Creating and sharing Rule Packages with CityEngine
Matthias Specht, Gert van Maren
Esri R&D Center Zurich
Agenda

• Examples (5 min)

• Introduction CityEngine (5 min)

• CGA 101 (25 min)

• Creating Rule Packages (5 min)

• Sharing / Consuming Rule Packages (5 min)
2014 - 3D Across the Platform

- Desktop
  - ArcGIS Pro
  - CityEngine

- Web
  - Web Scene
  - 3D Runtime

- Mobile
  - 3D Runtime

- Server
  - Hosted Services
CityEngine

Transform 2D GIS Data into Smart 3D City Models
Procedural Modeling

3D model creation using rules / algorithms

- Base geometry
Procedural modeling vs. Manual modeling

Time reduction / cost saving
3D City Content Creation

Procedural city modeling

Geometry

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
CGA and Rule Packages
CGA = Computer Generated Architecture

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

```csharp
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       : T
                       | ~rand(0.5,1.5): s
                       | ~1            : s
        15%: split(y){ 'rand48  : Tower
                       | ~1       : t('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

```plaintext
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

```javascript
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

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

```c
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)
Conditional Rules

- case blocks in rules or functions

```py
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

```
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 / paarcels
Use Case 2: Procedural Modeling for Urban Design

- Simple extrude

```plaintext
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**

```plaintext
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

\[
\text{Mass} \rightarrow \\
\quad \text{split}(y) \{ \\
\quad \quad 3.5 : \text{GroundFloor}. \\ 
\quad \quad \{ \sim 2.5 : \text{Floor.} \}^* \\
\quad \} 
\]
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

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

- envelope operation

- Occlusion checks
CGA Libraries

- CGA files can be imported

```python
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
Other CGA Features

• ...I don’t have time for:
  • 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 (ARGGis 10.2)
  - ARCGis Pro
  - Maya plugin
  - Your own specialized application
Creating a RPK

• Go to CE and create one of our rule
• Maybe open it in 7zip
• Pass it to Gert
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
  - 3rd party 3D apps
    - CityEngine SDK
  - Maya
3D Cities Information Model

Built Environment
- Created and actively managed by people
- Structures, utilities, transportation networks, installations

Legal Environment
- Defines restrictions on land use
- Land use zones, property ownership boundaries, regulations

Natural Environment
- Naturally occurring features on, above, or below the earth’s surface
- Land cover, subsurface geology, atmosphere/climate/weather
Rule Packages on AGOL

- User item on AGOL

- Esri Rule library

- ArcGIS Marketplace
  - planned for future releases
2014 - 3D Across the Platform

Desktop

Web

Mobile

Web Scene

3D Runtime

ArcGIS Pro

CityEngine

Server

Hosted Services
Where to go from here

• Get a free trial version of CityEngine at:
  - 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:

• http://www.esri.com/events/devsummit/session-rater
  - Session id: 55 offering 261