



Esri International Developer Summit

Palm Springs, CA

# Creating and sharing Rule Packages with CityEngine

Markus Lipp, Nathan Shephard

Esri R&D Center Zurich

# Agenda

- **Why RPK's, what can they do? (5 min)**
- **Overview of Procedural Modelling (5 min)**
- **Introduction CityEngine (5 min)**
- **CGA 101 (25 min)**
- **Creating Rule Packages (5 min)**
- **Important Tips for RPKs (5 min)**
- **Q&A**

# Using RPK's

- **Content creation**
  - Convert simple data into useful 3D content
  
- **Procedural Symbol**
  - Override current symbology for the task at hand

**DEMO**

# Using Rule Packages – Summary

- **Share them either a file or as item online**
  - **Must be available locally to be consumed**
- **Consume in:**
  - **ArcGIS 10.2**
    - **“Features from CityEngine Rules” GP tool**
      - **Requires schema-matching for parameters to be applied**
  - **ArcGIS Pro**
    - **Add as a procedural symbol to either polygon or multipatch data**
      - **Can manually configure the schema within Pro**
  - **3<sup>rd</sup> party 3D apps**
    - **Using the CityEngine SDK (not demonstrated)**

# Important Tips

- **Assets**

- All will be included → beware the 'random' use of textures and models
- Has a huge impact on file size → consider multiple / focused RPKs

- **Using Tags / Annotations**

- Define input geometry: @InPoint, @InLine, @InPolygon, @InPointCloud, @InMesh

- **Parameters** (ie: the 'attr' definition)

- Use it as much as needed... but no more
- Use a good name → shown on the UI
- Match to data model → auto-connect matching names
- Set Ranges: @Range (v1, v2)
- Provide 'type' information → @Distance, @Percent, @Angle

- **Protect the CGA** (as needed)

- If your rule is something you sell, or has IP, then protect it

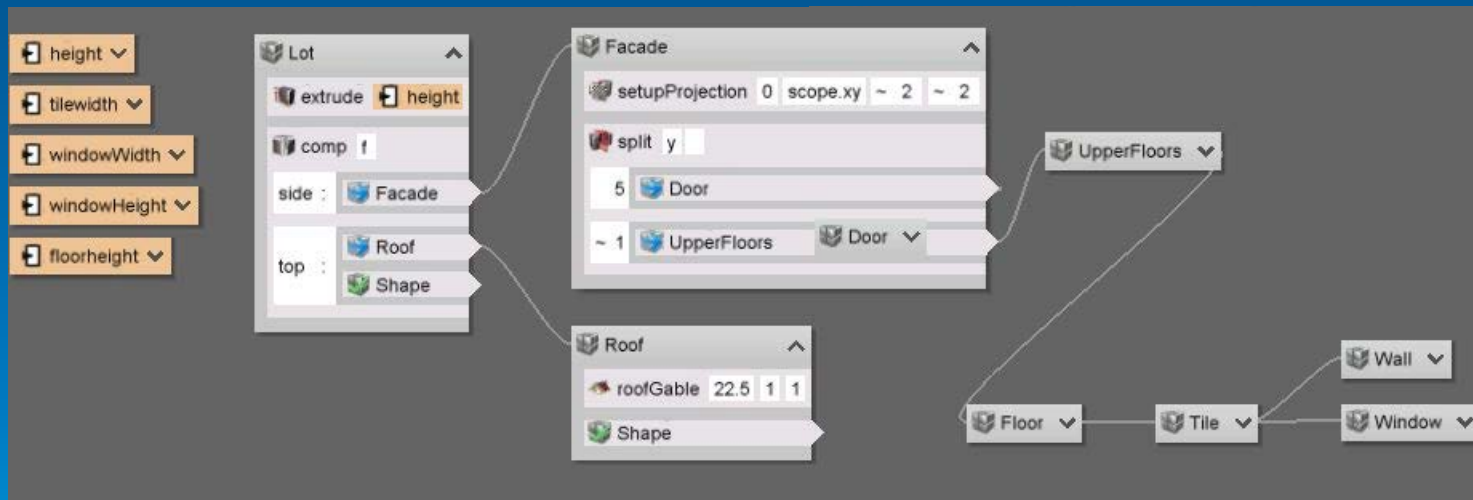
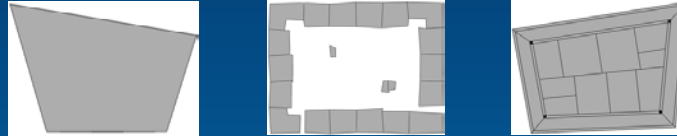
```
attr floorCount = 3
@Distance
attr roofHeightVal = 2
@Percent @Range(0,80)
attr percentageStripe = 5

@StartRule @InMesh
BuildingShell -->
```

# Procedural Modeling

*3D model creation using rules / algorithms*

- Base geometry



*Base geometry*

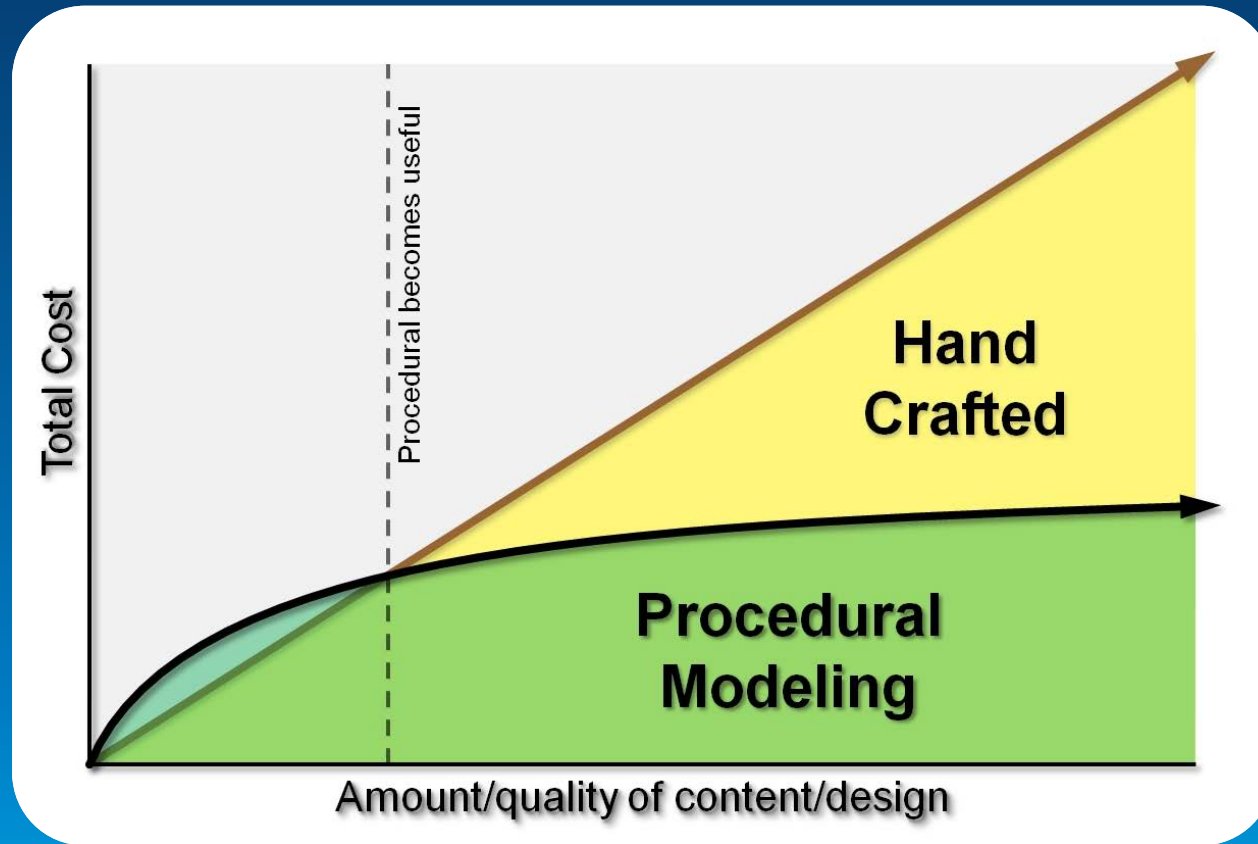


*Final 3D model*

*Iterative refinement*

**iteratively refine a design by creating more and more detail**

# Procedural modeling vs. Manual modeling

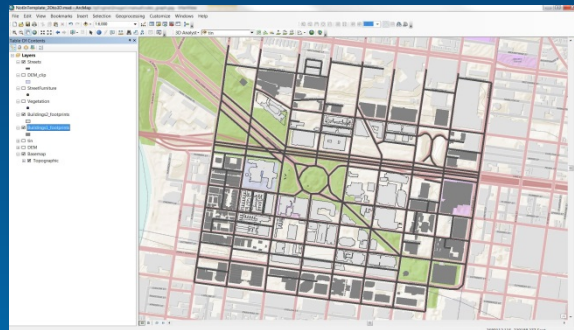


Time reduction / cost saving



# 3D City Content Creation

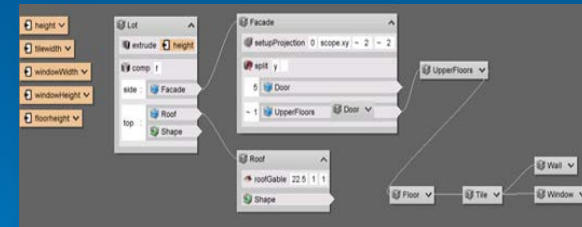
## Procedural city modeling



Geometry

OBJECTID*	Z Min	Z Max	Roof Form	Ridge Jn	Eave Jn	Z	RMg ID	SHAPE*	SHAPE Length	SHAPE Area
1	20.2522	64.412	fat	8.4	9.25	21.814448	25	Polygon	99.688535	618.683569
2	21.2027	65.6368	shed	9.85	8.65	22.639911	26	Polygon	58.555297	158.659564
3	36.9982	326.6588	fat	88.388716	88.388716	37.212535	120	Polygon	406.16474	10202.325473
4	22.3276	67.7987	shed	9.4	7.91	24.884243	34	Polygon	85.637074	376.840012
5	23.8841	66.8352	shed	12.83	12.23	28.071111	30	Polygon	23.89816	32.156816
6	26.6666	63.536	fat	3.73	3.73	28.928017	33	Polygon	63.137759	135.929382
7	19.7574	188.1827	fat	42	42	22.816426	111	Polygon	789.48338	27252.821564
8	13.8602	65.5987	buttefly	16.7	14.3	13.723567	9	Polygon	175.636887	689.287114
9	17.2831	68.762	fat	33	33	30.992004	113	Polygon	133.410688	968.164113
10	17.2831	68.762	fat	195	195	27.135445	113	Polygon	373.965933	6962.900716
11	13.8602	65.5987	fat	11.04	10.29	17.432316	9	Polygon	148.819964	1384.438745
12	8.3999	305.2586	fat	91.78	96.952511	42.109004	141	Polygon	931.426243	151219.733661
13	22.3276	67.7987	gable	10.58	8	25.656544	34	Polygon	93.104492	491.156073
14	40.6078	69.0667	fat	8.769382	8.769382	40.399333	118	Polygon	48.6991823	146.674535
15	29.0621	164.4961	fat	23.337251	23.337251	41.944603	148	Polygon	1098.37786	63206.909683
16	22.3276	67.7987	gable	6.71	5.82	24.665642	34	Polygon	109.817427	744.40786
17	35.9163	69.258	fat	18.185153	17.865688	48.728269	118	Polygon	48.728269	146.801611
18	20.2522	64.412	fat	11.84	10.26	21.588778	25	Polygon	150.421816	1296.800847

Attributes



Rule / Rule Package



# 3D City Content Creation



• Rule based 3D cities

# 3D City Design

## Parametric editing



Add a floor



Add a roof

## Dynamic editing

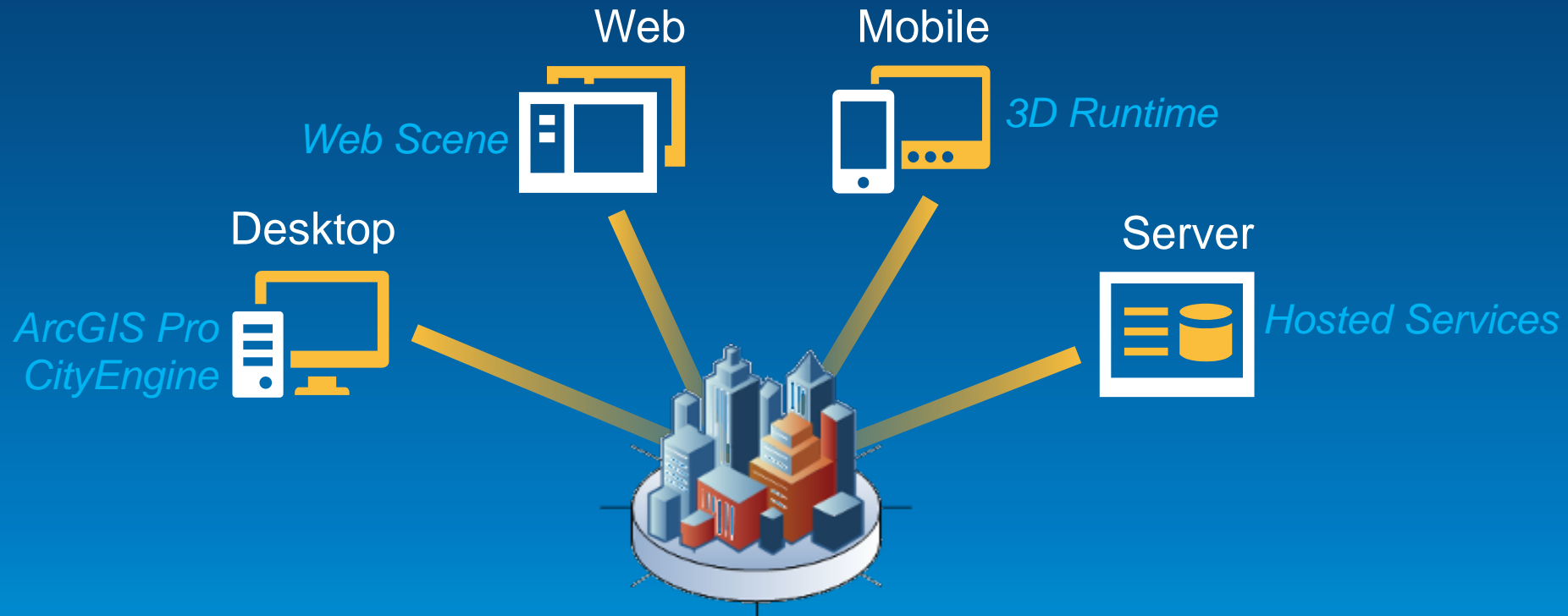


## Procedural reporting



Rule based design

# 3D Across the Platform



# CGA and Rule Packages



# CGA = Computer Generated Architecture

- Computer Generated Architecture
- Shape Grammar
- Programming language for 3D models of buildings

```
Envelope -->
  case scope.sx*scope.sz > 6000:
    15%: split(y) { '0.9 : Tower
                  | ~3   : s('0.8, '1,
                  | ~1   : s('0.6, '1,
    15%: split(y) { 'rand68 : Tower
                  | ~rand(0.5, 1.5) : Tower
                  | ~1 : Tower
    15%: split(y) { 'rand48 : Tower
                  | ~1 : s('ran
    15%: split(y) { 'rand48 : Tower
                  | ~1 : t('ran
  else: Tower
  else: Tower
```



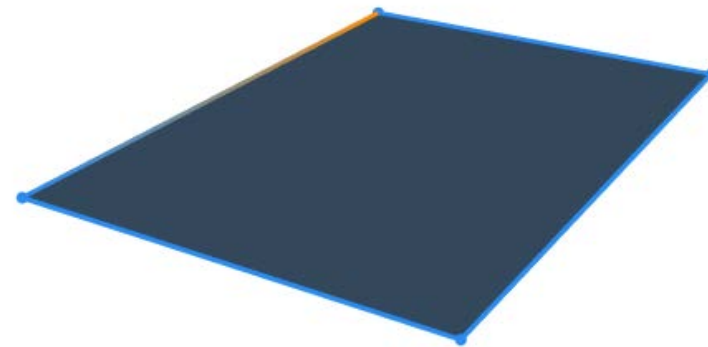
# Basic Concepts: Rules & Shape Operations

- Rules

- Based on shape operations

```
Init-->  
  extrude(10)  
  comp(f) {  
    side : Facade. |  
    top  : Roof  
  }
```

```
Roof-->  
  roofHip(30)
```

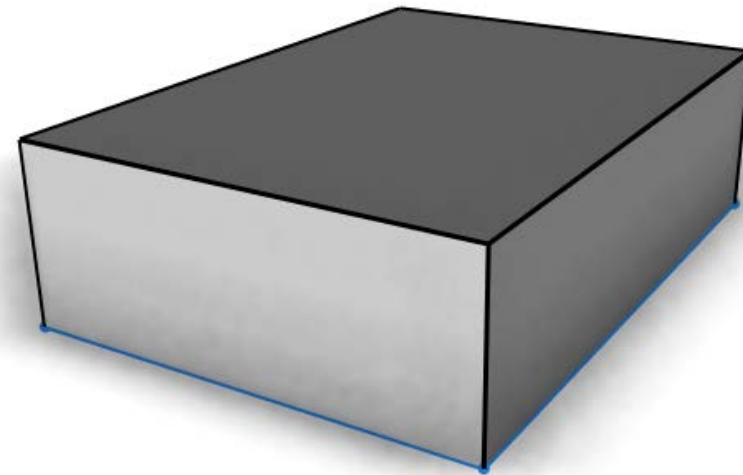


# Basic Concepts: Rules & Shape Operations

- Rules

- Based on shape operations

```
Init-->  
  extrude(10)  
  comp(f) {  
    side : Facade. |  
    top  : Roof  
  }  
  
Roof-->  
  roofHip(30)
```





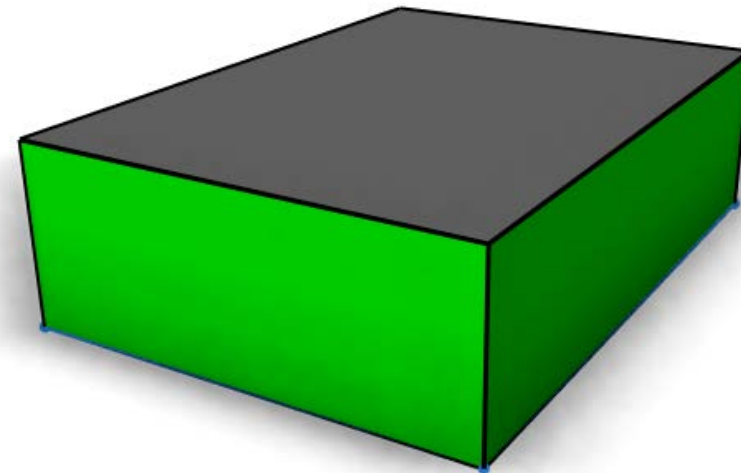
# Basic Concepts: Rules & Shape Operations

- Rules

- Based on shape operations

```
Init-->  
  extrude(10)  
  comp(f) {  
    side : Facade.  
    top  : Roof  
  }
```

```
Roof-->  
  roofHip(30)
```



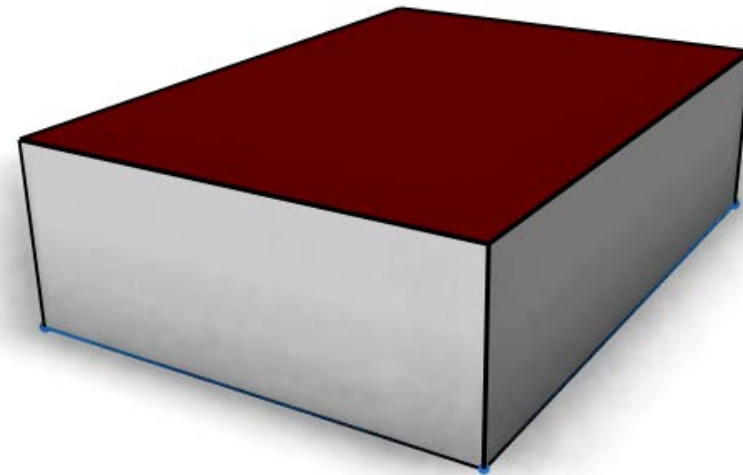
# Basic Concepts: Rules & Shape Operations

- Rules

- Based on shape operations

```
Init-->
  extrude(10)
  comp(f) {
    side : Facade. |
    top  : Roof
  }

Roof-->
  roofHip(30)
```



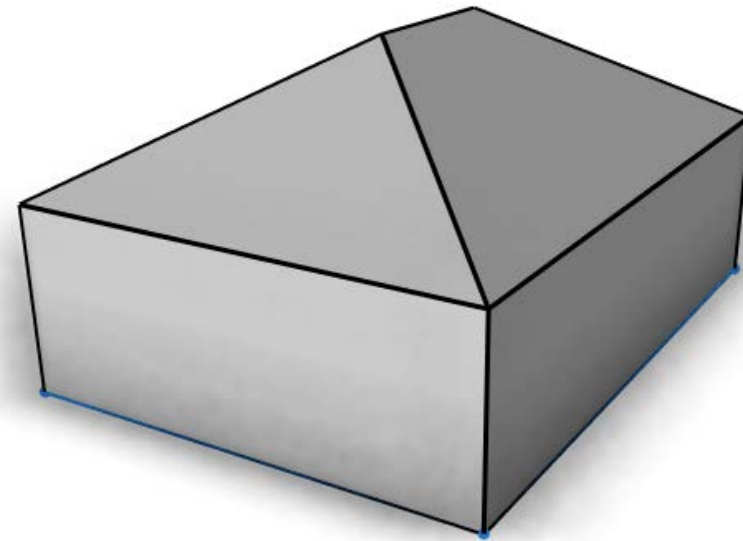
# Basic Concepts: Rules & Shape Operations

- Rules

- Based on shape operations

```
Init-->  
  extrude(10)  
  comp(f) {  
    side : Facade. |  
    top  : Roof  
  }
```

```
Roof-->  
  roofHip(30)
```



# Basic Concepts: Functions

- **Functions**

- Strongly typed
- Types “auto-deferred”

- **constant = simple function**

```
height = 15  
angle  = 35
```

```
Init-->  
  extrude(height)  
  comp(f) {  
    side : Facade. |  
    top  : Roof  
  }
```

```
Roof-->  
  roofHip(angle)
```

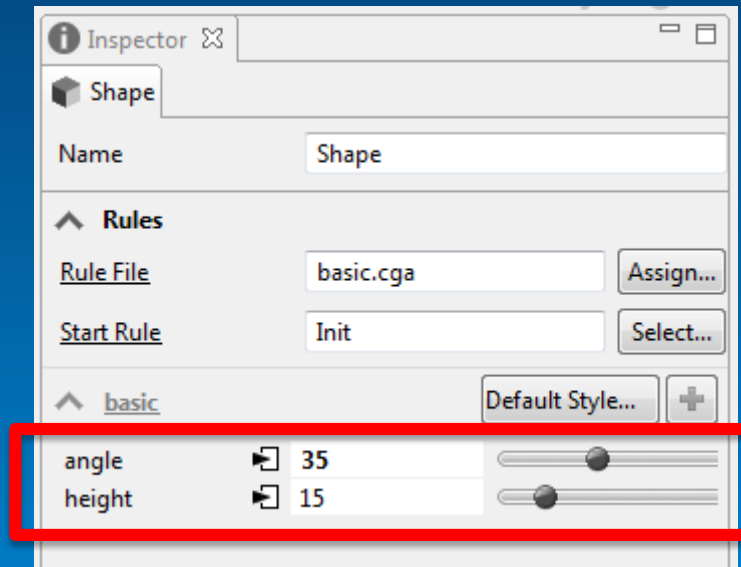
# Basic Concepts: Rule Attributes

- attr functions can be set from outside, e.g. GUI

```
attr height = 15
attr angle  = 35

Init-->
  extrude(height)
  comp(f) {
    side : Facade. |
    top  : Roof
  }

Roof-->
  roofHip(angle)
```



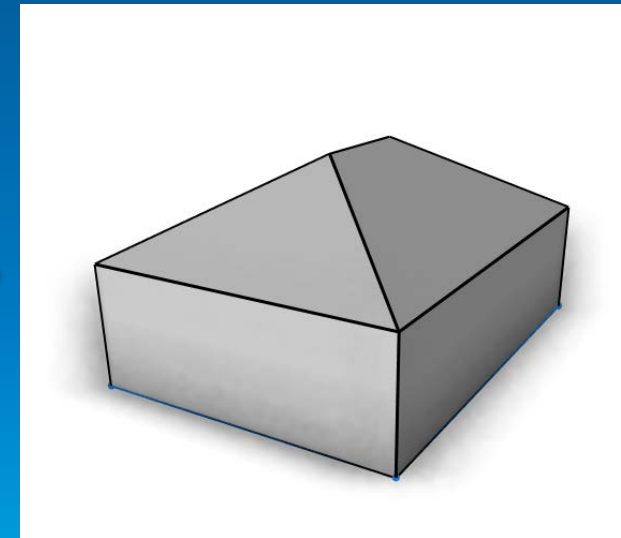
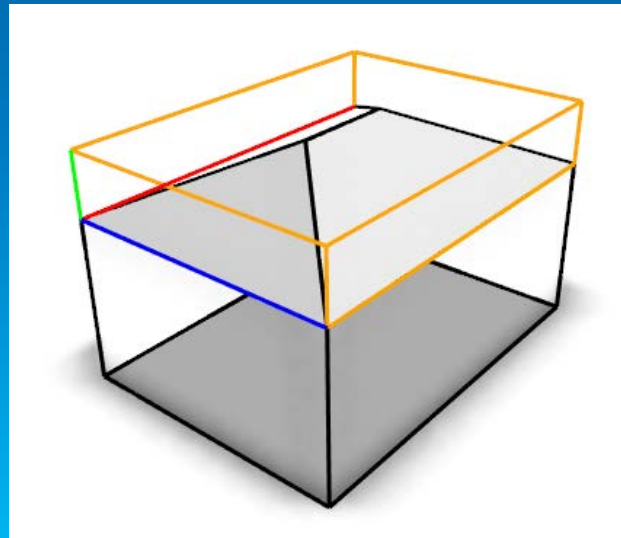
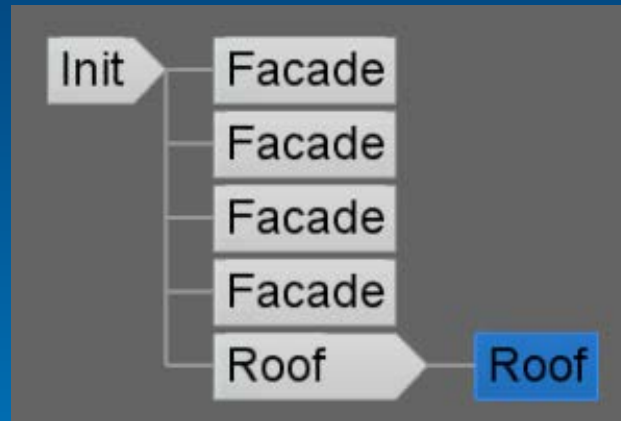
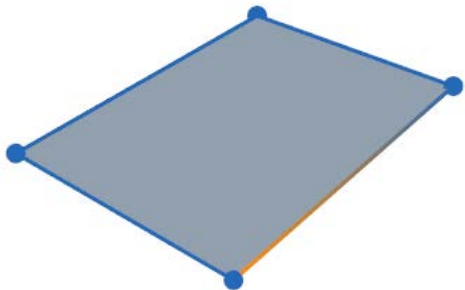
# Shapes & 3D Models

```
attr height = 15  
attr angle  = 35
```

```
Init-->  
  extrude(height)  
  comp(f) {  
    side : Facade |  
    top  : Roof  
  }
```

```
Roof-->  
  roofHip(angle)
```

angle   
height



# Conditional Rules

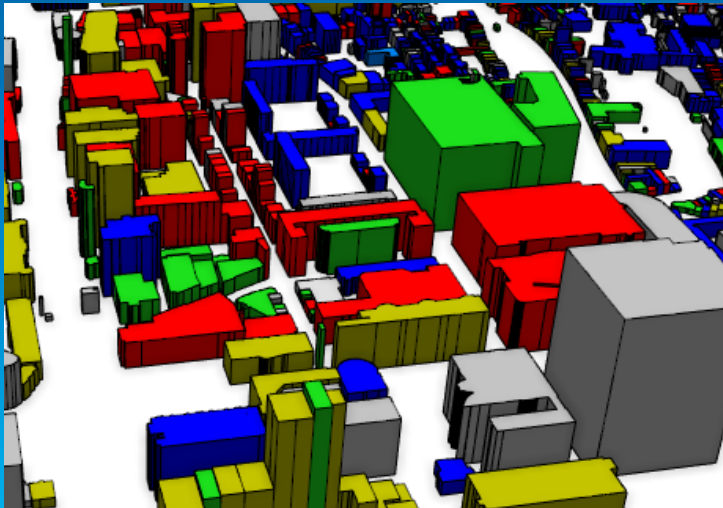
- case blocks in rules or functions

```
Roof-->  
    case geometry.area > 1000:  
        color(1,0,0)  
        X.  
    case geometry.area > 500:  
        color(1,0.5,0)  
        X.  
    else:  
        print(geometry.area)  
        X.
```

# Use Case 1: Thematic Visualization

- Use CGA to visualize your data in 3D

Object Attributes	
BuildingFID	Building/0599100000423926
OBJECTID	16652
Shape_Area	12826.692636
Shape_Length	534.682151
beginLifespan_date	30 Dec 0002 23:00:00 GMT
status	Completed
totalHeight	87.412849
usage	Public



```
attr usage = ""
attr totalHeight = 0

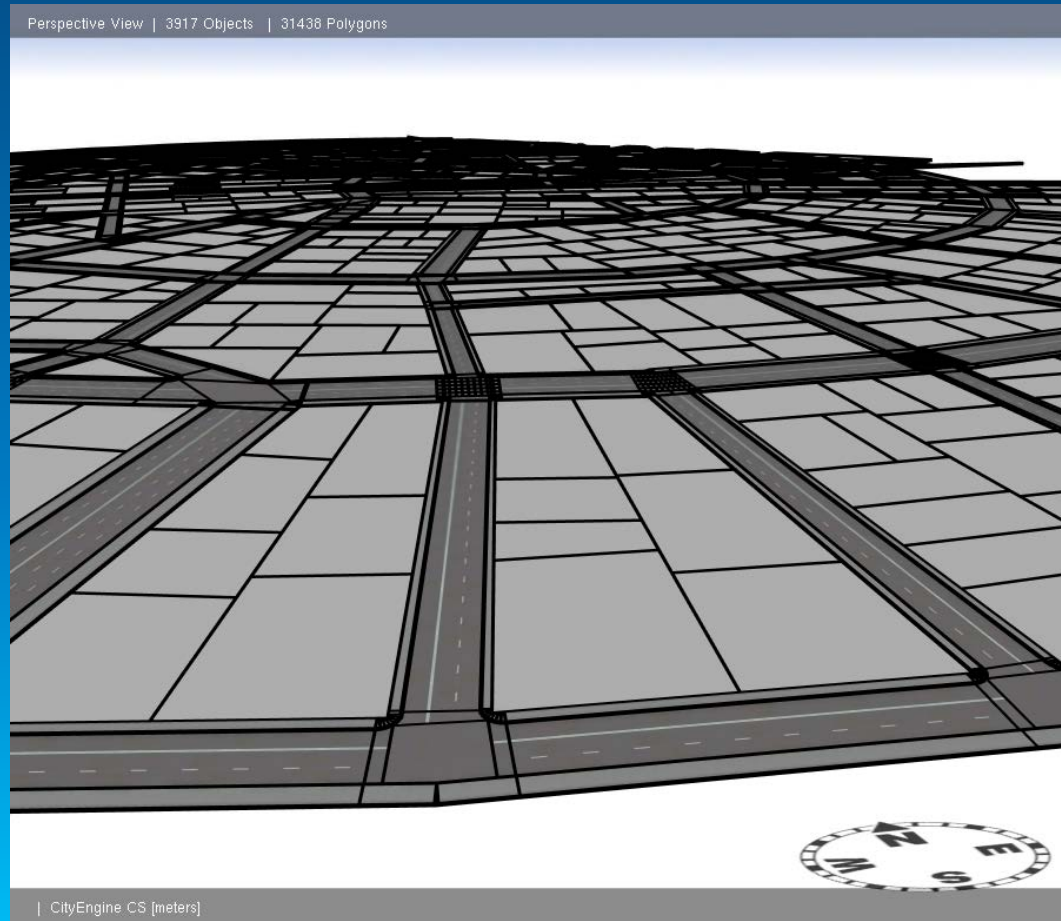
Init-->
  case usage == "Public":
    color(0.1,0.8,0.1)
    X
  case usage == "Educational":
    color(0,0.5,1)
    X
  ...

X-->
  extrude(totalHeight)
```



## Use Case 2: Procedural Modeling for Urban Design

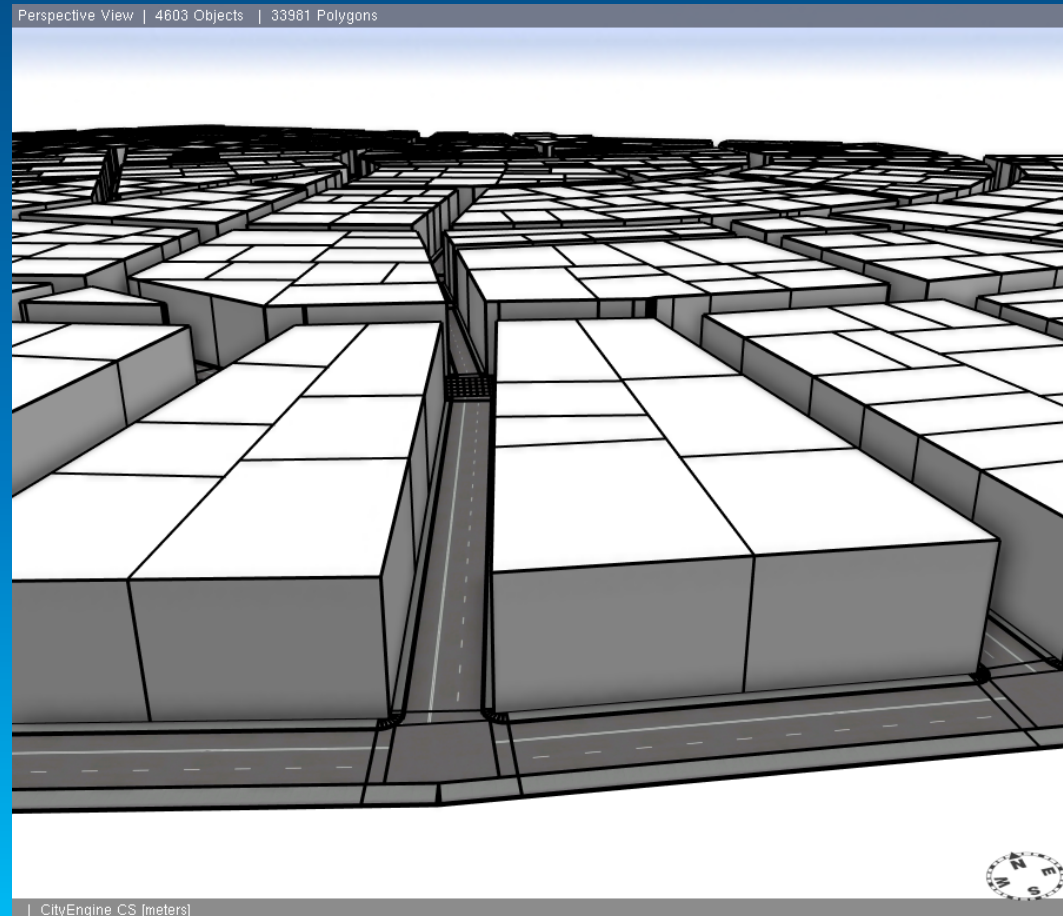
- Let's visualize a City based on those streets / parcels



# Use Case 2: Procedural Modeling for Urban Design

- Simple extrude

```
Lot-->  
  extrude(20)
```

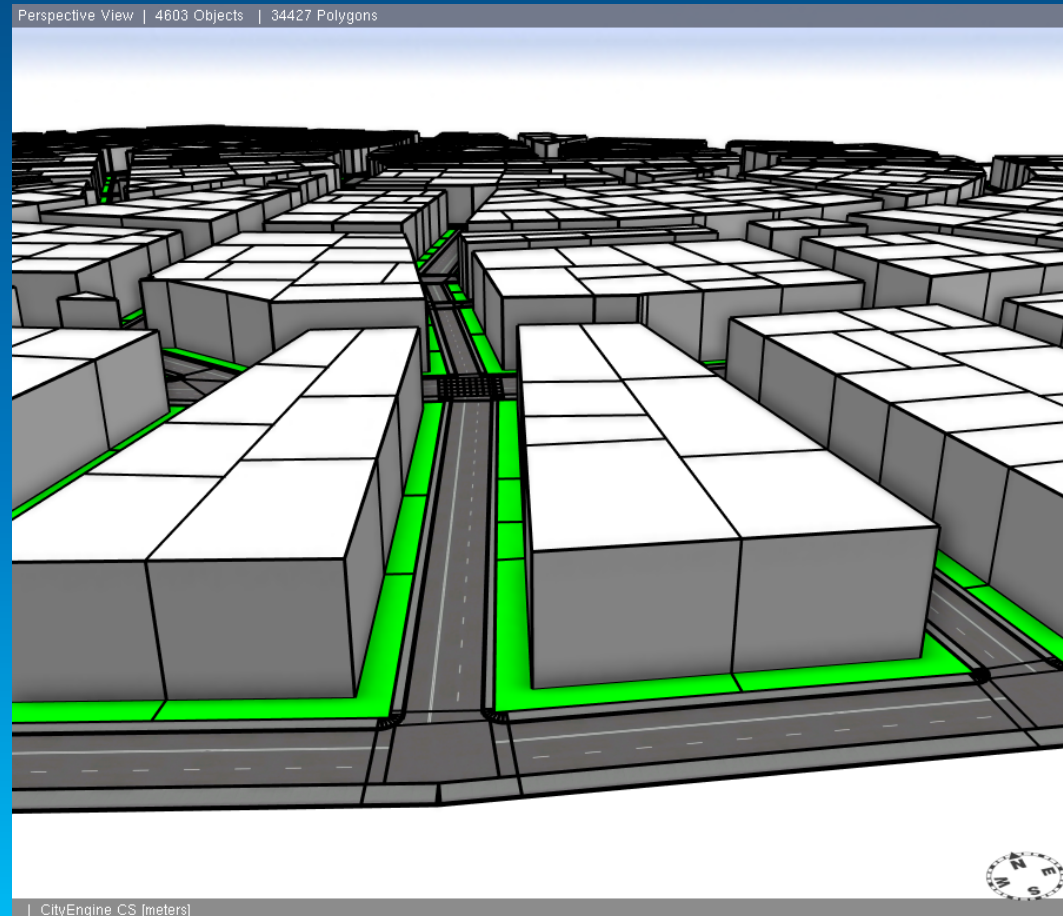


# Use Case 2: Procedural Modeling for Urban Design

- setback operation
- selectors based on neighbouring streets

```
Lot-->
```

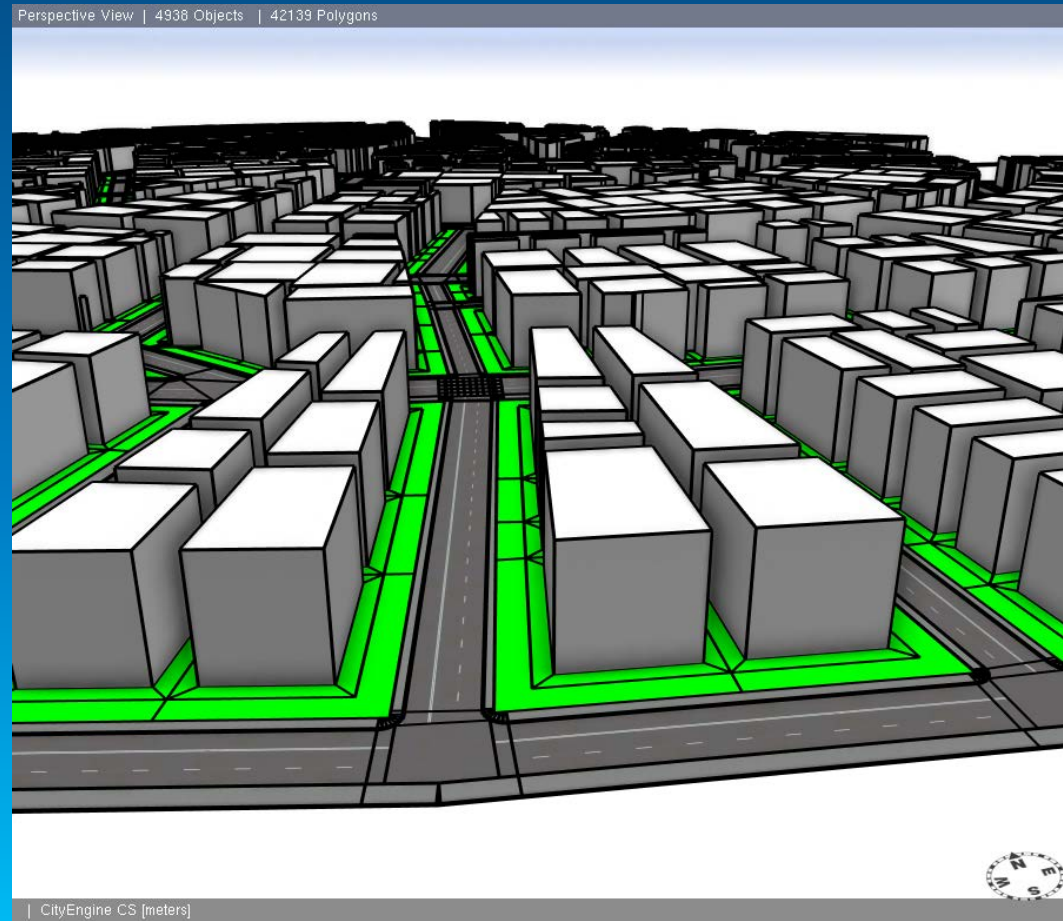
```
setback(5) {  
  street.front:  
    color(0,1,0)  
    X. |  
  remainder:  
    extrude(20)  
    X.  
}
```



# Use Case 2: Procedural Modeling for Urban Design

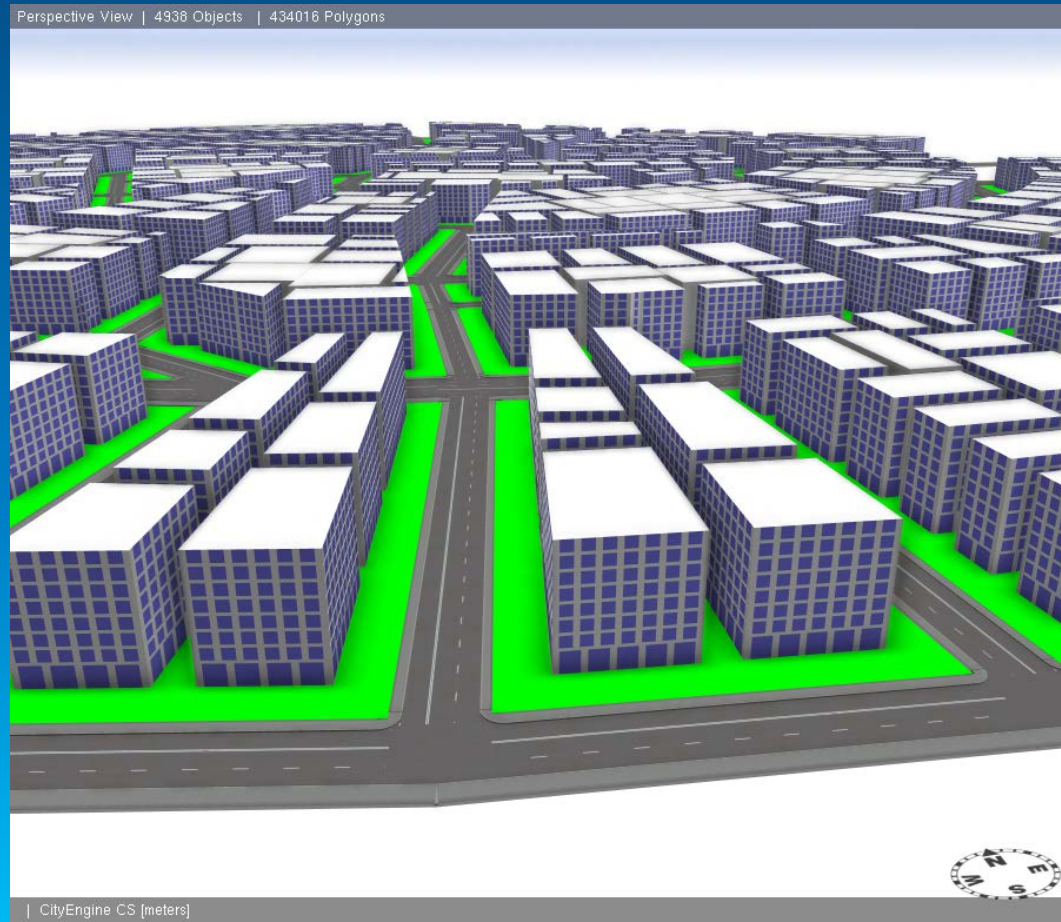
- offset operation to subdivide parcels

```
Lot-->
  setback(5) {
    street.front:
      color(0,1,0)
      X. |
    remainder:
      offset(-3)
      comp(f) {
        border:
          color(0,1,0)
          X. |
        inside:
          extrude(20)
          X.
      }
  }
}
```



## Use Case 2: Procedural Modeling for Urban Design

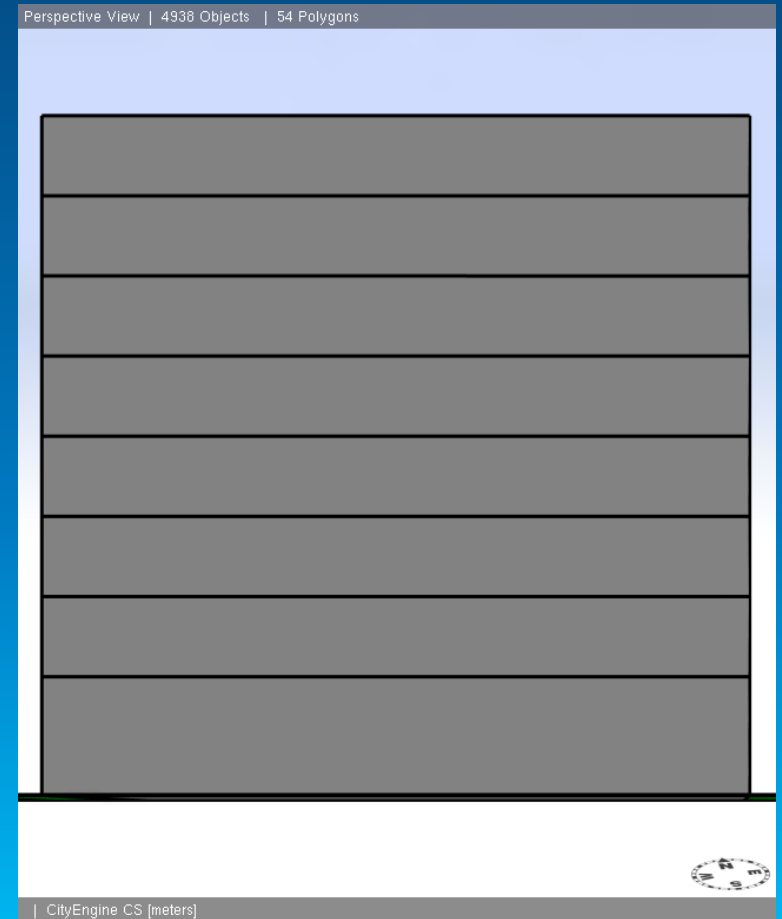
- split into floors and facades



# Use Case 2: Procedural Modeling for Urban Design

- split mass model into floors

```
Mass-->
  split(y) {
    3.5 : GroundFloor. |
    { ~2.5 : Floor. }*
  }
```



# Use Case 2: Procedural Modeling for Urban Design

- split floors into wall + windows

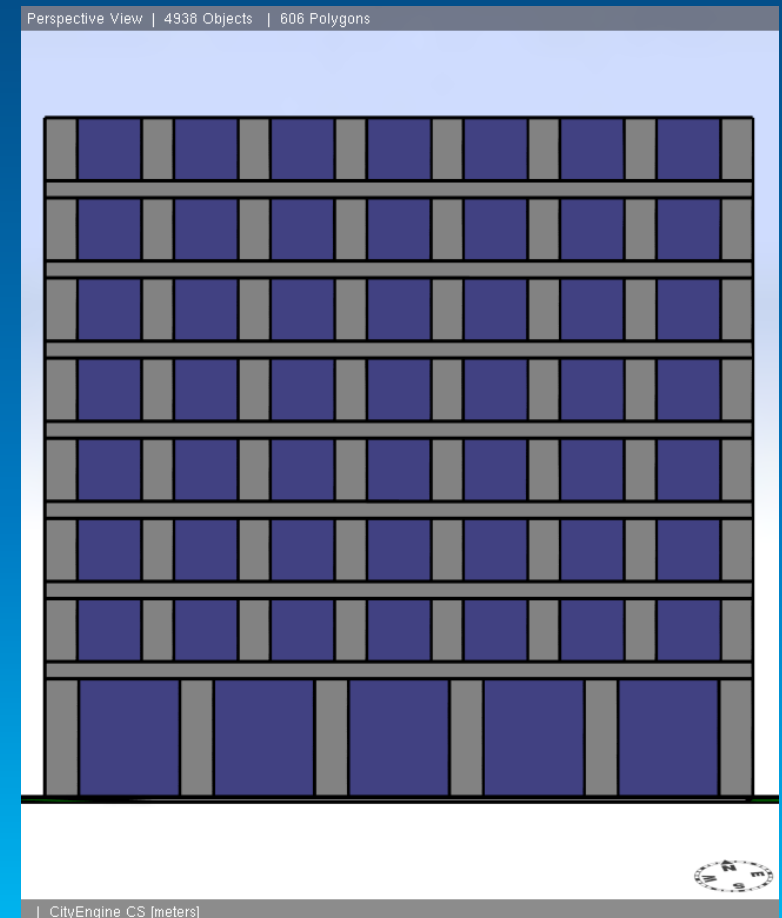
```
GroundFloor-->
  comp(f) { side : GFFacade }

GFFacade-->
  split(x) {
    { ~1 : Wall. | ~3 : Window }* | ~1 : Wall.
  }

Floor-->
  comp(f) { side : FloorFacade }

FloorFacade-->
  split(y) {
    0.5 : Wall. | ~1 : split(x) {
      {~1 : Wall. | ~2 : Window }* | ~1 : Wall. }
    }

Window-->
  color(0.5,0.5,1)
```

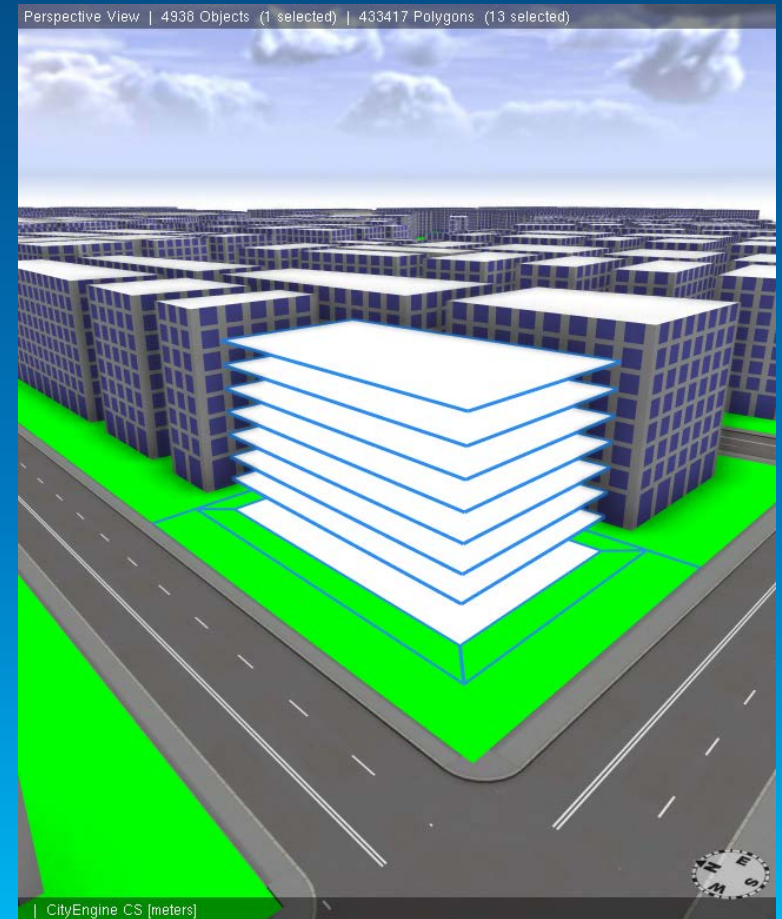


# Use Case 2: Procedural Modeling for Urban Design

- Use reporting for analysis, e.g. gross floor area

```
FloorGFA-->  
  comp(f) {  
    bottom:  
      report("GFA", geometry.area)  
  }
```

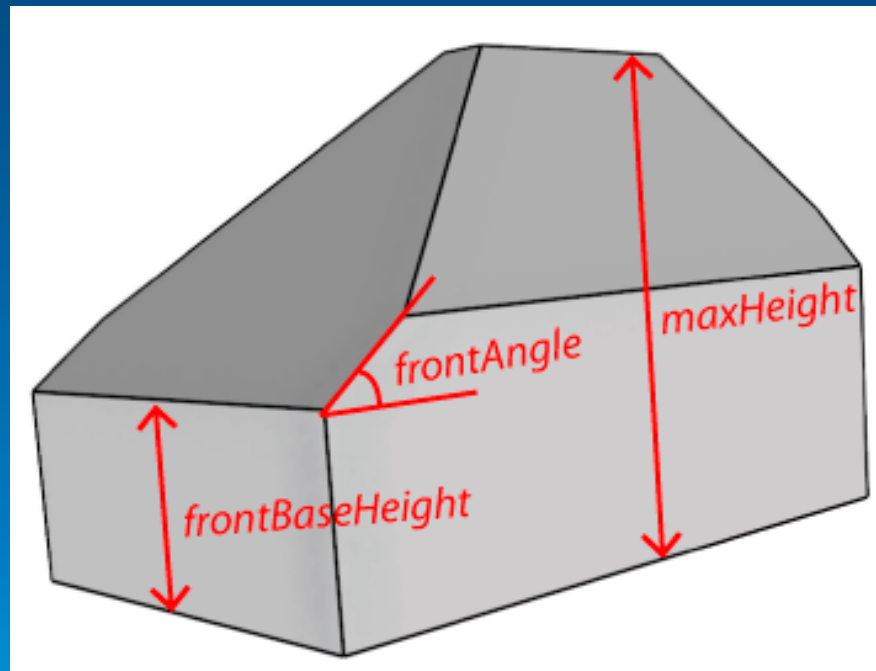
Report	N	%	Sum	%	Avg	Min	Max	NaNs
GFA	8	0.00	4997.25	0.00	624.66	624.66	624.66	0



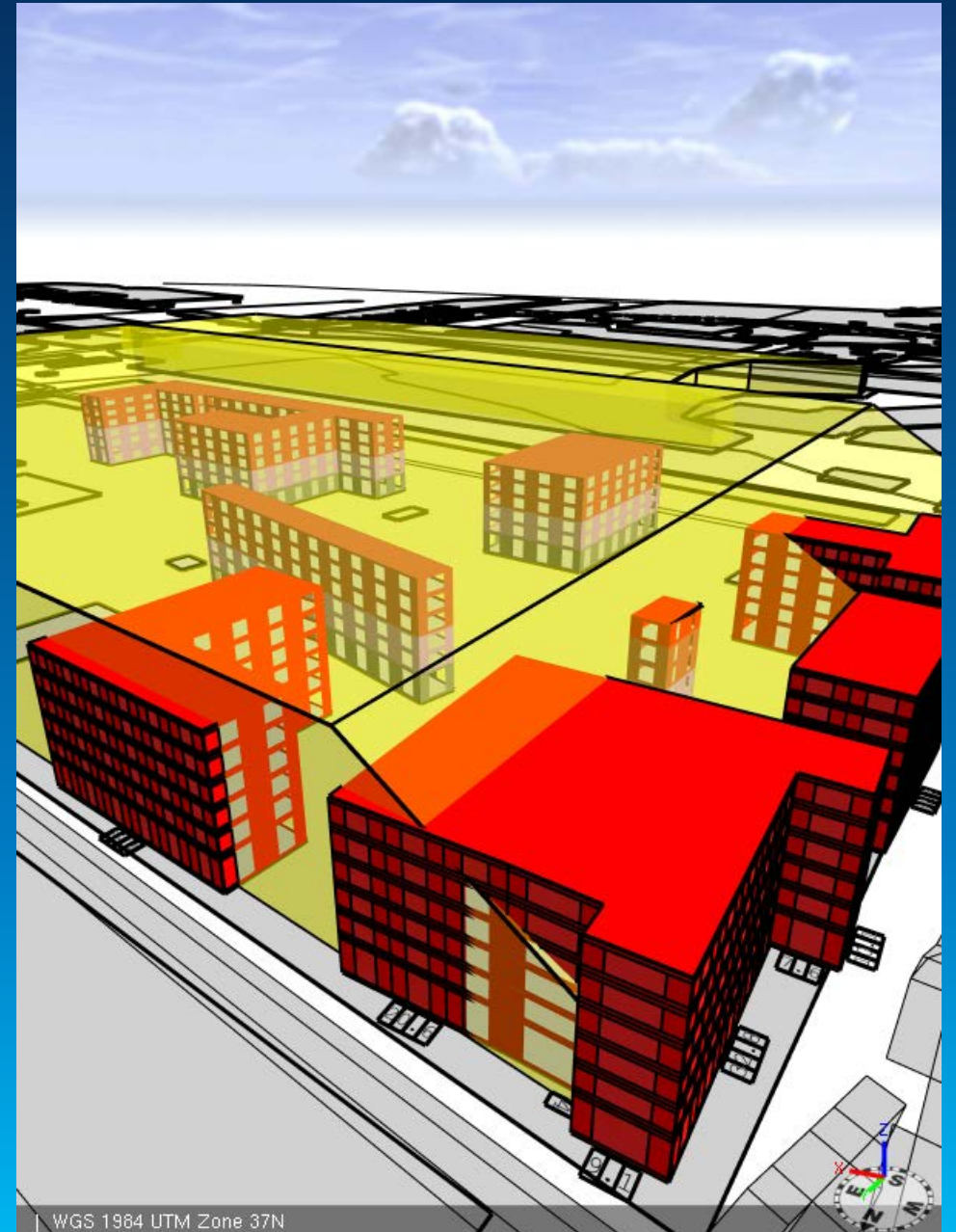


# Zoning Volumes

- envelope operation



- Occlusion checks



# CGA Libraries

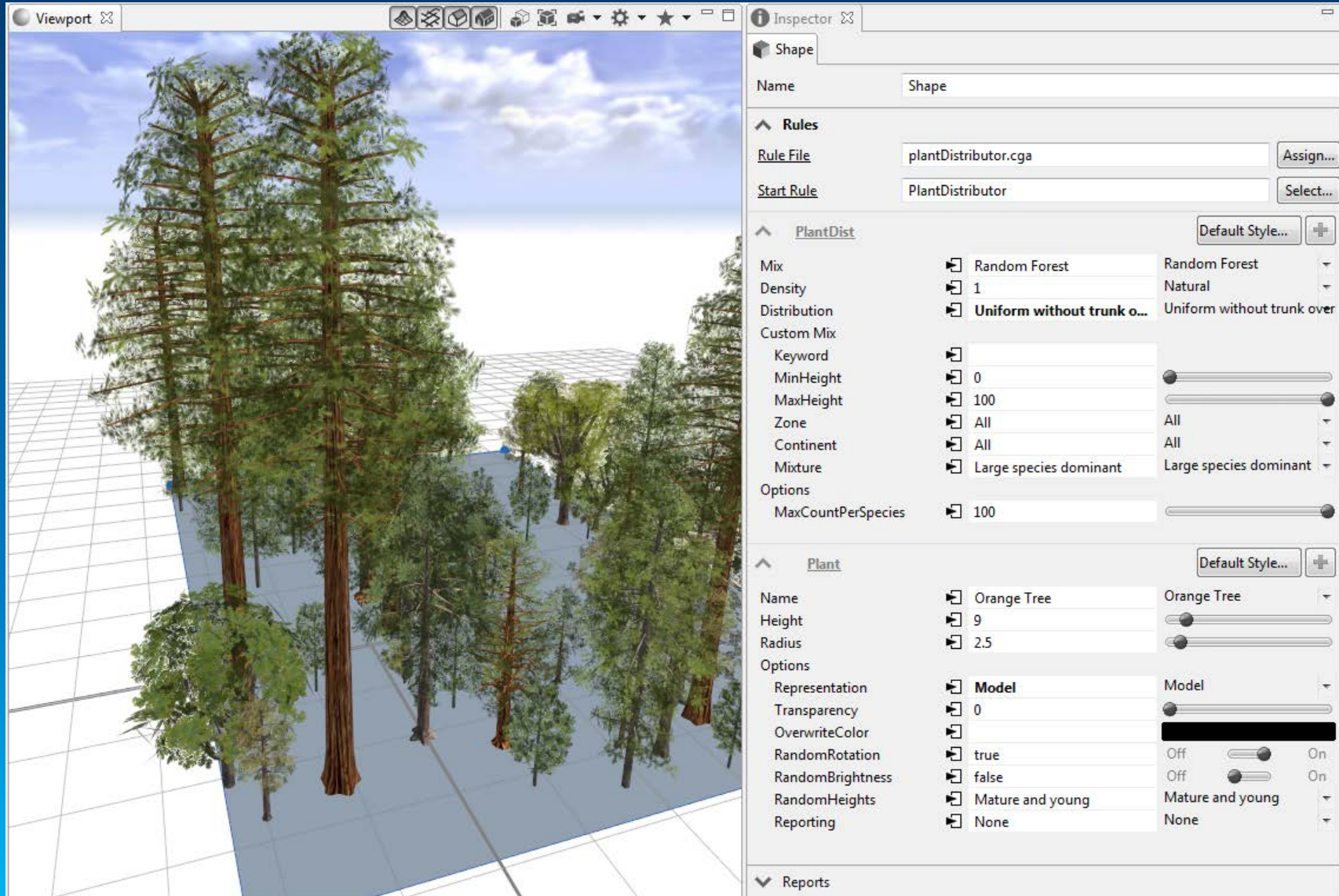
- CGA files can be imported

```
import PlantDist:"/my_city/rules/EsriVegLibDistr.cga"
```

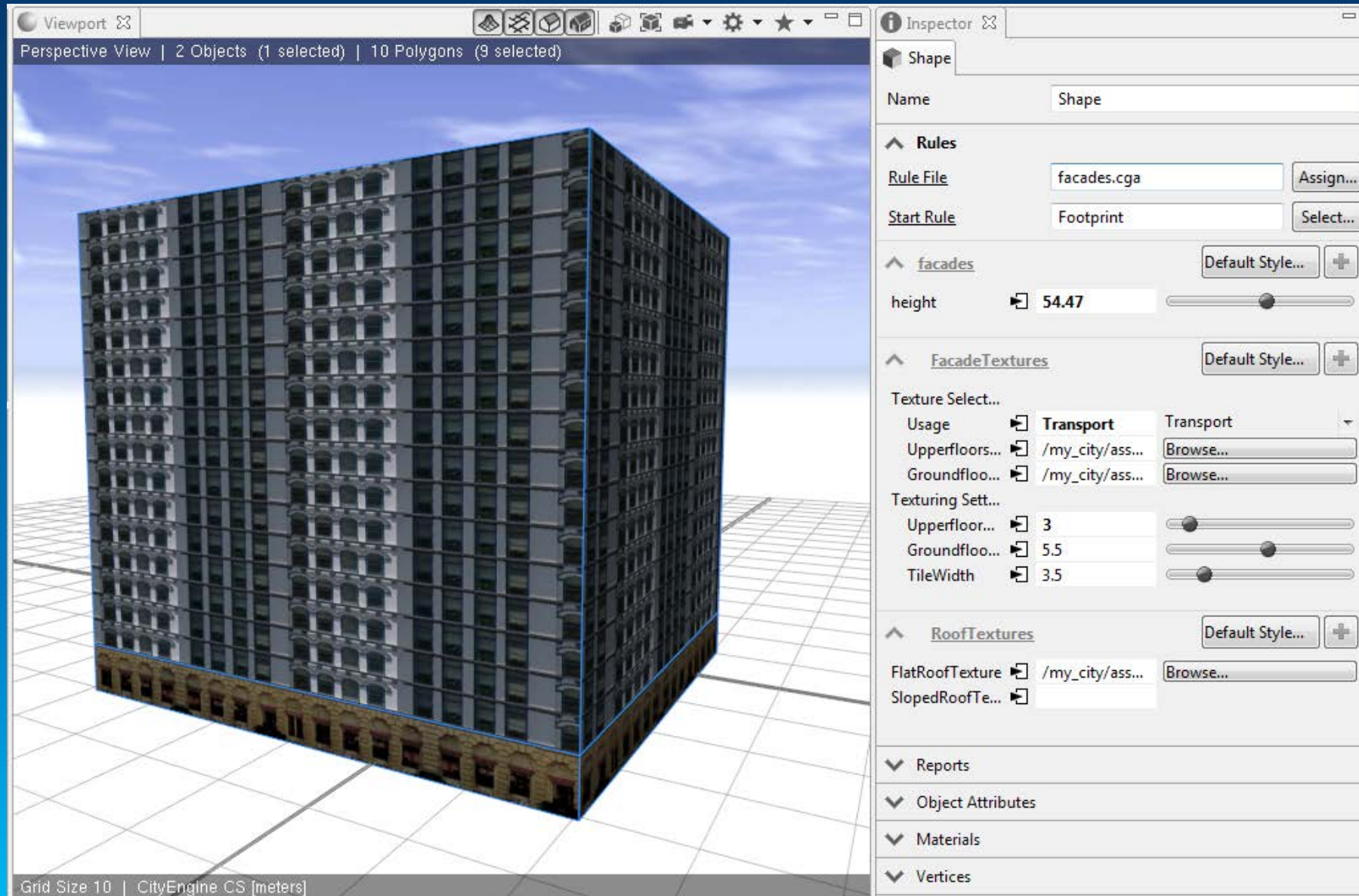
```
PlantDistributor-->  
  PlantDist.Generate
```

- Allows for complex rule & function libraries, e.g.
  - Vegetation
  - Façade Textures
  - Roof Textures
  - Street Construction
  - Utilities

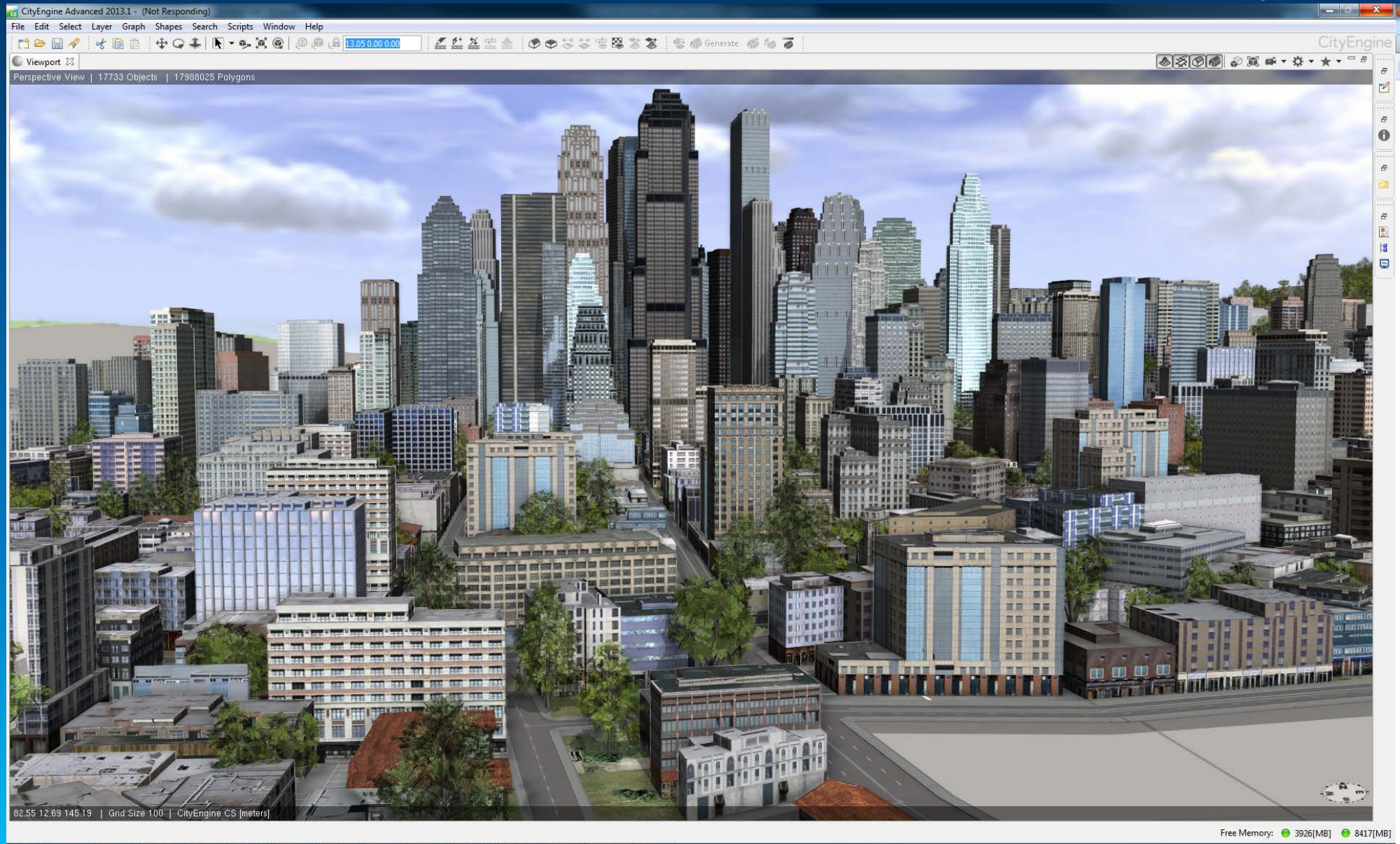
# Use Case 3: Realistic Visualization - Plants



# Use Case 3: Realistic Visualization – Façade Textures

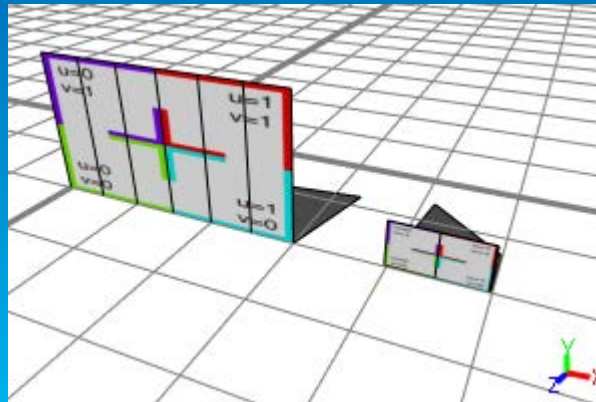


# Use Case 3: Realistic Visualization – International City



# Texturing

- LOD (“simplest geometry”)
- Make model more realistic
- From photogrammetry: textured massmodels
- UV splits (parameterized surfaces!)
- Rendering



## Other CGA Features

That I don't have time to show right now...

- Stochastic Rules / Random
- Recursions for loops
- Texturing
- Roofs
- Styles
- Occlusion
- Geometry cleanup / reduction
- Asset search, (stochastic / best ratio) insertion

# Rule Packages (RPKs)

- **Format to share and exchange compiled CGA code + assets**
- **Compiler compiles textual CGA code to binary CGB**
  - Based on java classfile / bytecode
- **Platform independent**
- **Procedural runtime = virtual machine to execute CGB, RPK = “executable”**
- **Can be used wherever Procedural Runtime is used:**
  - GPTool (ARCGis 10.2)
  - ARCGis Pro
  - Maya plugin
  - Your own specialized application



# Important Tips

- **Assets**

- All will be included → beware the 'random' use of textures and models
- Has a huge impact on file size → consider multiple / focused RPKs

- **Using Tags / Annotations**

- Define input geometry: @InPoint, @InLine, @InPolygon, @InPointCloud, @InMesh

- **Parameters** (ie: the 'attr' definition)

- Use it as much as needed... but no more
- Use a good name → shown on the UI
- Match to data model → auto-connect matching names
- Set Ranges: @Range (v1, v2)
- Provide 'type' information → @Distance, @Percent, @Angle

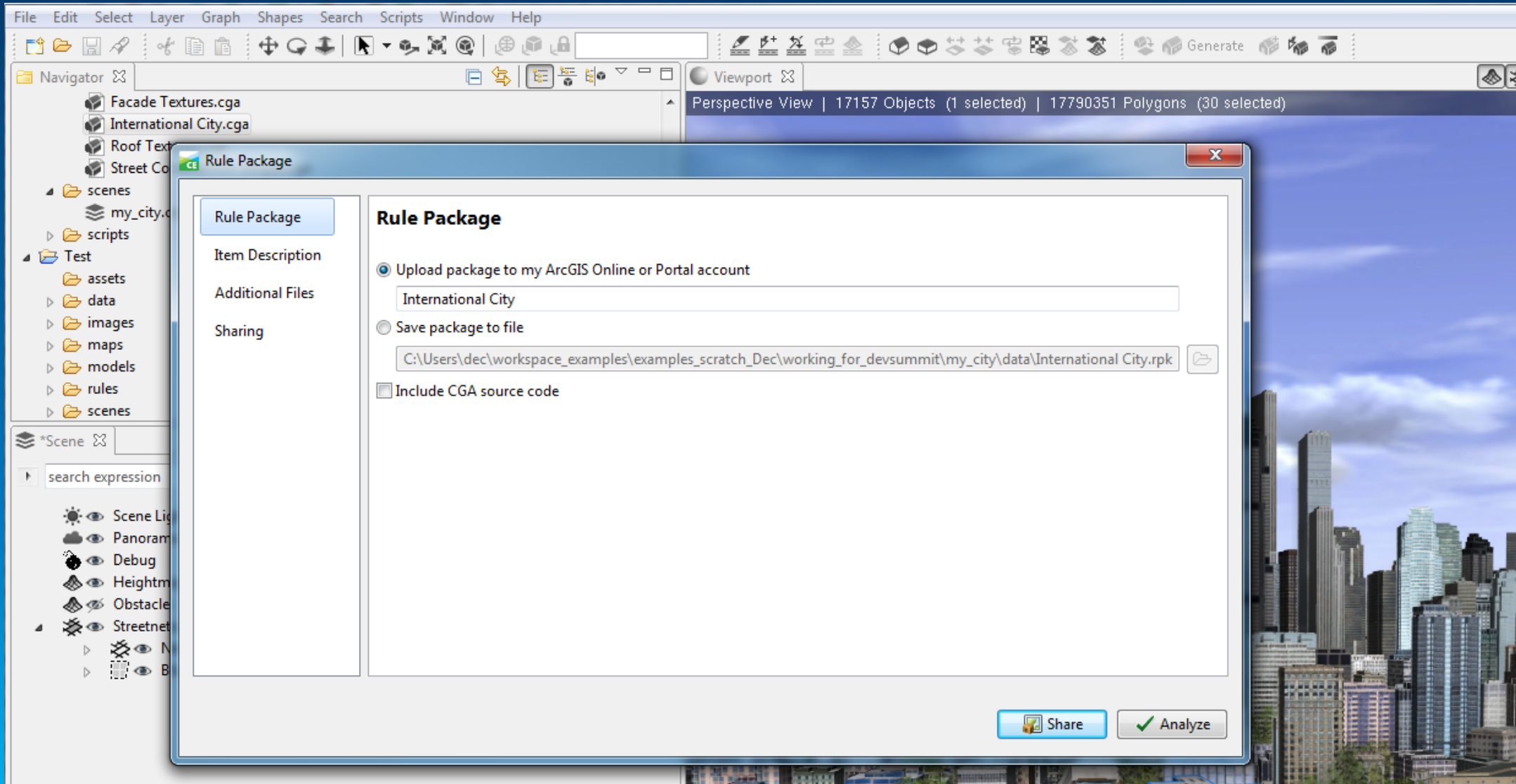
- **Protect the CGA** (as needed)

- If your rule is something you sell, or has IP, then protect it

```
attr floorCount = 3
@Distance
attr roofHeightVal = 2
@Percent @Range(0,80)
attr percentageStripe = 5

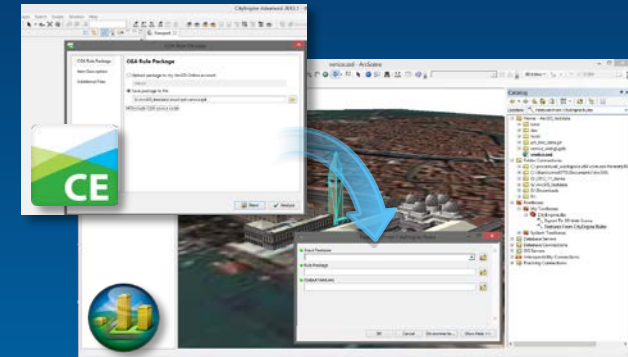
@StartRule @InMesh
BuildingShell -->
```

# Creating a RPK

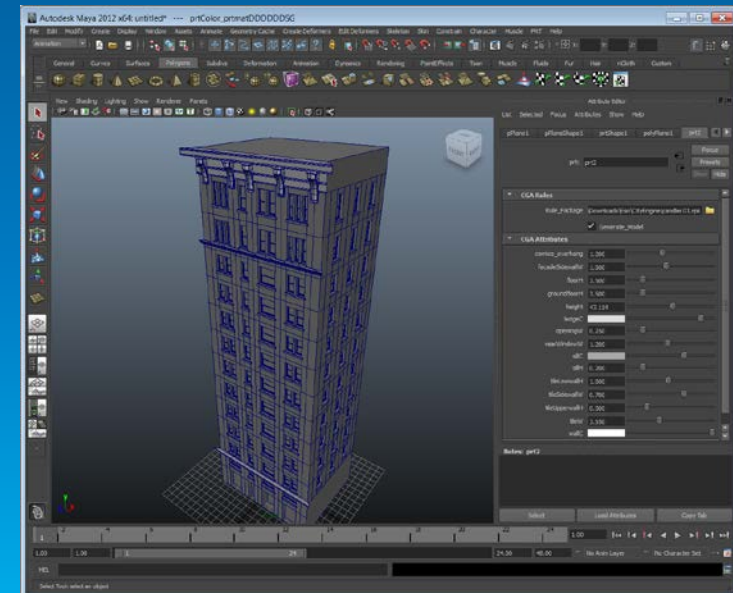


# Sharing / Consuming Rule Packages

- Write CGA rules in CityEngine
- Share as rule package
  - Within your organization
  - Item on Portal
  - Item on AGOL
- Consume in:
  - ArcGIS 10.2
  - ArcGIS Pro
  - 3<sup>rd</sup> party 3D apps
    - CityEngine SDK



ArcScene 10.2



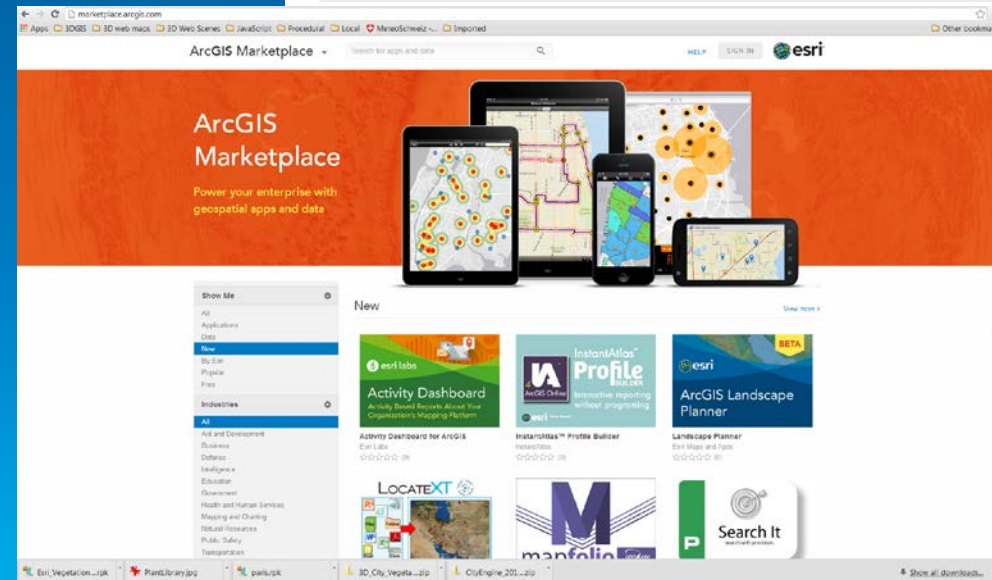
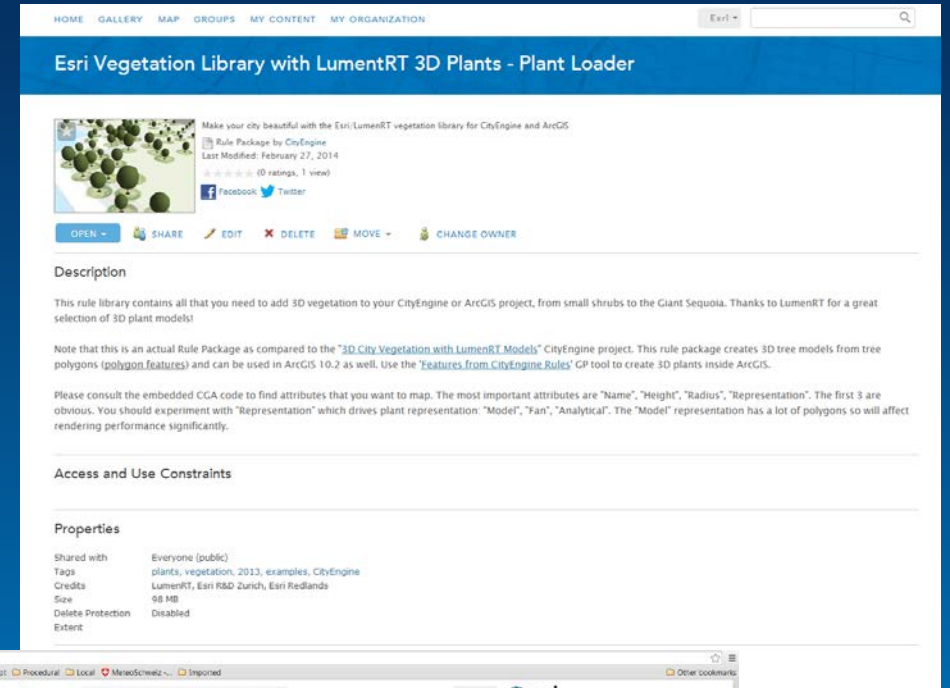
Maya

# Rule Packages on ArcGIS Online

- Item type on ArcGIS Online

- Esri Rule library

- ArcGIS Marketplace
  - *planned for future releases*



## Where to go from here

- **Get a free trial version of CityEngine at:**
  - [http:// www.esri.com/cityengine](http://www.esri.com/cityengine)
- **Have a look at our tutorials**
- **Check out the CityEngine help**
  - Also available online: <http://cehelp.esri.com>
- **Have a look at the forum:**
  - [geonet.esri.com](http://geonet.esri.com) → search for CGA
- <http://www.esri.com/events/devsummit/session-rater>
  - *Session id: 55 offering 261*



Understanding our world.