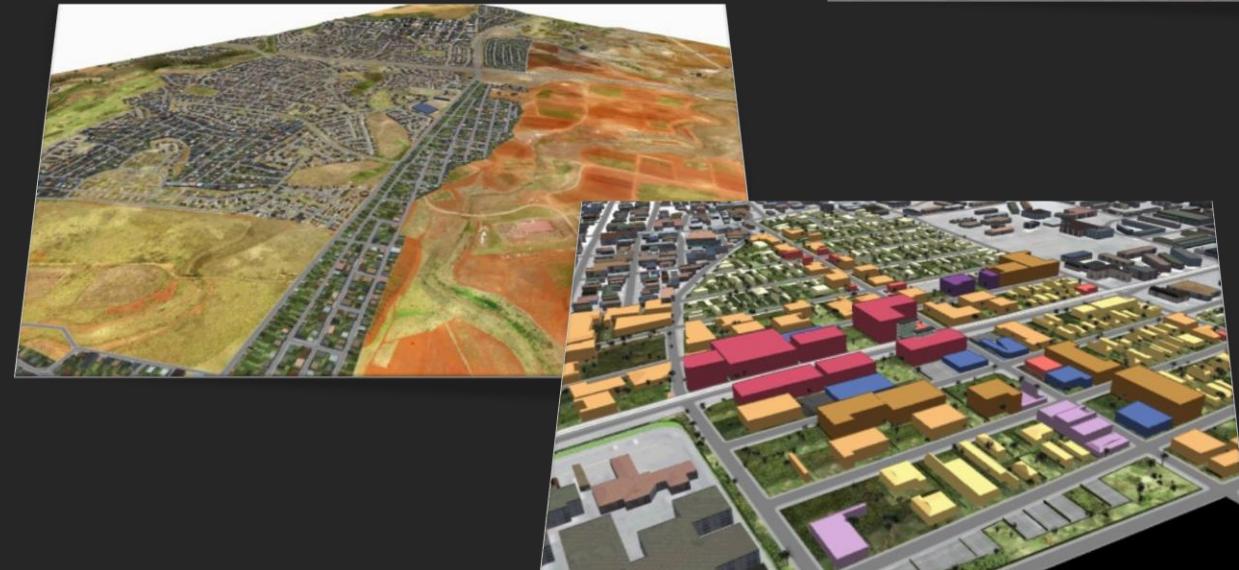
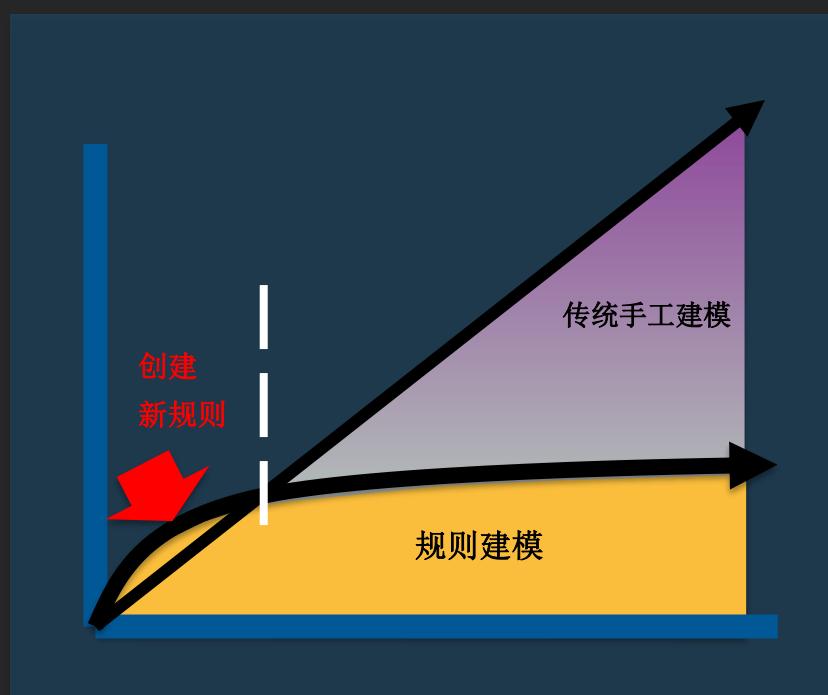
- GIS数据作为建模基础
  - 要素+属性

A



- 规则驱动创造三维模型
  - CGA
- 动态的智能编辑与布局
  - 参数调整、联动编辑

B





# CityEngine 新技术 速递

# Esri CityEngine

不断推陈出新

全新的软件界面

新增Scenarios

新增Local Edit

新增视域分析工具

优化Dashboard

优化CGA近邻查询

增强数据导入

.....

**2017**

支持导入FBX

增强图层管理模式

引入Procedural Handle

引入报表仪表盘

优化贴图效率

2015

深入集成ArcGIS

CityEngine SDK

提供植物库

支持VR

支持在线数据获取

优化OSM数据支持

云平台一体化

2013

增强植物库

新增道路库

新增建筑物库

-立面, 屋顶

增强CityEngine SDK

-Unity, Maya插件

2014

投影坐标系增强

导出FileGDB

CityEngine web scene

交互式手工建模

2011

进入Esri的怀抱

FileGDB导入

2012

2016

投影坐标系增强

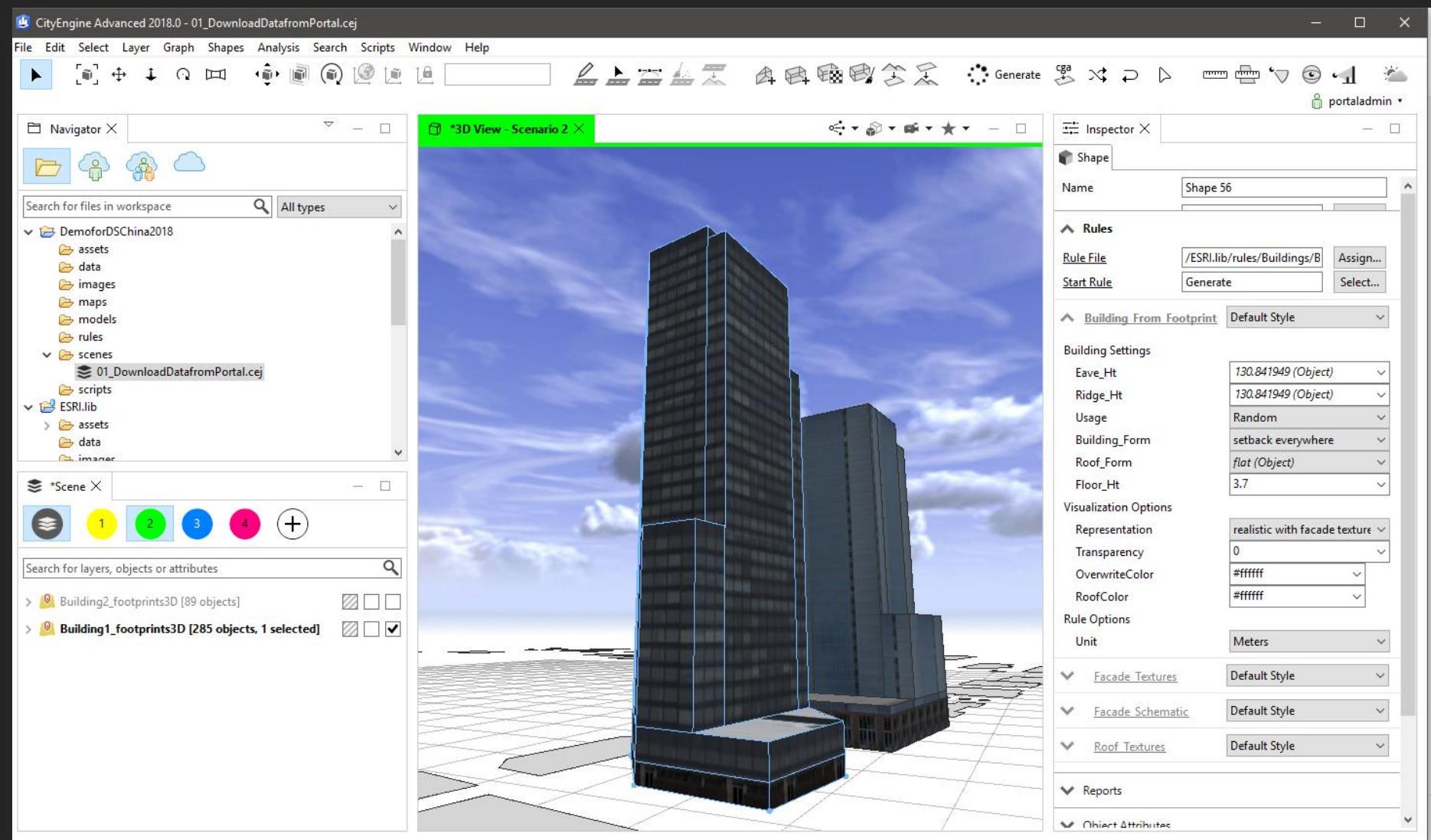
导出FileGDB

CityEngine web scene

交互式手工建模

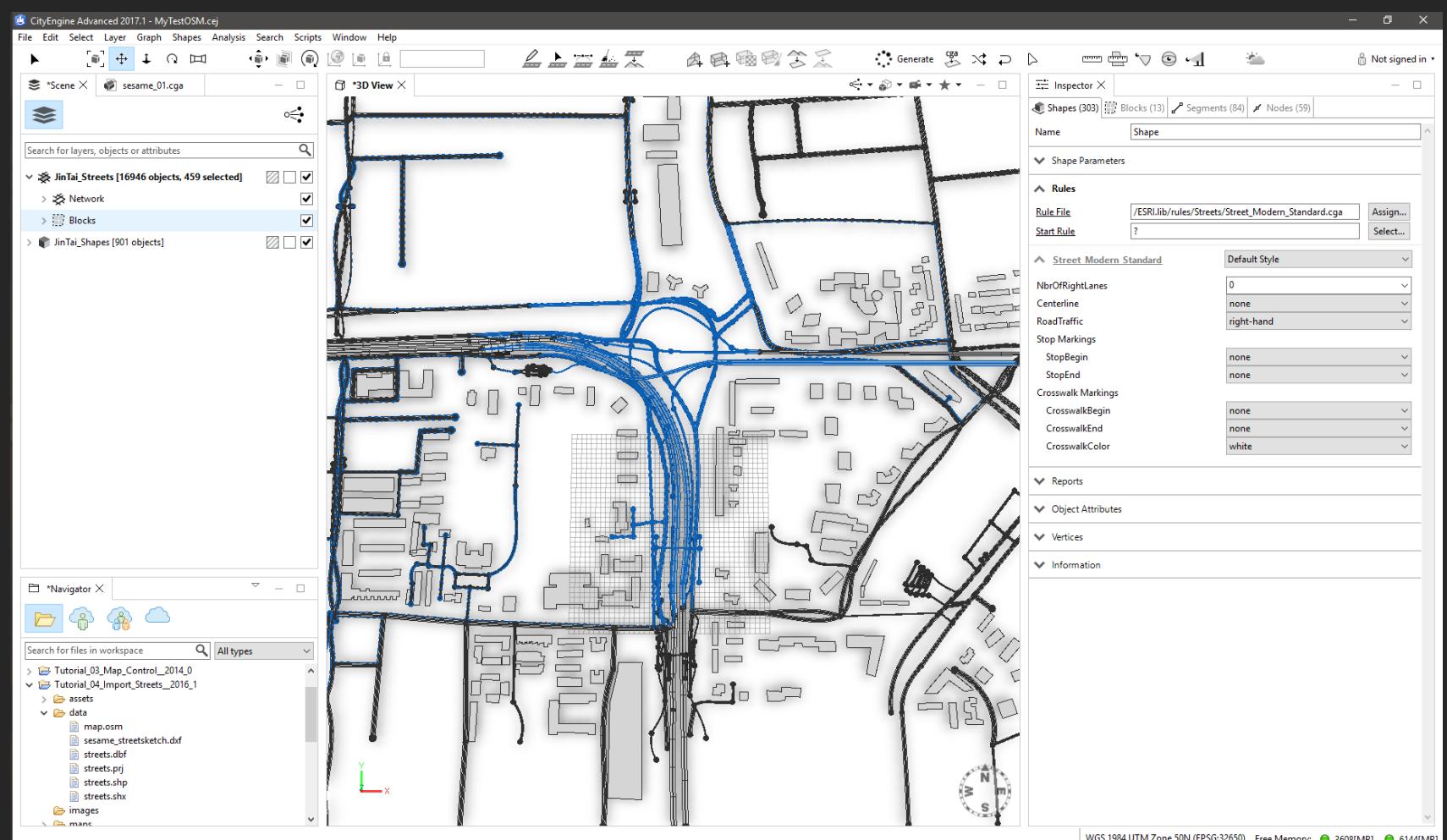
# 全新的UI

- 简洁
- 高效
- 专注
- 高DPI
- 改进菜单
- 工具重组
- 提升用户体验
- .....



# 更丰富的GIS数据来源

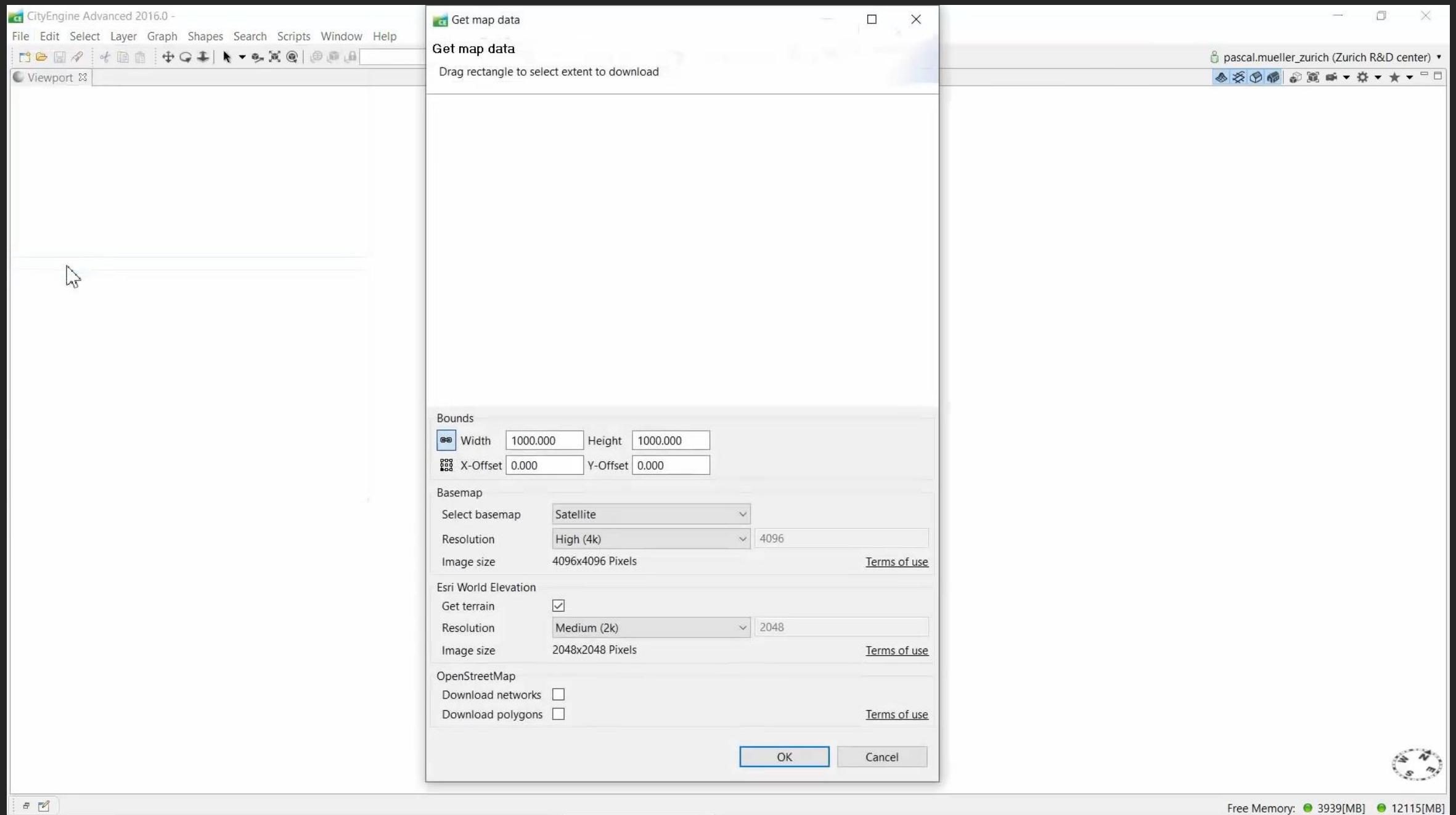
- File Geodatabase (.gdb)
- Shapefile
- DXF
- Open Street Map (.osm)



# 更丰富的GIS数据来源

*Update at CityEngine 2017.0*

- 获取云端影像底图及OSM数据



Free Memory: 3939[MB] 12115[MB]

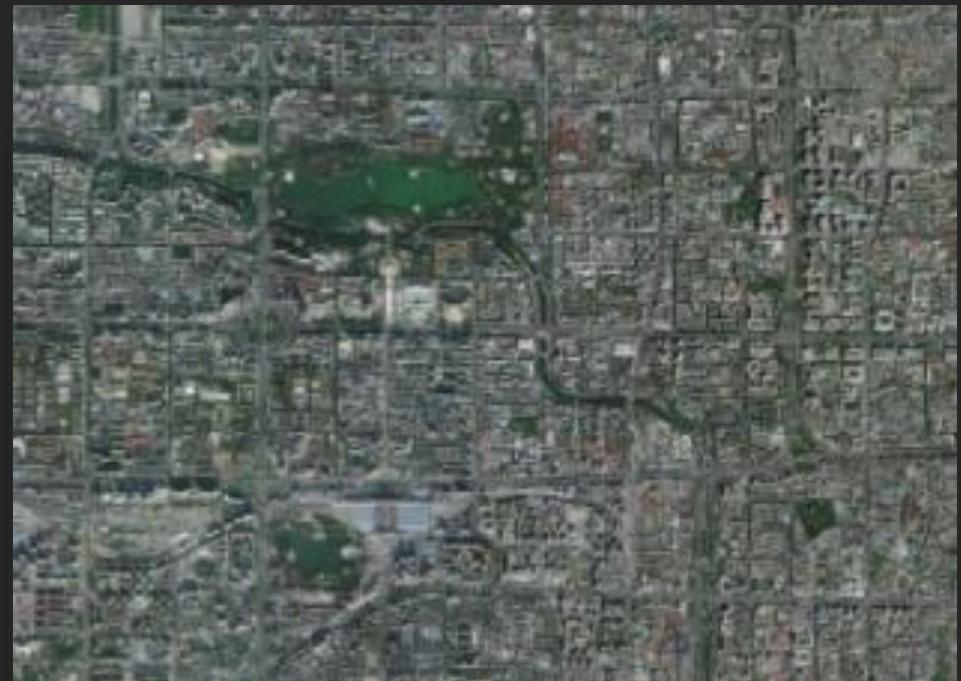
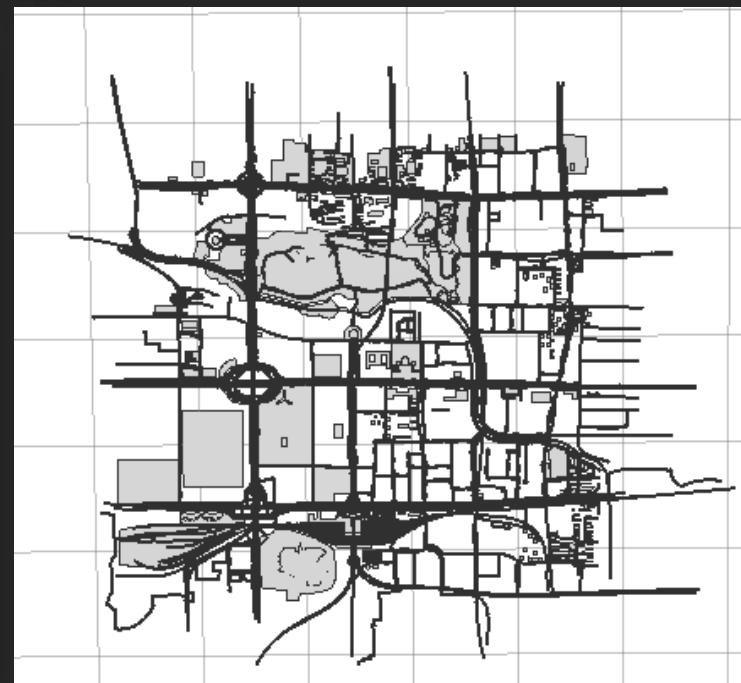
# 更丰富的GIS数据来源

- 获取云端影像底图及OSM数据

- ✓ elevation.tif
- ✓ map.osm
- ✓ Texture.jpg



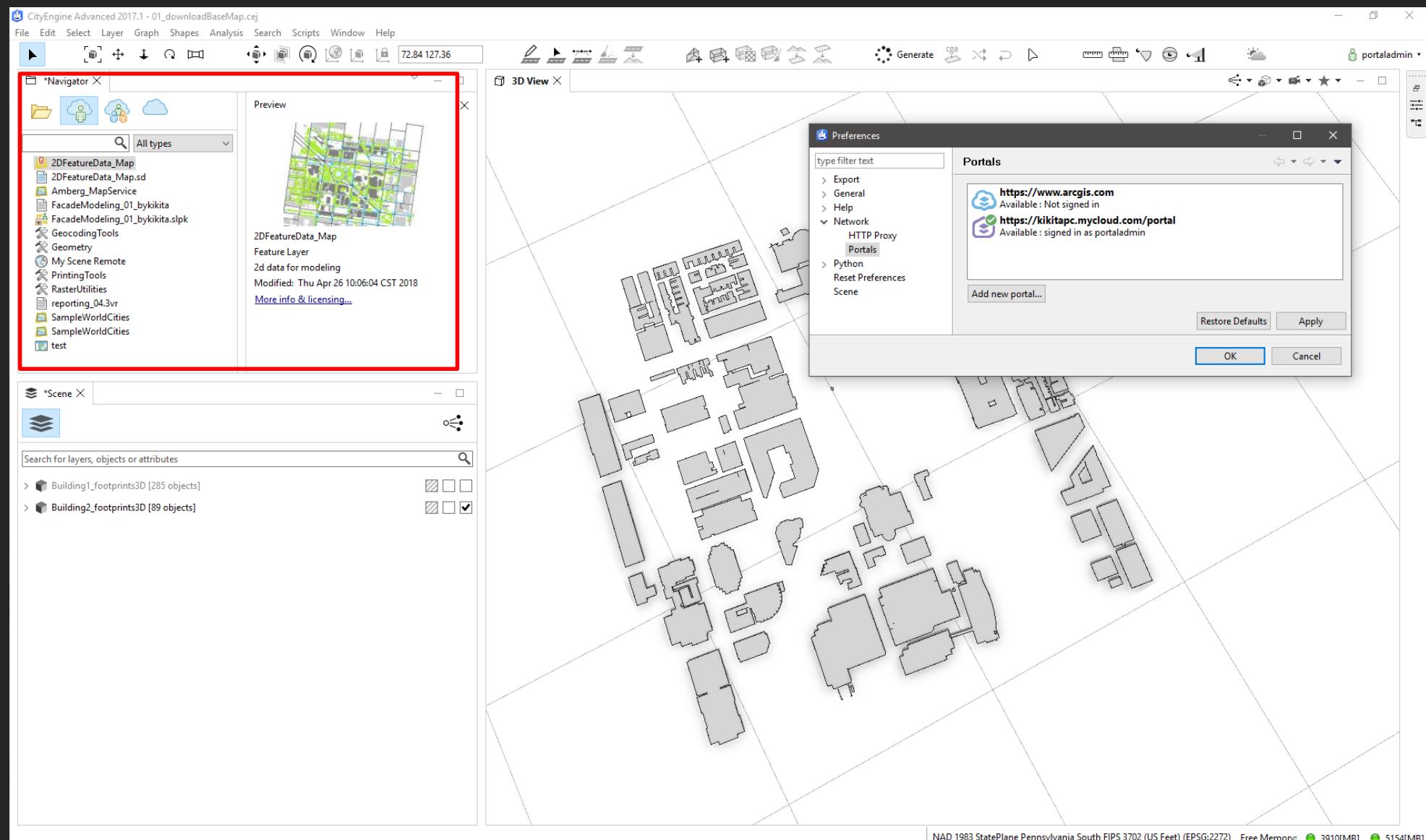
中国科技会堂周边



# 更丰富的GIS数据来源

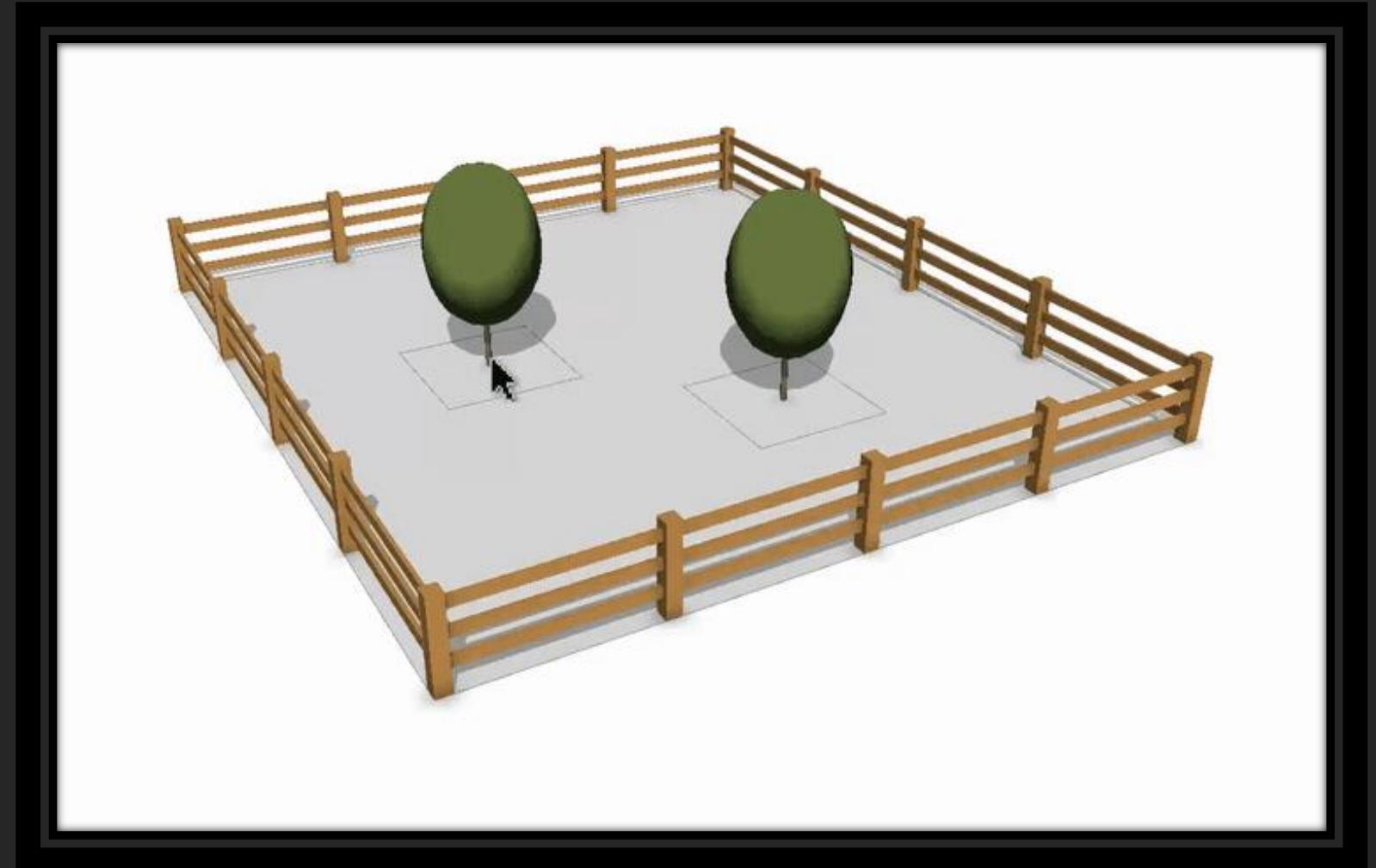
*New at CityEngine 2017.1*

- 导入来自 ArcGIS Online 或 Portal for ArcGIS 中的 web feature layer
- 多设备编辑、同步 ArcGIS Online web feature layer



# 更贴合规划需求的建模体验

- CGA: 新增近邻分析逻辑
- *New at CityEngine 2017.0*



# 更贴合规划需求的建模体验

*New at CityEngine 2017.0*

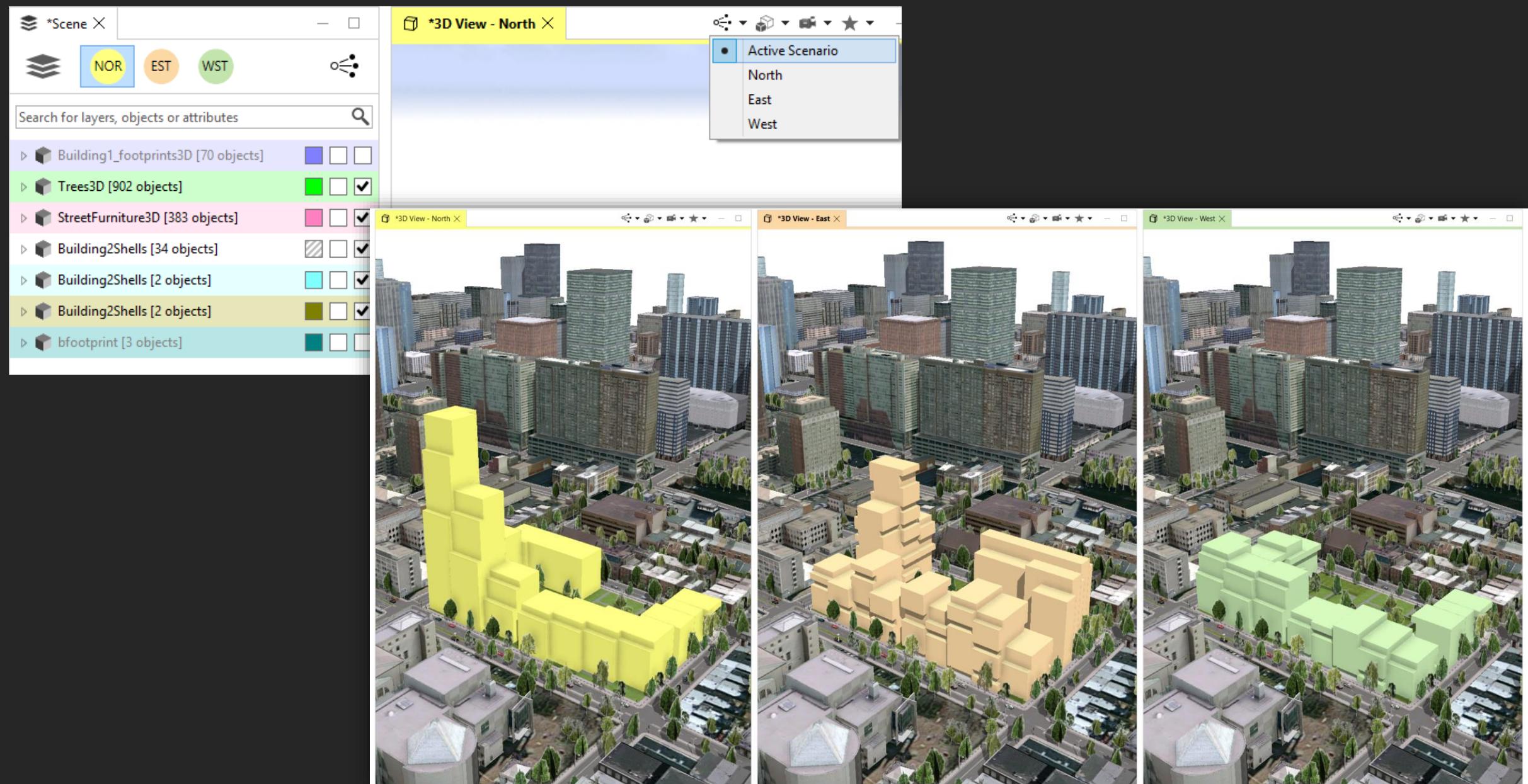
- Local Edit
  - 精细局部调整
- 结合 Procedural Handles



# 多场景规划方案对比

*New at CityEngine 2017.0*

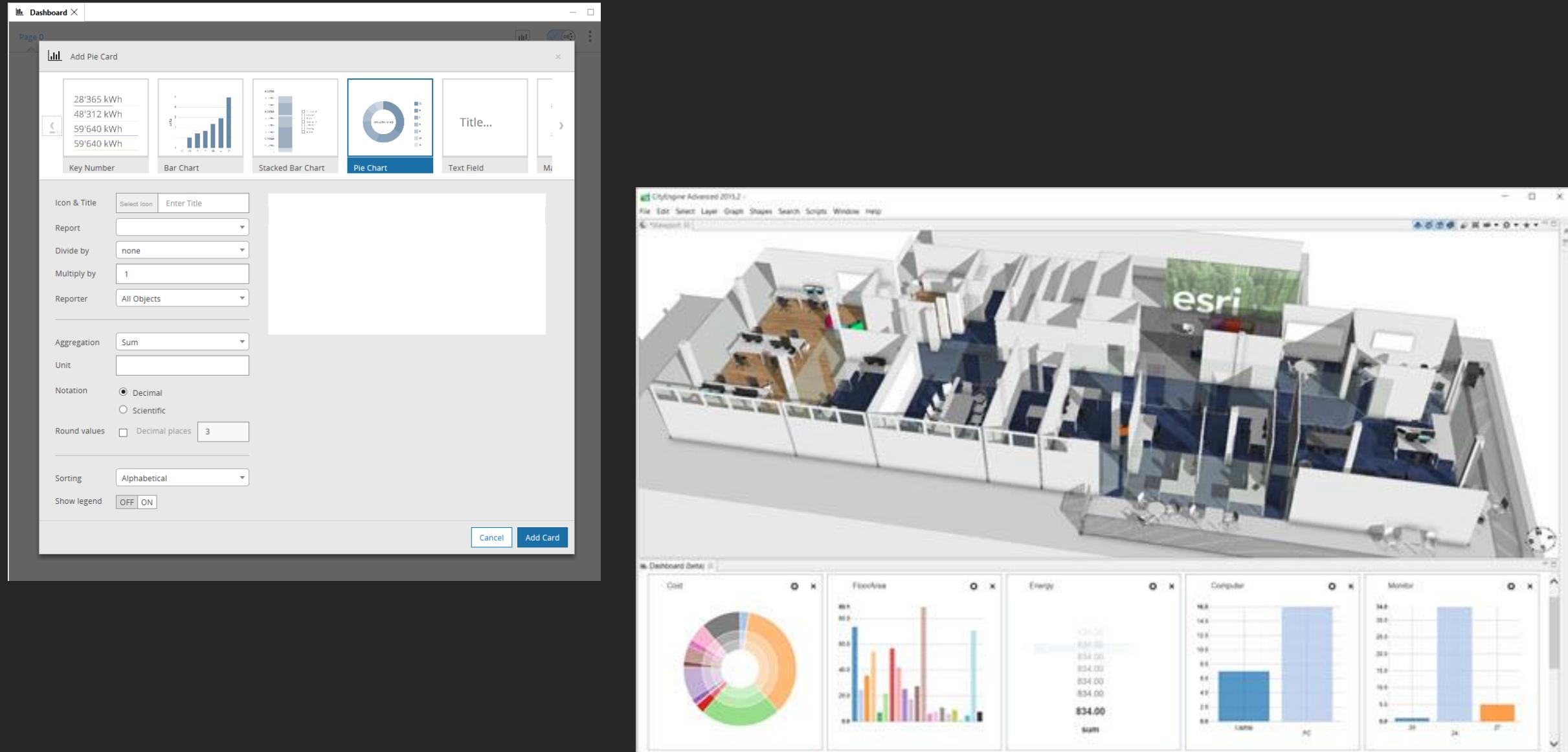
- Scenario



# 更自由可配的仪表盘

*Update at CityEngine 2017.0*

- Dashboard



# 场景内实时量测

*New at CityEngine 2017.0*

- Measure Tool



# 更智能的3D视域分析

*New at CityEngine 2017.1*

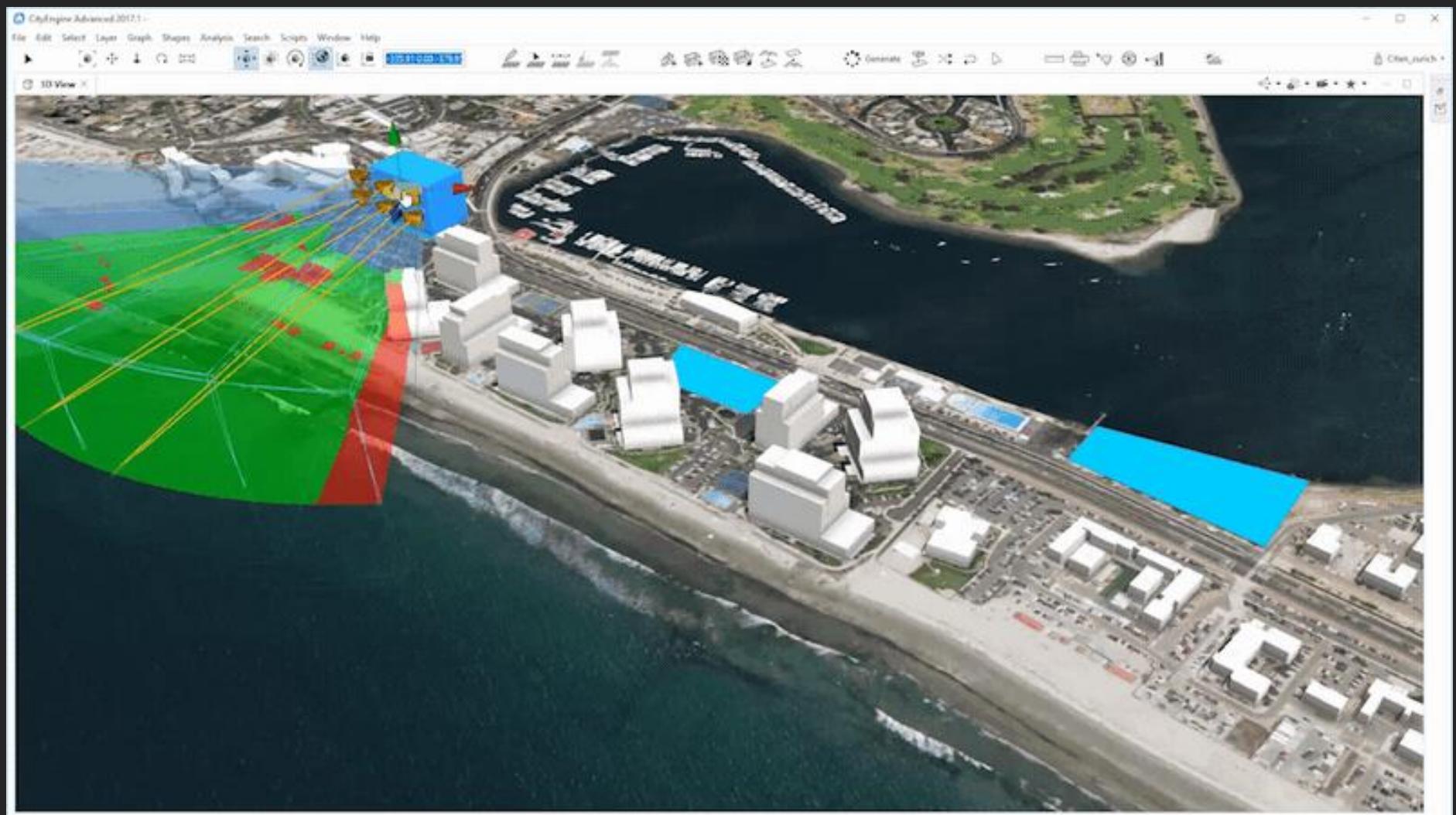
- 实时可见性分析
  - 分析工具
  - 实时
  - 交互式
  - 支持多视点
  - 与场景一同保存
  - 可以使用Python控制

# 更智能的3D视域分析

*New at CityEngine 2017.1*

- 实时可见性分析

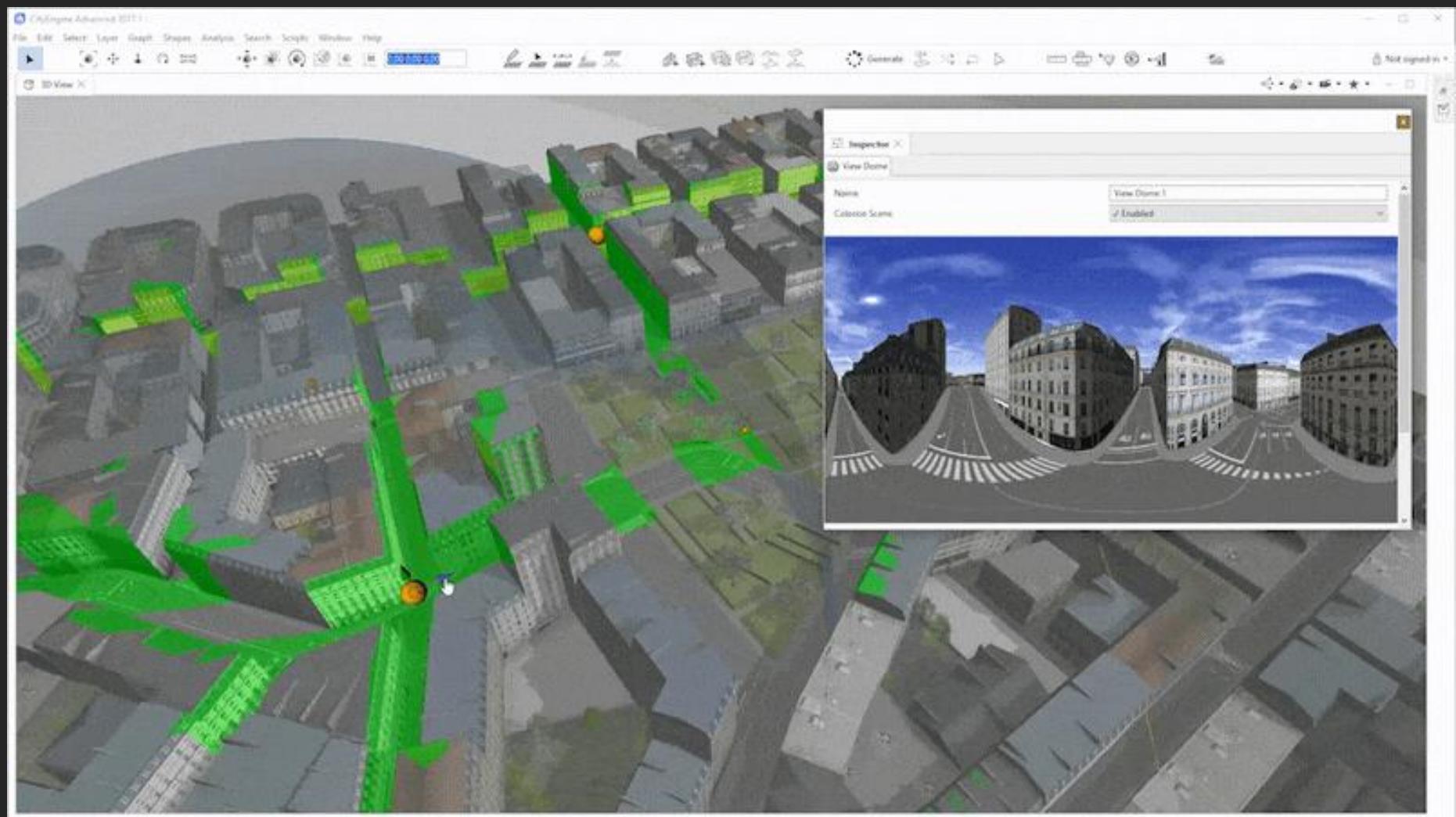
- Viewshed



# 更智能的3D视域分析

*New at CityEngine 2017.1*

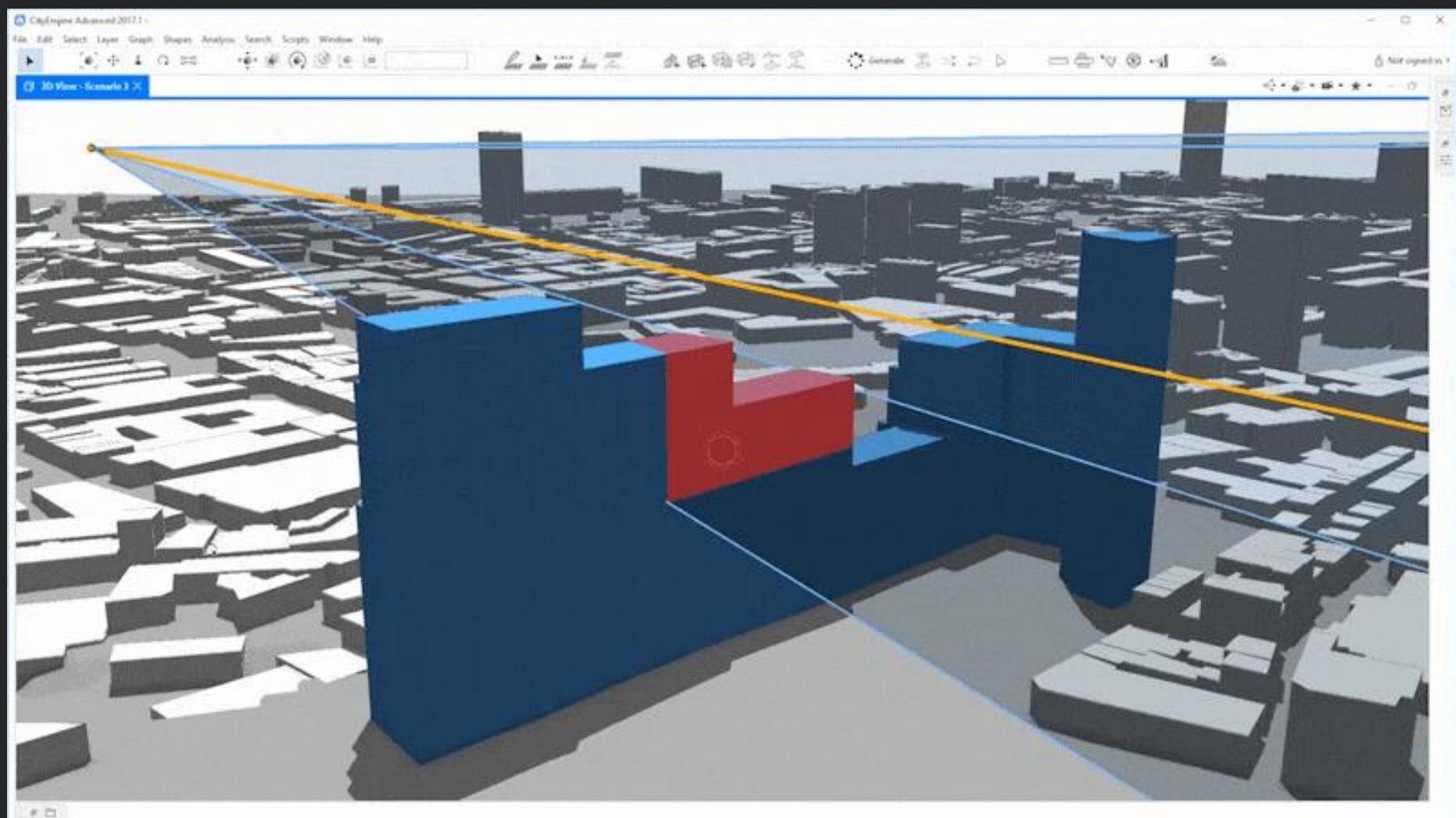
- 实时可见性分析
- View Dome



# 更智能的3D视域分析

*New at CityEngine 2017.1*

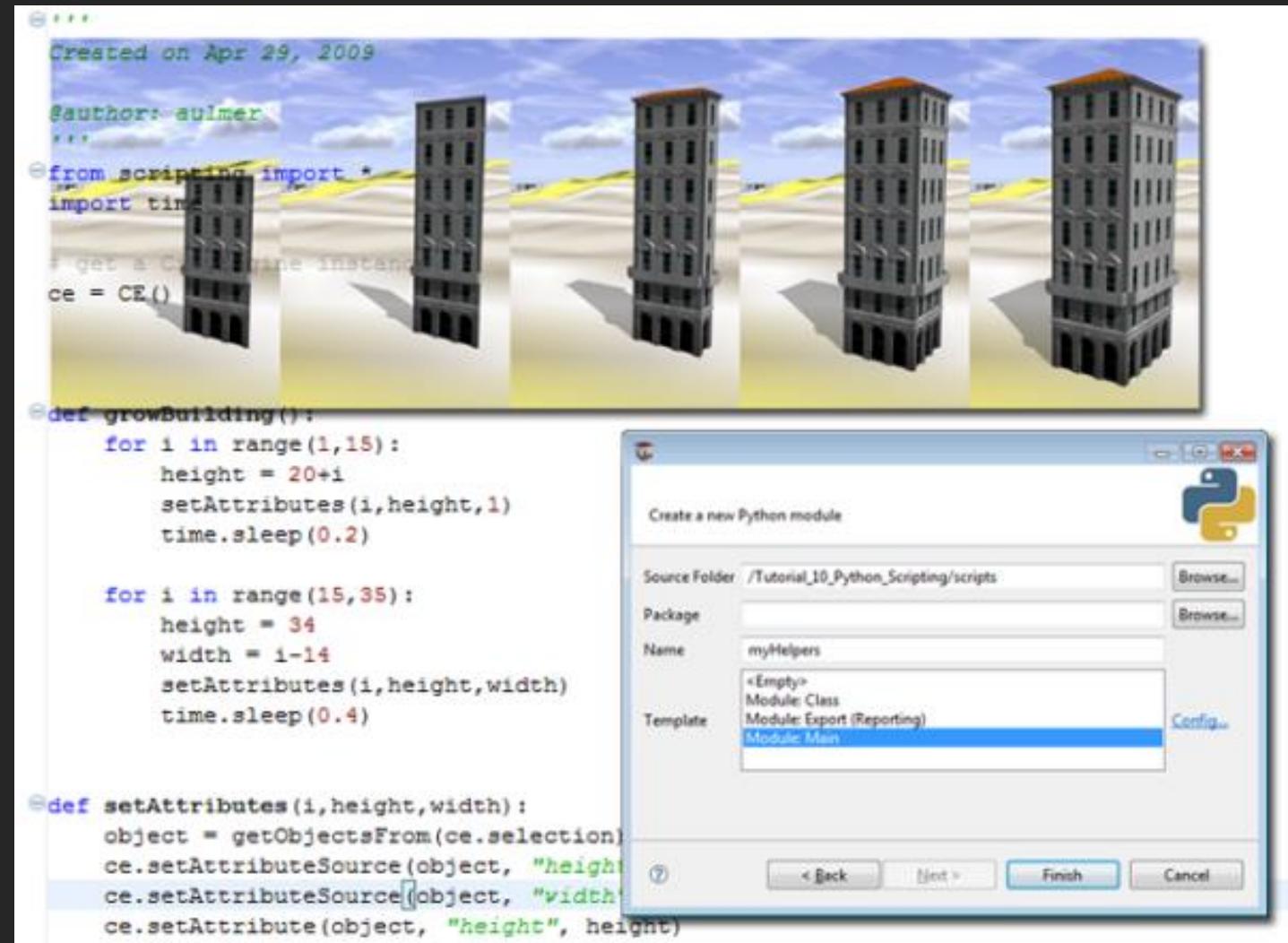
- 实时可见性分析
- View Corridor



# 增强 Python API

*New at CityEngine 2017.1*

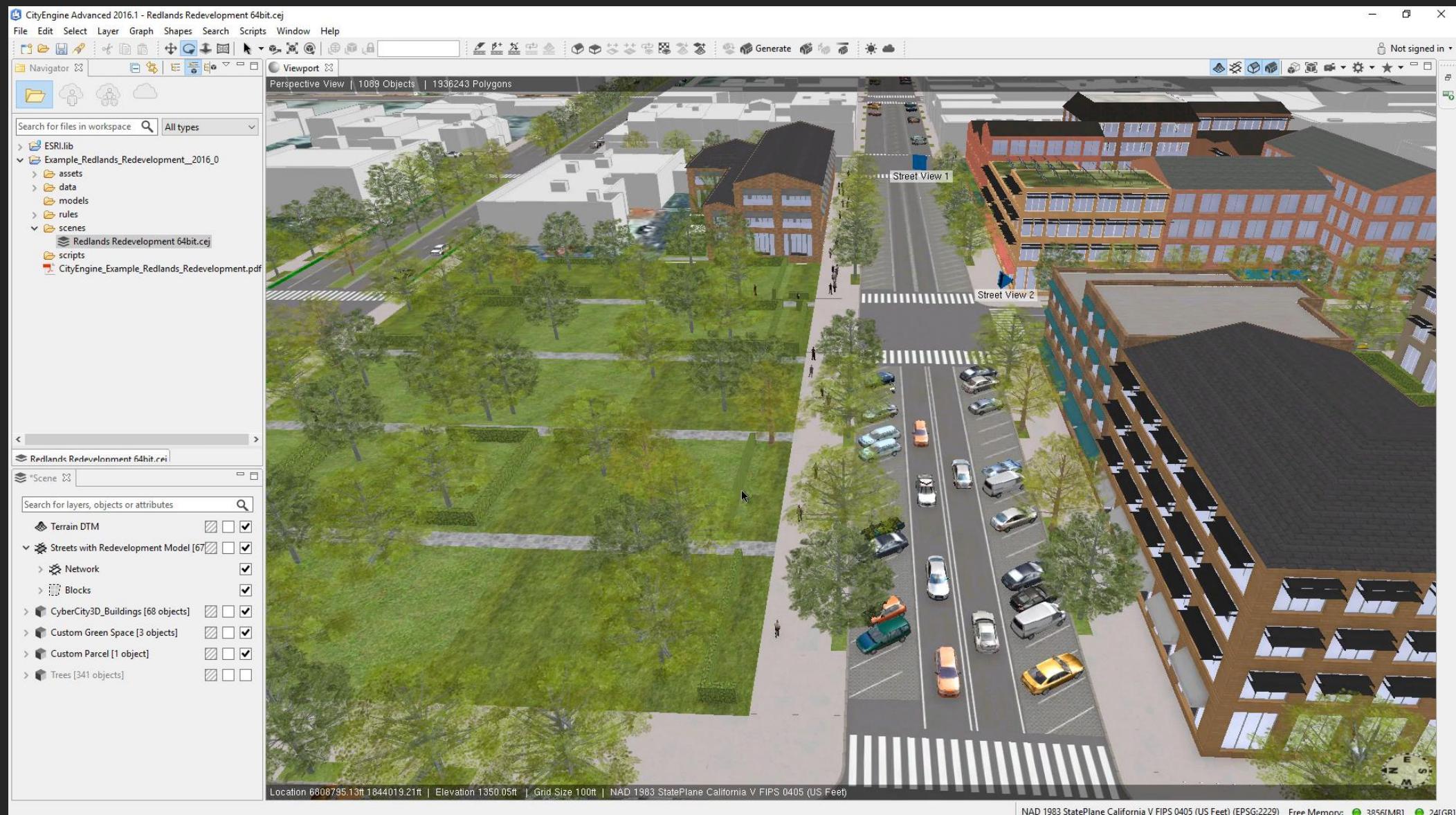
scripting.py 优化自动导入自定义模块



# 增强支持VR

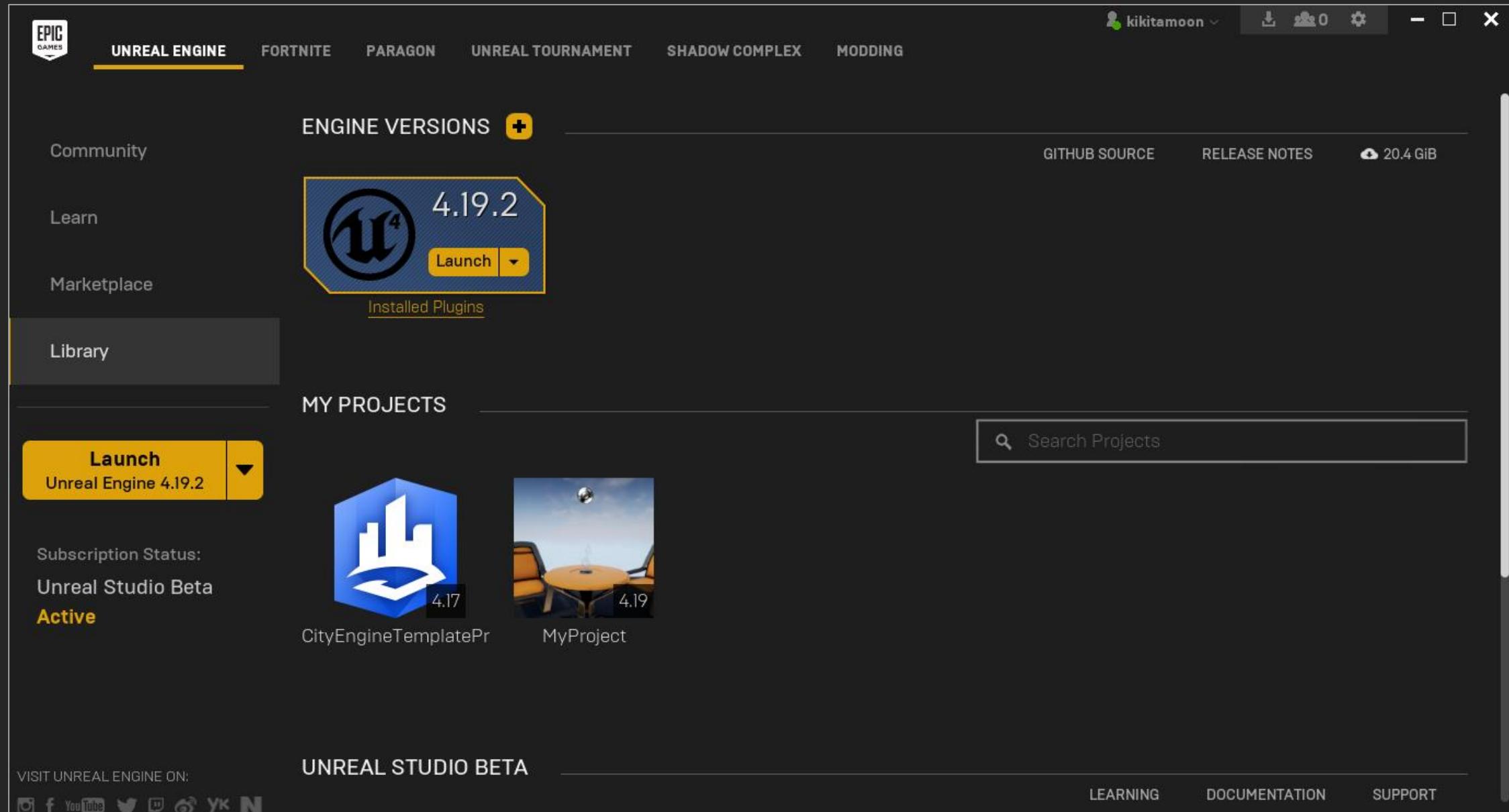
*Update at CityEngine 2017.1*

- 360 VR Experiences



# 增强支持Unreal

- 导出模型至高端视觉引擎 Unreal





深入浅出CGA

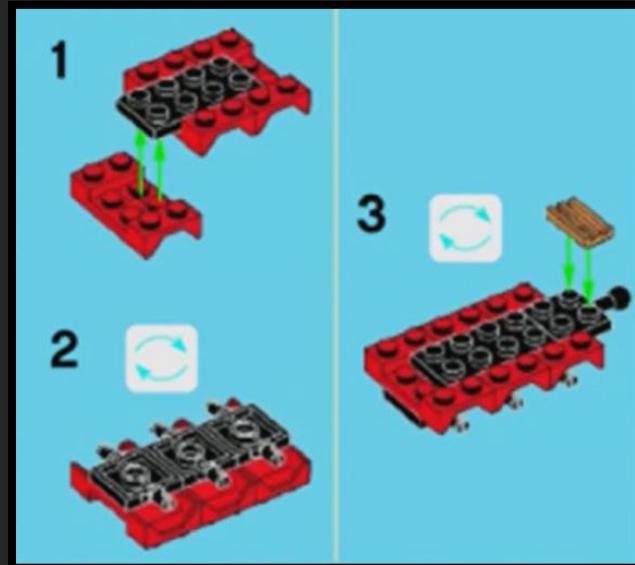
# 什么是CGA?

- **CGA**: Computer Generated Architecture
- CityEngine的独特脚本语言
- 越来越详细地迭代细化模型



# 理解CGA

- “乐高积木拼装图纸”

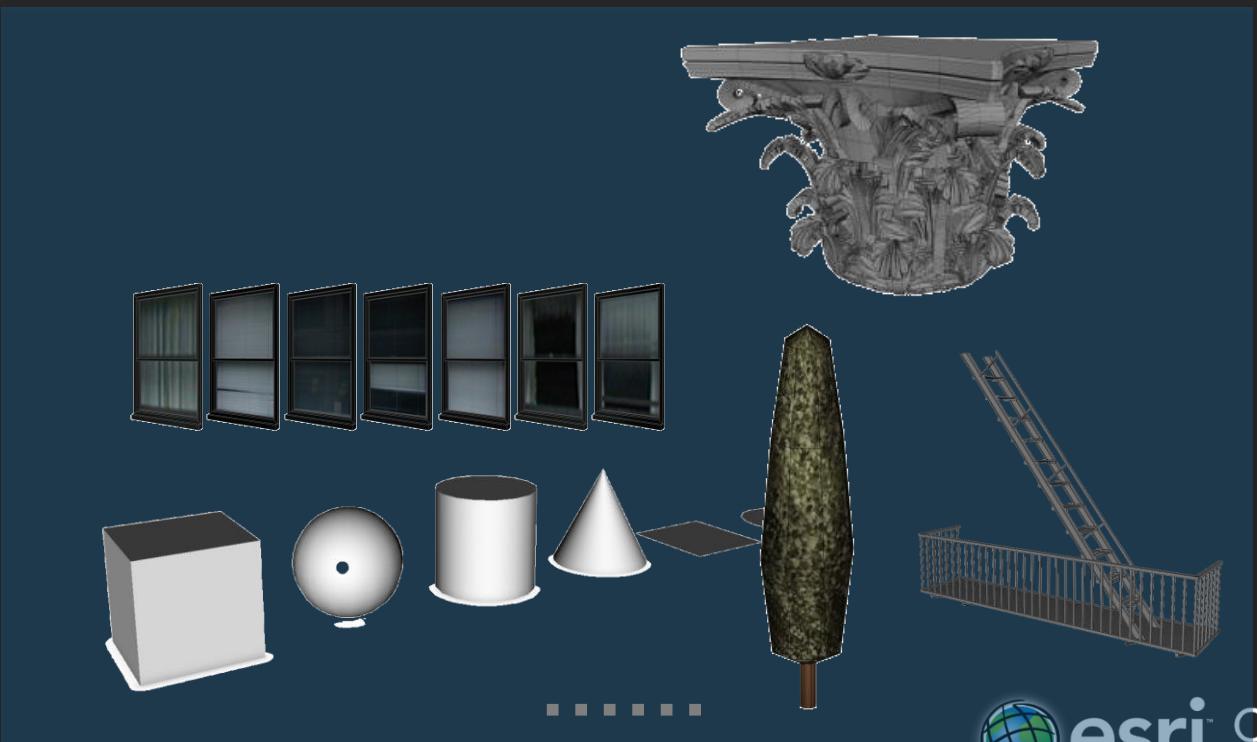


+



```
Complex_facade.cga ×  
@StartRule  
Lot --> extrude(height) Building  
  
Building --> comp(f)(front : FrontfacadeTex)  
  
FrontfacadeTex -->  
    setupProjection(0, scope.xy, 2.25, 1.5, 1)  
    setupProjection(2, scope.xy, '1, '1)  
    Frontfacade  
  
Frontfacade -->  
    split(y){  
        groundfloor_height : Floor(split.index) // Groundfloor  
        | floor_height : Floor(split.index) // First Floor  
        | floor_height : Floor(split.index) // Second Floor  
        | (-floor_height : Floor(split.index))* // Mid Floors  
        | floor_height : Floor(999) // Top Floor, index  
        | 0.5 : s('1,'1,0.3) LedgeAsset  
        | The top ledge just  
    }  
  
Floor(floorindex) -->  
    case floorindex == 0 :  
        Subfloor(floorindex)  
    case floorindex == 2 :  
        split(y){~1 : Subfloor(floorindex) Balcony | 0.5 : TopLedge} # Add Balcony  
    else :  
        split(y){1 : BottomLedge(floorindex)  
        | ~1 : Subfloor(floorindex) | 0.5 : TopLedge}  
  
Subfloor(floorindex) -->  
    split(x){ 0.5 : Wall(1)  
    | { -tile_width : Tile(floorindex) }*  
    | 0.5 : Wall(1) }  
  
Wall(walltype) -->  
    // dark bricks with dirt  
    case walltype == 1 :  
        color(wallcolor)  
        texture(wall_tex)  
        set(material.dirtmap, dirt_tex)  
        projectUV(0) projectUV(2)  
    // bright bricks with dirt  
    case walltype == 2 :  
        color(wallcolor)  
        texture(wall_tex)  
        set(material.dirtmap, dirt_tex)  
        projectUV(0) projectUV(2)  
    // dirt only  
    />
```

+



# 理解CGA

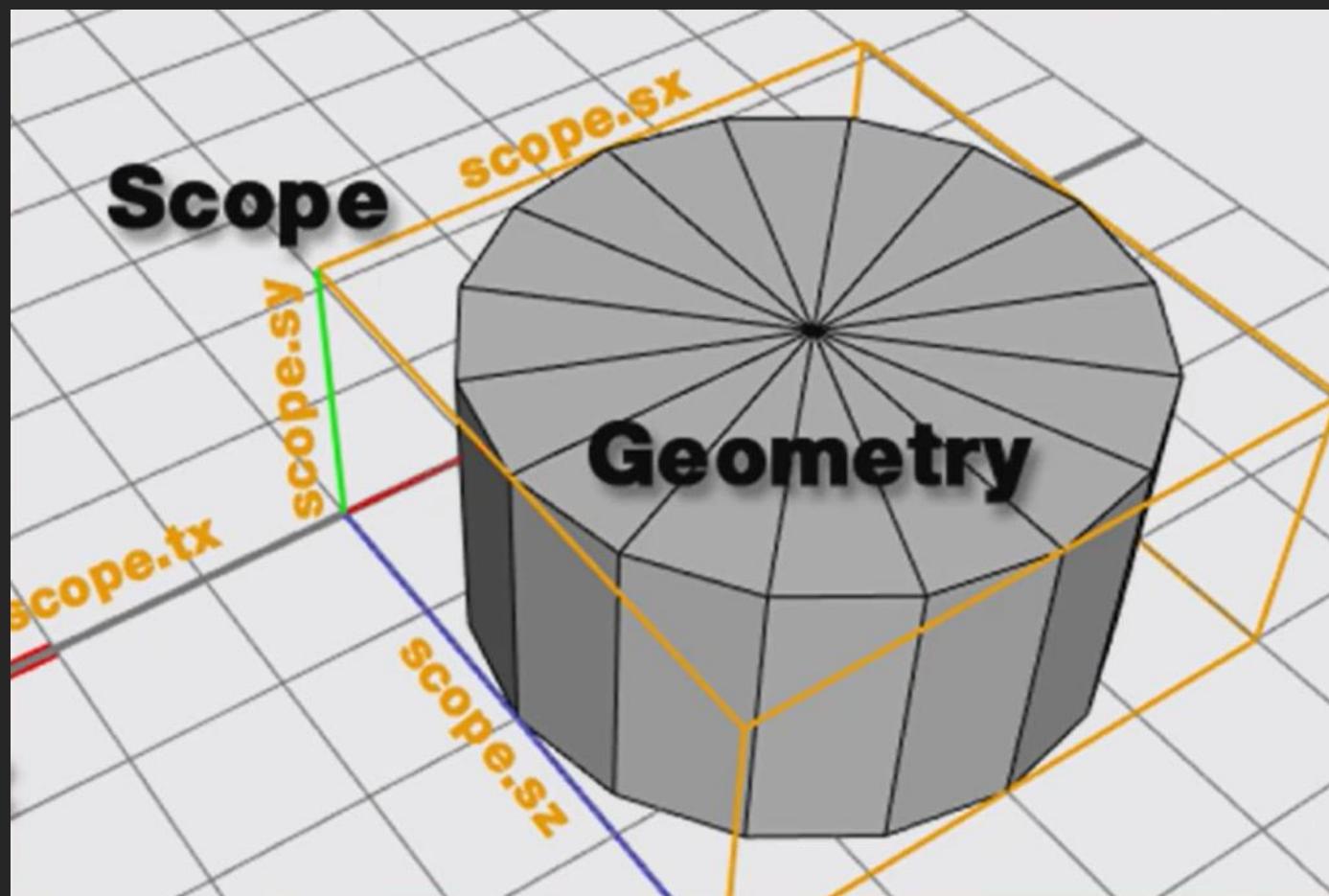
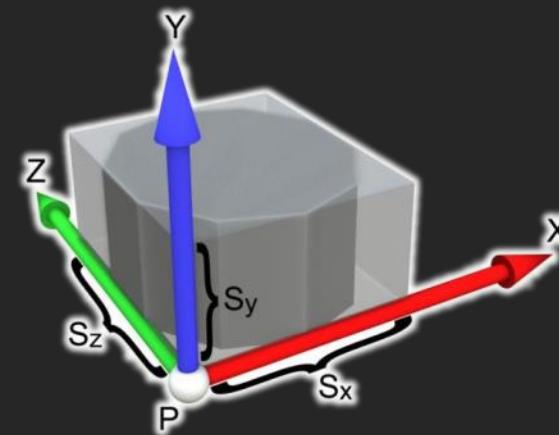
- CGA真的能创建非常复杂的模型吗？



# 什么是shape?

CityEngine Shape

Shape = Scope + Geometry



# CGA能做什么？

- CGA规则 = 对shape的一系列操作
- 修改 scope 和 geometry
- 函数示例：

Operations			
Geometry Creation			
<a href="#">i (insert)</a>	<a href="#">extrude</a>	<a href="#">envelope</a>	<a href="#">taper</a>
<a href="#">roofGable</a>	<a href="#">roofHip</a>	<a href="#">roofPyramid</a>	<a href="#">roofShed</a>
<a href="#">innerRectangle</a>	<a href="#">primitiveQuad</a>	<a href="#">primitiveDisk</a>	<a href="#">primitiveCube</a>
<a href="#">primitiveSphere</a>	<a href="#">primitiveCylinder</a>	<a href="#">primitiveCone</a>	
Geometry Subdivision			
<a href="#">split</a>	<a href="#">splitArea</a>	<a href="#">comp</a>	<a href="#">offset</a>
<a href="#">setback</a>	<a href="#">shapeL</a>	<a href="#">shapeU</a>	<a href="#">shapeO</a>
<a href="#">scatter</a>			

Texturing			
<a href="#">texture</a>	<a href="#">setupProjection</a>	<a href="#">projectUV</a>	<a href="#">translateUV</a>
<a href="#">scaleUV</a>	<a href="#">normalizeUV</a>	<a href="#">tileUV</a>	<a href="#">rotateUV</a>
<a href="#">deleteUV</a>			

Transformations			
<a href="#">t (scope translate)</a>	<a href="#">translate</a>	<a href="#">s (scope size)</a>	<a href="#">r (scope rotate)</a>
<a href="#">rotate</a>	<a href="#">center</a>		

Scope			
<a href="#">alignScopeToAxes</a>	<a href="#">alignScopeToGeometry</a>	<a href="#">rotateScope</a>	<a href="#">setPivot</a>
<a href="#">mirrorScope</a>			

# Esri Lib 规则库

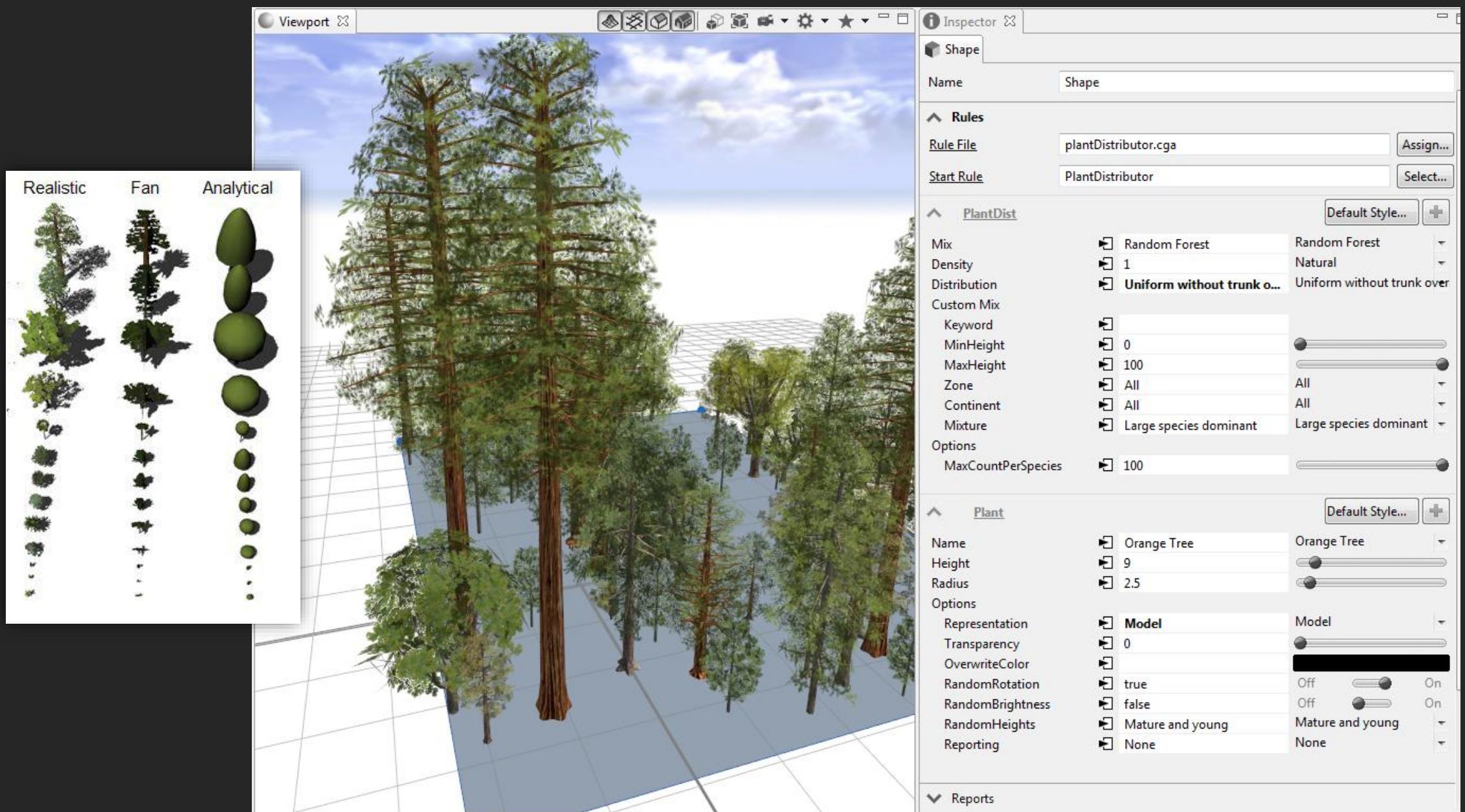
“即拿即用”

Build-in Esri rule library

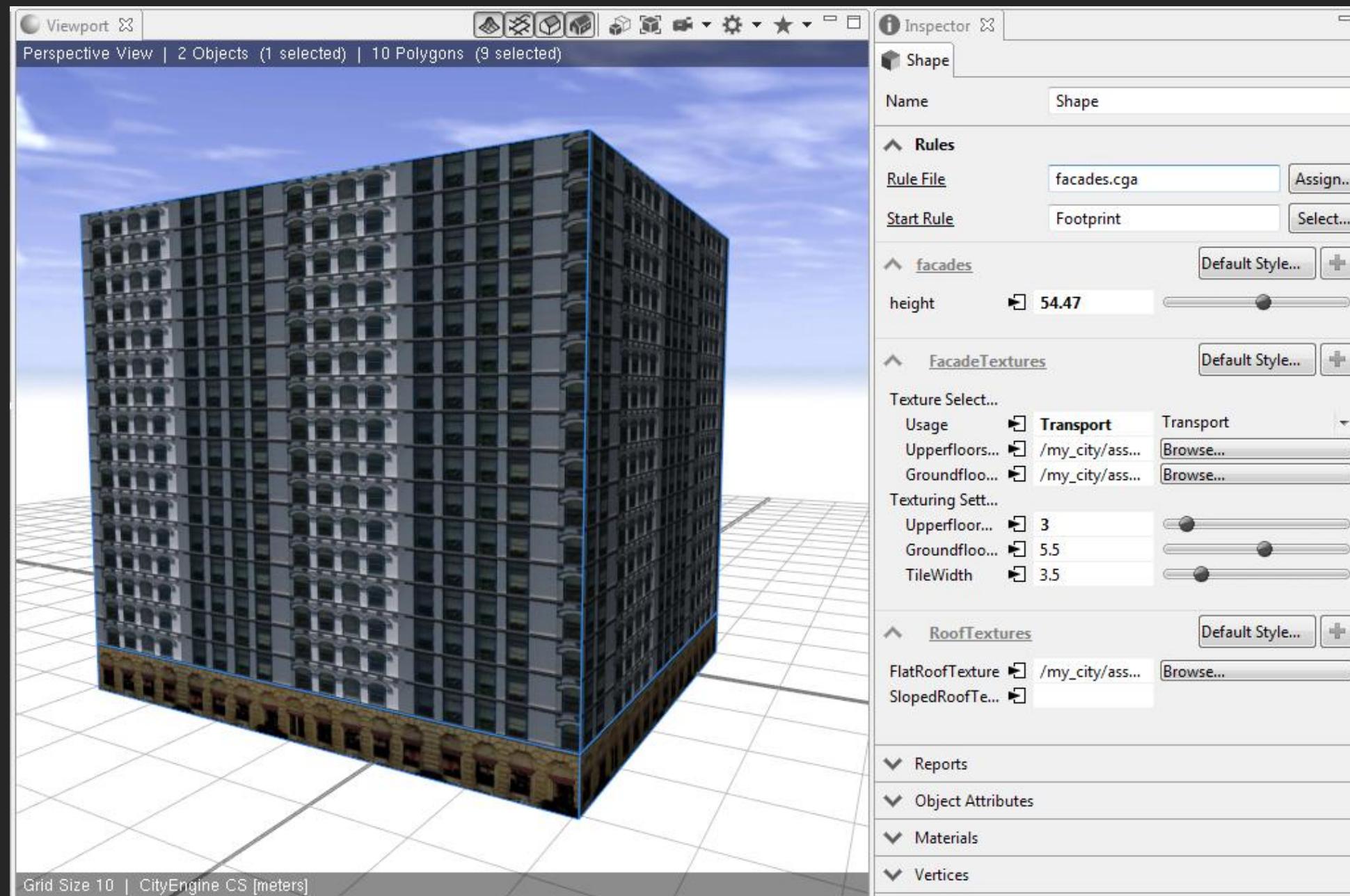
- 3D Vegetation
- Facade Textures
- Roof Textures
- Street Construction
- Utilities

```
import PlantDis:"/ESRI.lib/rules/Plants/Plant_Distributor.cga"  
  
Plants -->  
    PlantDis.Generate
```

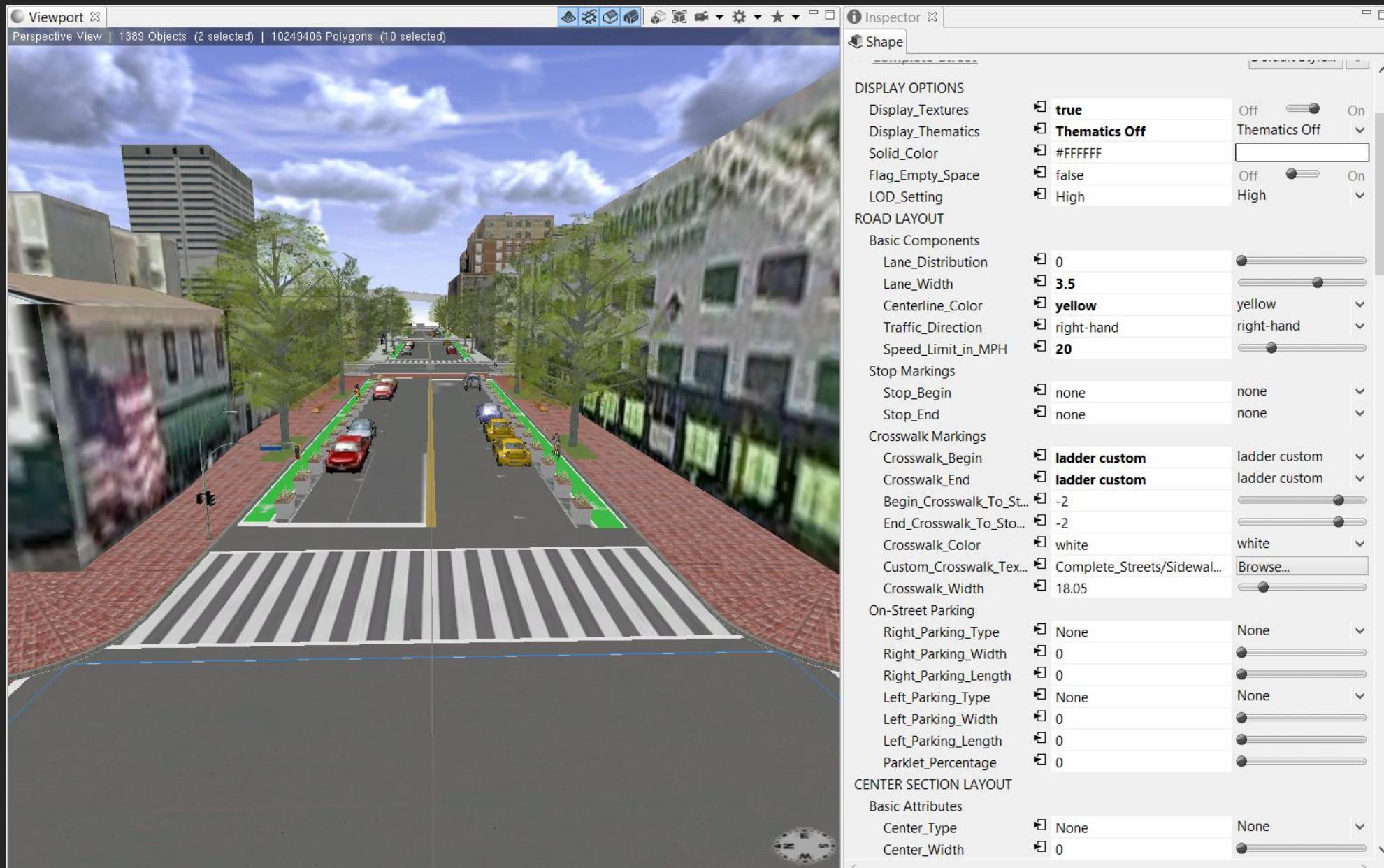
# 植物库



# 立面与屋顶库



# 街道库





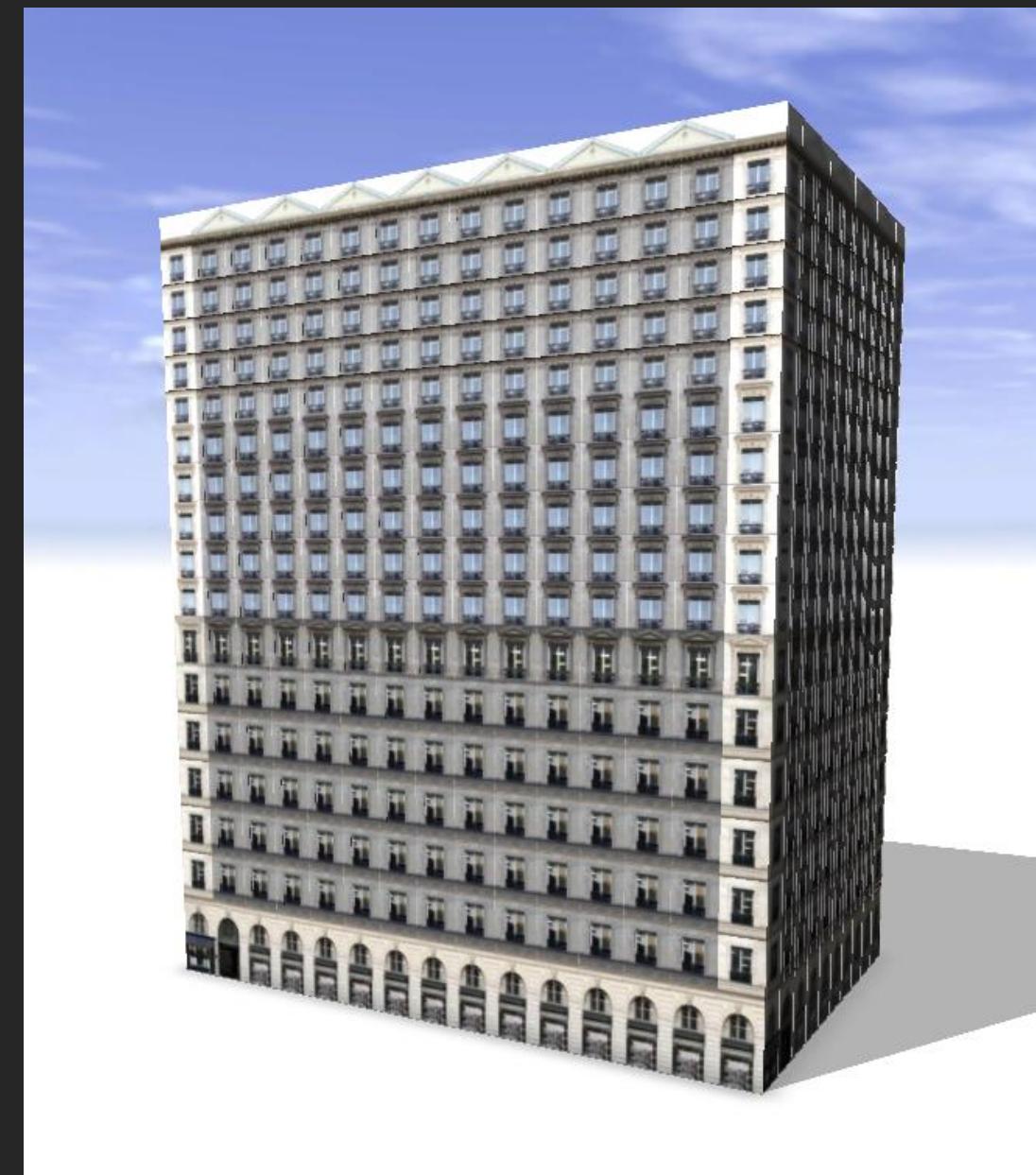
### Demo

- CGA建模基础示例
- 引用 Esri Lib 快速搭建场景

# 智能立面建模工具

- Façade Wizard

庖丁解牛  
触类旁通

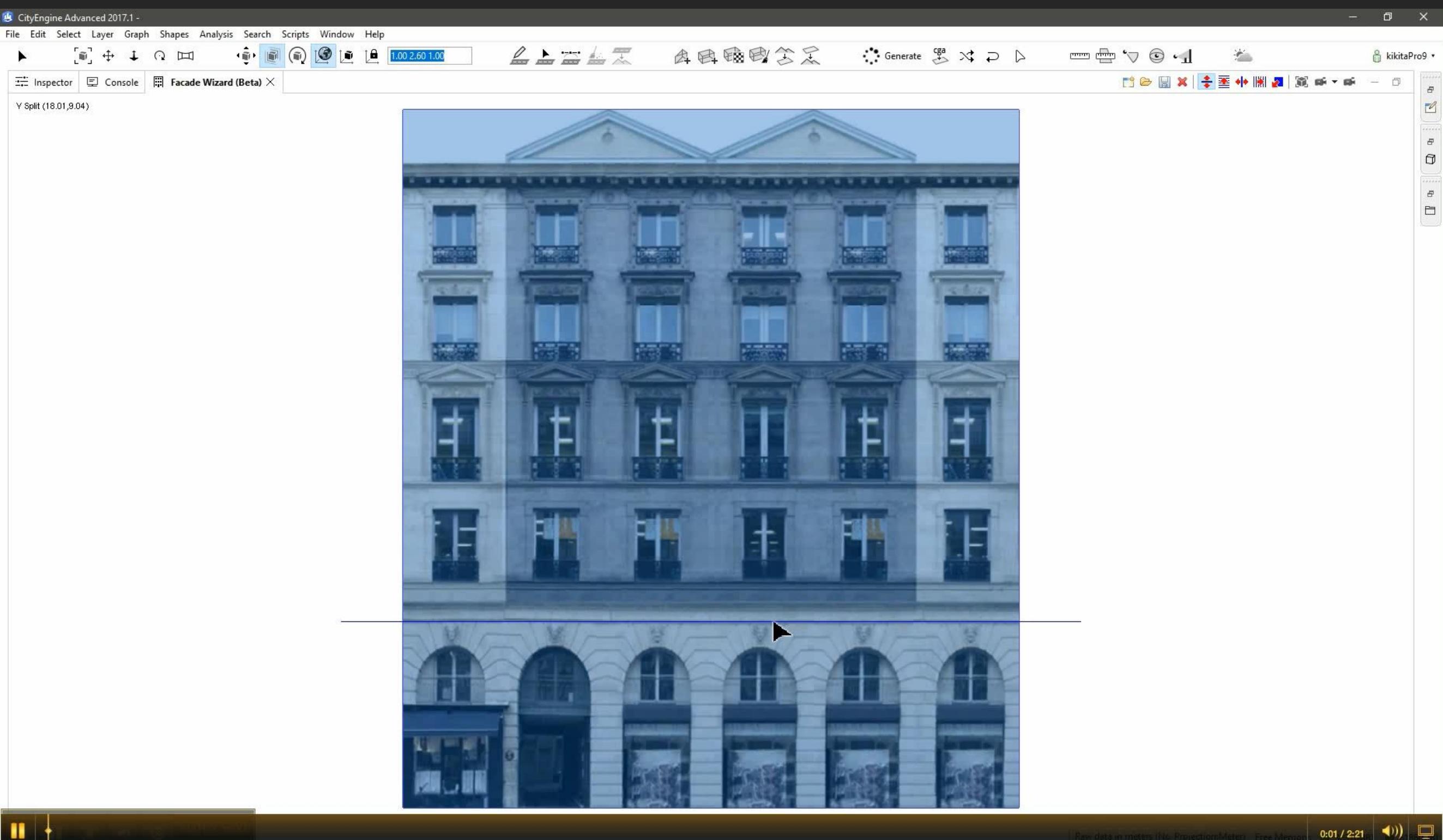




Demo

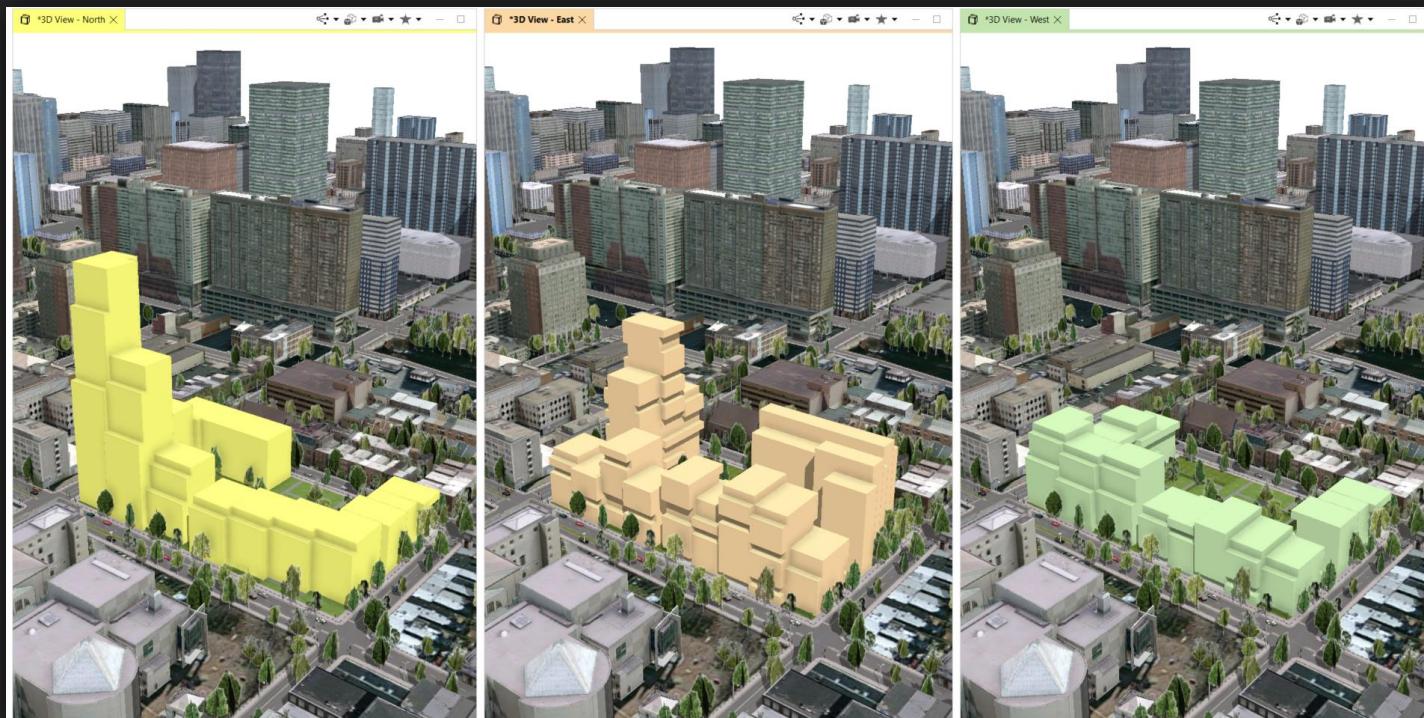
- 使用智能立面工具，高效建模

# Façade Wizard



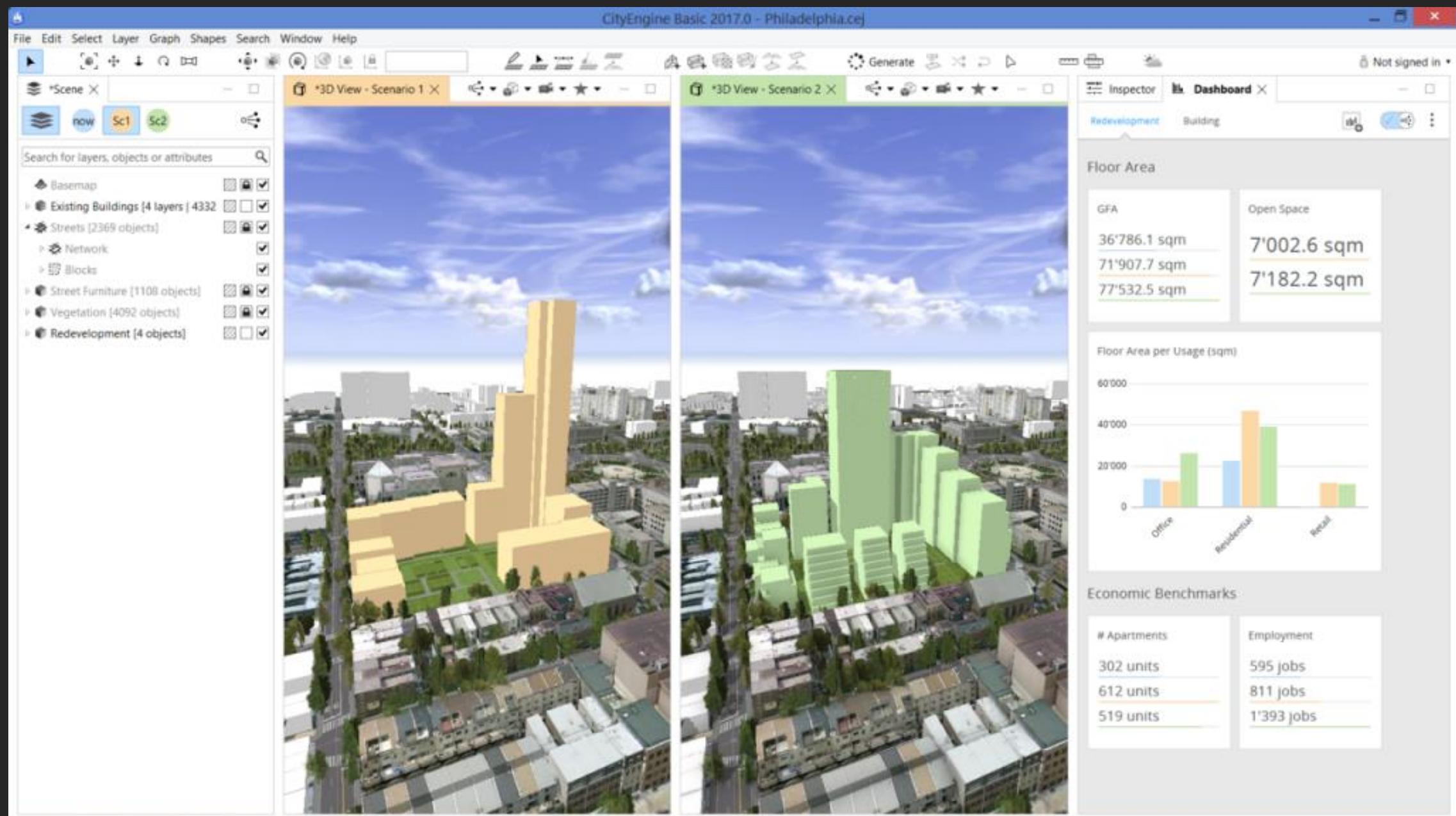
# 规划设计方案对比

- 多Scenario支持



# 规划设计方案对比

- 结合仪表盘更清晰的解读设计方案





Demo

## ■ 城市规划设计应用示例



# 使用python增强3D建模



# Python API

- Python 控制台
- Python 编辑器

The screenshot shows a Pydev Console window with the following Python code:

```
>>> import sys; from scripting import *; ce = CE(); print '%s\nJython %s [%s]\nCityEngine 2017.1.3674 1029R (c) 2008-2017 Esri R&D Center Zurich Inc.\nJython 2.7b2 (default:a5bc0032cf79+, Apr 22 2014, 21:20:17)\n[Java HotSpot(TM) 64-Bit Server VM (Oracle Corporation)] [javavm.7.0_80]\n>>> ce.setSelection(ce.getObjectsFrom(ce.scene, ce.withName("Broadway")))\n>>> ce.se
```

A code completion dropdown menu is open, listing various methods like `selectContinuousGraphObjects`, `selection`, etc.

The screenshot shows a Pydev Editor window with a script named `list_esri_plants_lib_bymxy.py`. The script contains the following code:

```
'''Created on 2018-5-3

@author: kikita
'''

from scripting import *

# get a CityEngine instance
ce = CE()

@def writeCGAlib():
    # open the cga file to be written
    cga = /*Asset Library Loader : Generated by asset lib.py*/\n ver
    # write
    cga += '\n'
    cga += '# write'
    cga += '\n'
    cga += '# get all .jpg files from asset directory, and call their loader'
    for jpg in ce.getObjectsFrom("/", ce.isFile, ce.withName("/ESRI.1"))
        cga += "\n\tt(2,0,0) Texture(\""+jpg+"\")"

    # write
    cga += '\n'

# get all .jpg files from asset directory, and call their loader
for jpg in ce.getObjectsFrom("/", ce.isFile, ce.withName("/ESRI.1"))
    cga += "\n\tt(2,0,0) Texture(\""+jpg+"\")"
```

A modal dialog titled "Create a new Python module" is open, showing the following settings:

- Source Folder: /DemoforDevSumChina2018/scripts
- Package: (empty)
- Name: my\_extension
- Template: Module: Main (selected)

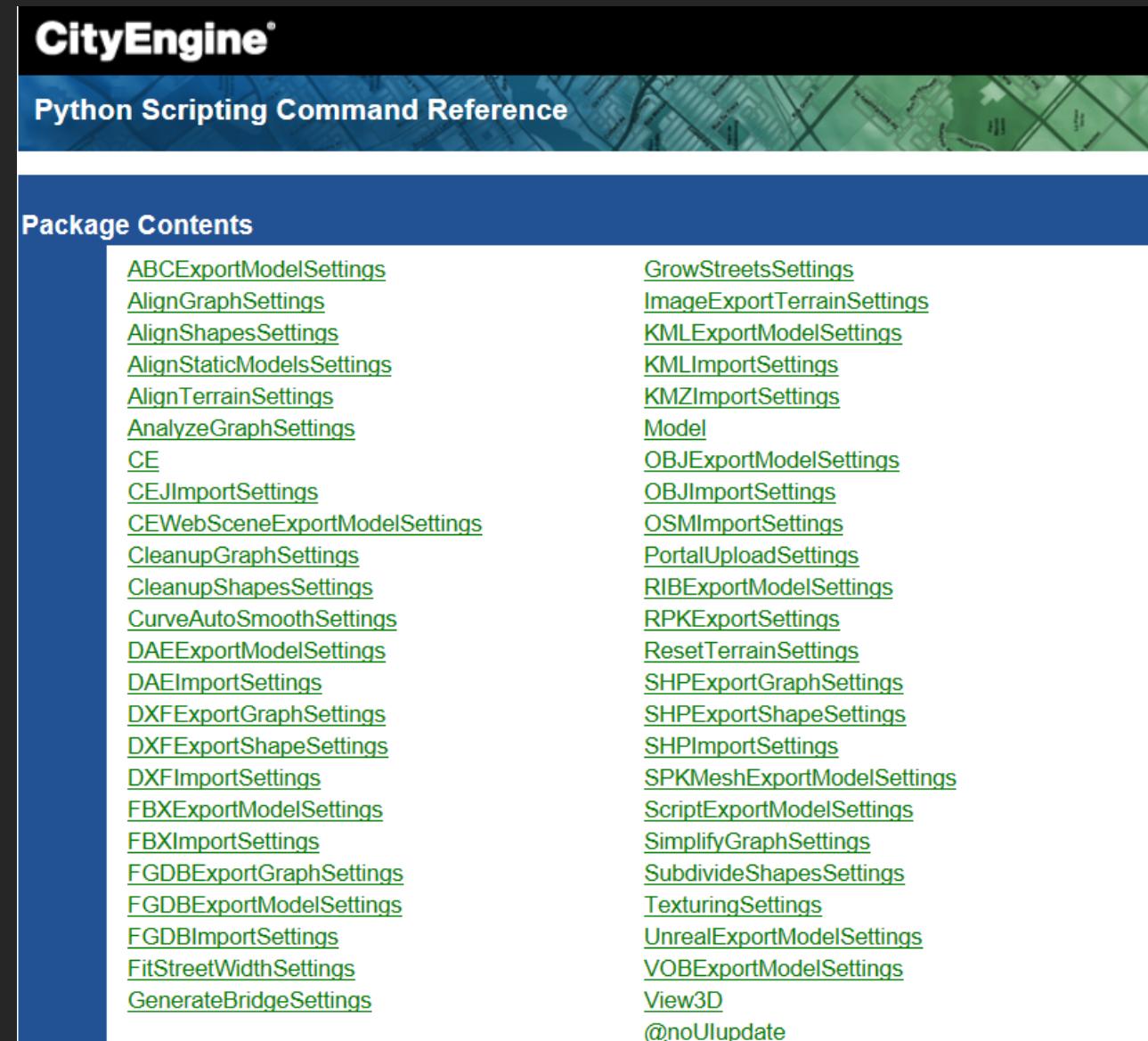
Buttons at the bottom of the dialog are "Finish" and "Cancel".

# Python Commands

- Scene Objects
- Attributes
- Map/Terrain Layer Operations
- Static Model Operations
- Shape Operations
- Graph/Street Operations
- Visibility Tools
- GUI
- Import
- Export
- File
- System

更多：

[CityEngine Help > Python Scripting Reference](#)



The screenshot shows the 'CityEngine' Python Scripting Command Reference interface. The top bar has the 'CityEngine' logo and the title 'Python Scripting Command Reference'. Below the title is a map thumbnail. The main area is titled 'Package Contents' and lists numerous command settings, each with a green link:

- ABCExportModelSettings
- AlignGraphSettings
- AlignShapesSettings
- AlignStaticModelsSettings
- AlignTerrainSettings
- AnalyzeGraphSettings
- CE
- CEJImportSettings
- CEWebSceneExportModelSettings
- CleanupGraphSettings
- CleanupShapesSettings
- CurveAutoSmoothSettings
- DAEExportModelSettings
- DAEImportSettings
- DXFExportGraphSettings
- DXFExportShapeSettings
- DXFImportSettings
- FBXExportModelSettings
- FBXImportSettings
- FGDBExportGraphSettings
- FGDBExportModelSettings
- FGDBImportSettings
- FitStreetWidthSettings
- GenerateBridgeSettings
- GrowStreetsSettings
- ImageExportTerrainSettings
- KMLExportModelSettings
- KMLImportSettings
- KMZImportSettings
- Model
- OBJExportModelSettings
- OBJImportSettings
- OSMImportSettings
- PortalUploadSettings
- RIBExportModelSettings
- RPKExportSettings
- ResetTerrainSettings
- SHPEExportGraphSettings
- SHPEExportShapeSettings
- SHPImportSettings
- SPKMMeshExportModelSettings
- ScriptExportModelSettings
- SimplifyGraphSettings
- SubdivideShapesSettings
- TexturingSettings
- UnrealExportModelSettings
- VOBExportModelSettings
- View3D
- @noUlupdate

# 扩展自定义功能

- 自动加载自定义模块



scripting.py

```
import sys
sys.path.append({PATH_TO_YOUR_SCRIPTS_DIRECTORY})
# e.g. sys.path.append("C:\user\CityEngine\MyProject\scripts")

import my_extension
```



Demo

■ 通过python扩展功能示例

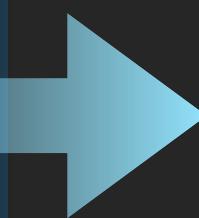


# 3D内容导出与分享

# 规则包/RPK

- Rule Package
- CGA代码与资源的压缩包

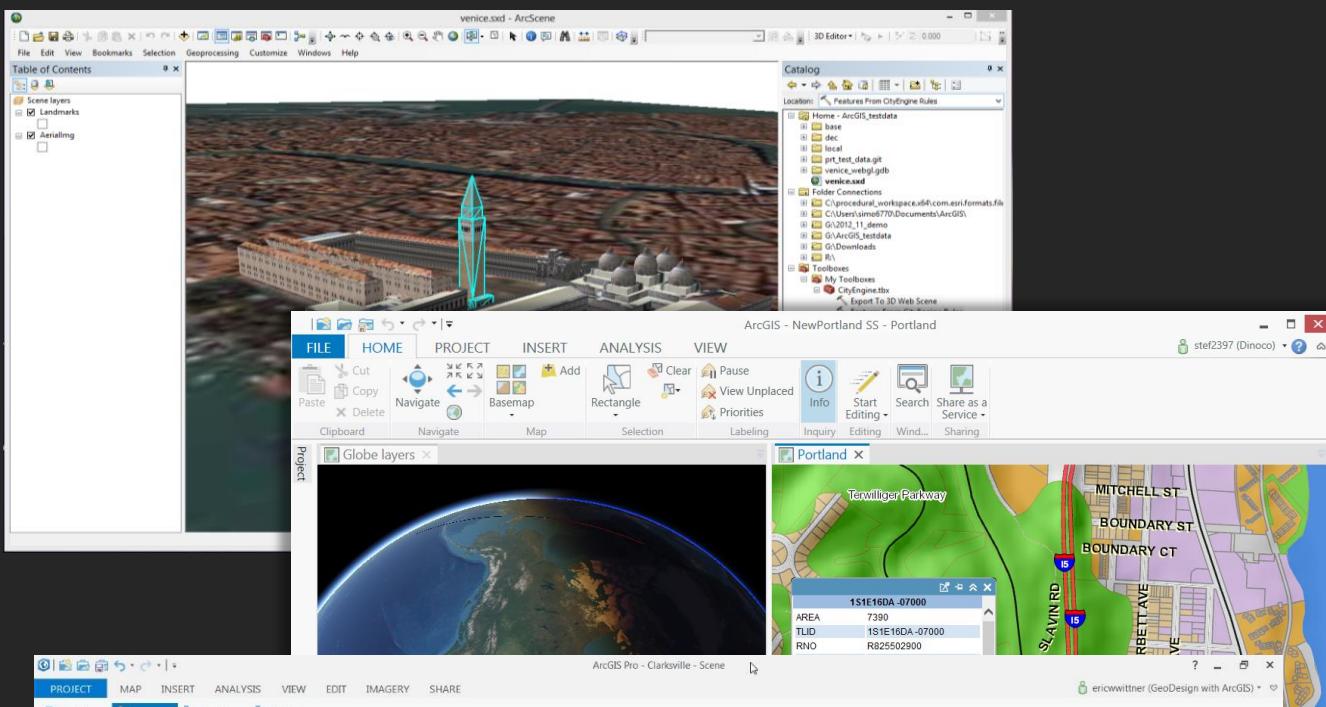
\*.rpk



# 使用规则包

## 用途

- 从2D数据快速创建3D场景
- ArcGIS Pro高级3D程序符号

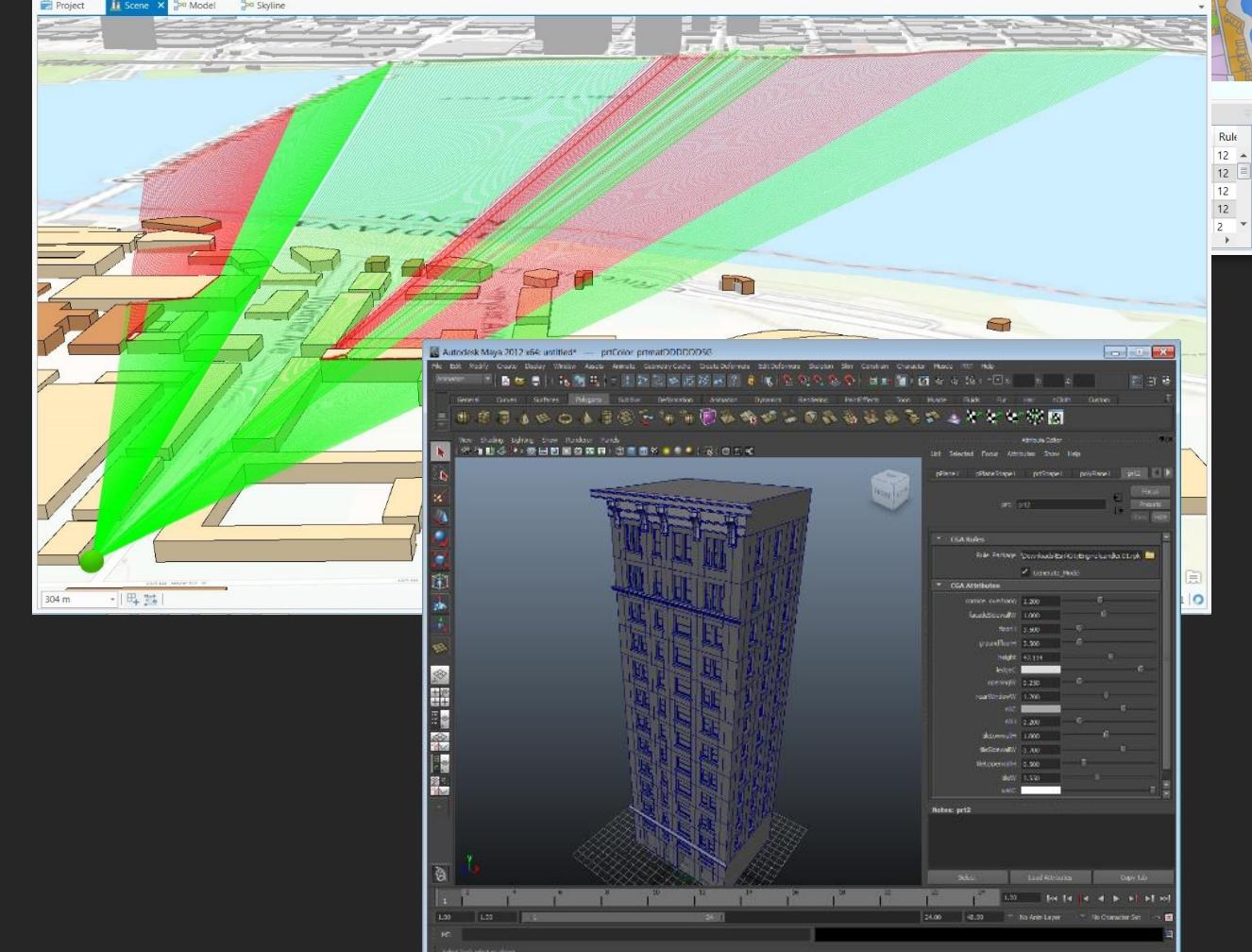


## 分享至组织

- AGOL / ArcGIS Enterprise

## 使用rpk的应用程序：

- ArcGIS Desktop
- ArcGIS Pro
- 3rd party 3D apps
- CityEngine SDK

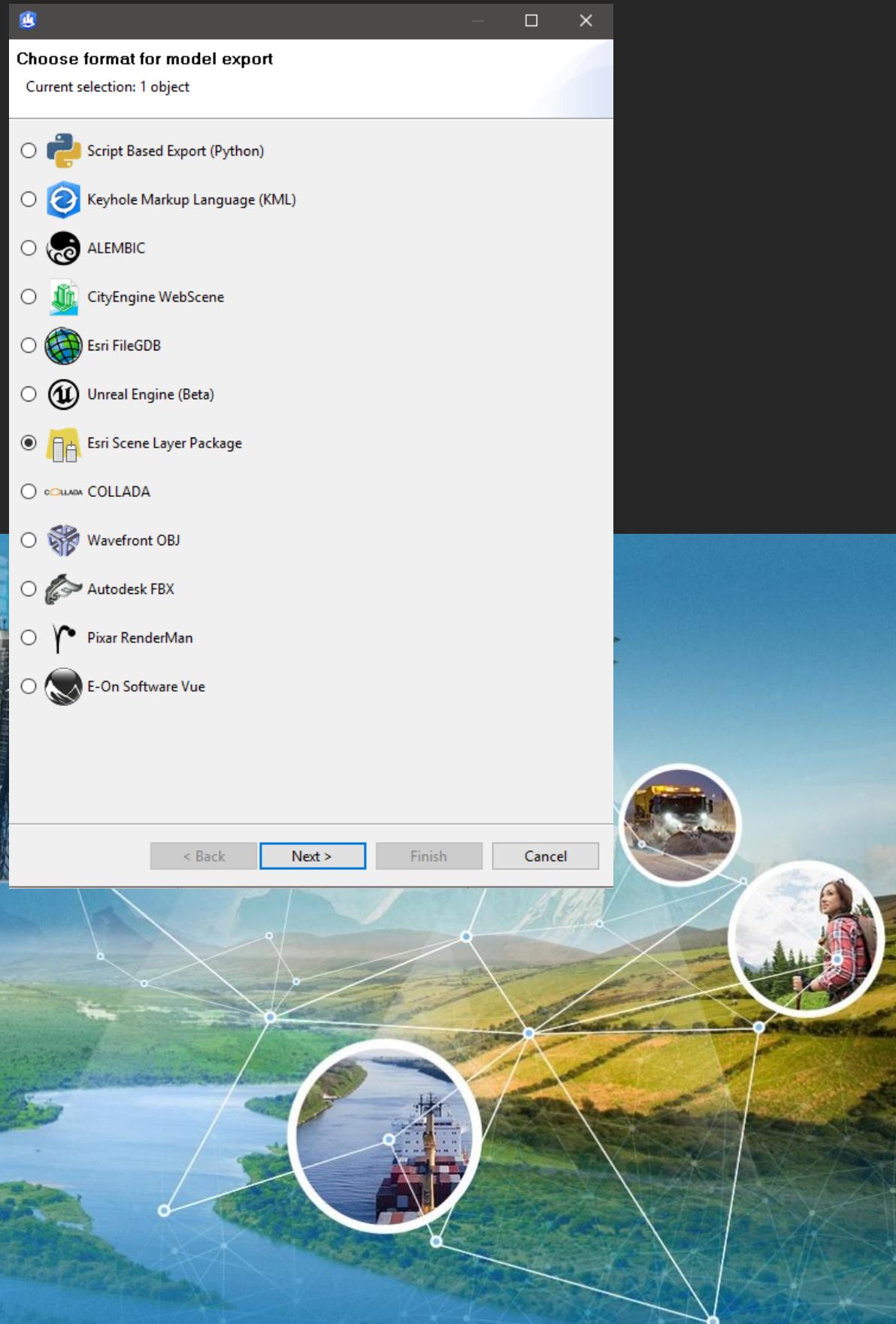


# 场景图层包/SLPK

- Esri Scene Layer Package

分享至组织并发布 web scene

- AGOL / ArcGIS Enterprise



# 即将到来

- CityEngine 2018 Beta
- CityEngine 2018 中文版
- VR/AR 增强
- Mobile Scene Layer
- GIS BIM 深度结合



# 谢谢！



CityEngine 30天试用版

<http://www.esri.com/software/cityengine/free-trial>

CityEngine 学习资源

<http://desktop.arcgis.com/en/cityengine/>

ArcGIS 知乎

<http://zhihu.esrichina.com.cn>



Email: [muxy@esrichina.com.cn](mailto:muxy@esrichina.com.cn)



Blog: <http://kikitamap.com/>



Github: <https://github.com/kikitaMoon>



精彩继续，下一场

# 三维服务发布与应用实战

谢军

# **THE SCIENCE OF WHERE**