ArcGIS Runtime: Analysis

Lucas Danzinger

Mark Baird

Mike Branscomb

2018 Esri DEVSummit Conference | Palm Springs, CA

ArcGIS Runtime session tracks at DevSummit 2018

- ArcGIS Runtime SDKs share a common core, architecture and design
- Functional sessions promote common capabilities and workflows
  - An Introduction to the API and Architecture
  - Working with Your Portal
  - Building Great User Experiences
  - Working with Maps Online and Offline
  - Editing Your Data Online and Offline
  - Integrating Imagery
  - Analysis
  - Building 3D Applications

- Product sessions promote specific development experiences
- Demo theaters highlight examples of specific technical capabilities

Shared workflows, any platform, any device

Agenda

- Geometry operations
- Geoprocessing
- Visual Analysis
- Summary & Roadmap

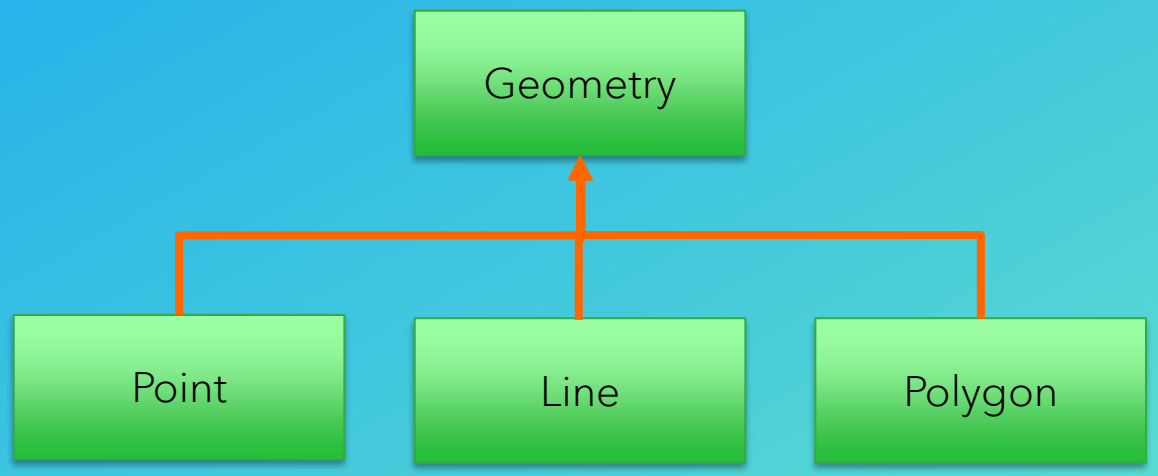
Geometry Operations

Mark Baird

Geometry Engine

- Available on all Runtime SDKs
- Class with static methods for… Geometry
- Geometry can be points, lines or polygons.
- Found in features and graphics
- Client side processing

- 3 groups of functions
  - Geometry dimension questions
  - Geometry spatial questions
  - Building new geometries



Geometry dimension questions

- How long is this line?
- What is the area of this polygon?
- What is the boundary of this polygon?
- What is the distance between A and B?

Java sample

```
// get planar distance between 2 points
double planarDistance = GeometryEngine.distanceBetween(pointA, pointB);
```

- General information:
  - Results are in the unit of the spatial reference of the geometry
  - All geometries must be in the same spatial reference.

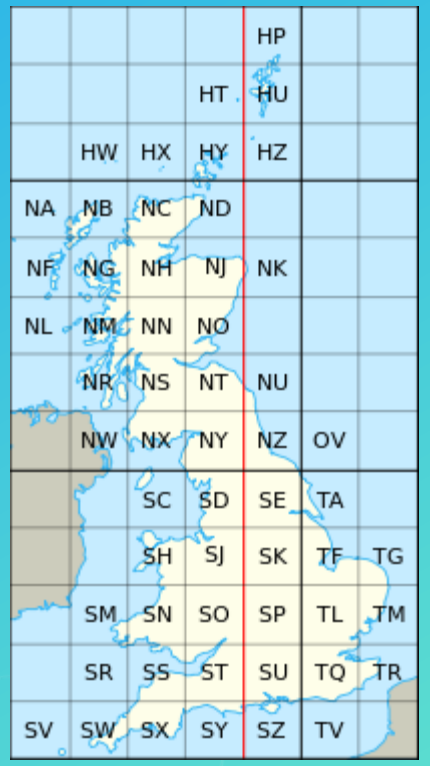
Projecting between spatial references

- Project method.
- Convert from Web Mercator into a local coordinate system.

Java sample

```
// create a OSGB point from a Web Mercator point
Point reprojectedGeometry = (Point) GeometryEngine.project(point, OSGB);
```

- Transformation used to project is chosen for you, but you can specify your own.



Answer spatial questions about geometries

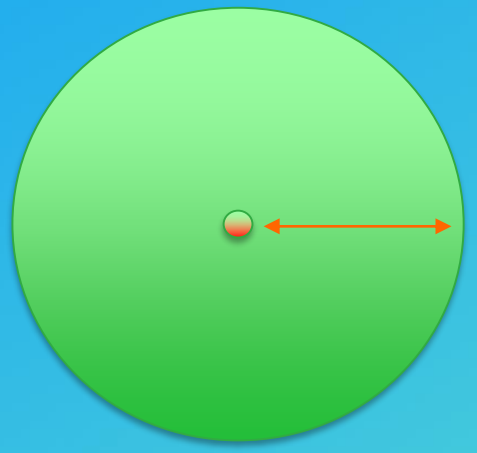
- Contains
- Crosses
- Intersects
- Nearest vertex
- Overlaps
- Touches
- Within

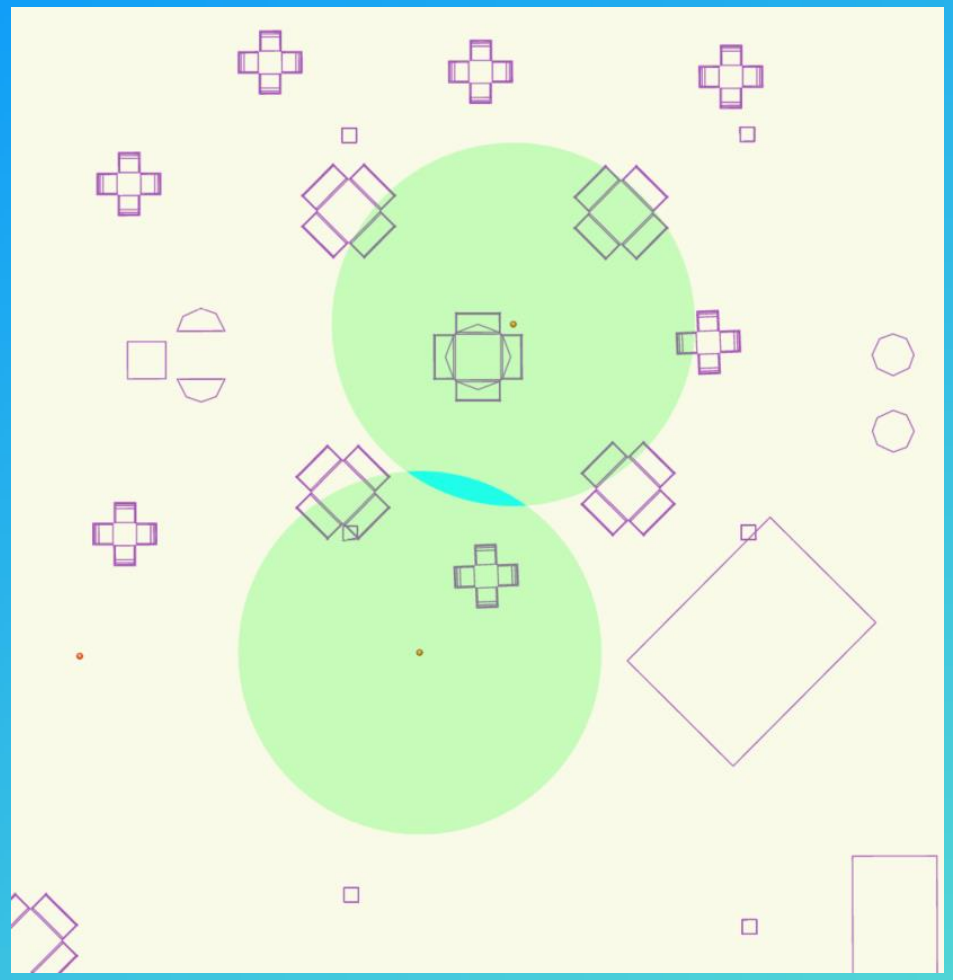
Java sample

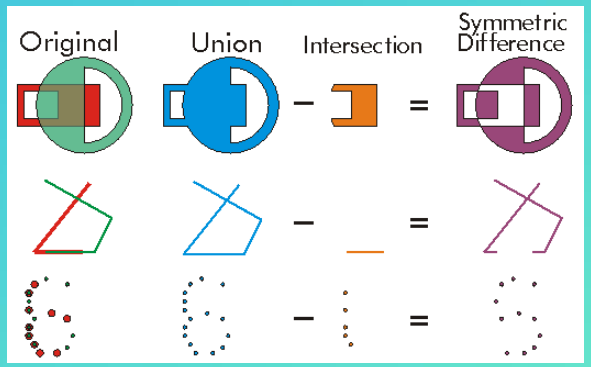
```
// does line cross polygon?
if (GeometryEngine.crosses(polyline, polygon)) {
    // color it green to say yes
    fillSymbol.setColor(0xFF00FF00);
} else {
    // color it red to say no
    fillSymbol.setColor(0xFFFF0000);
}
```

Create geometries based on geometries

- Buffer
- Union
- Intersects
- Difference



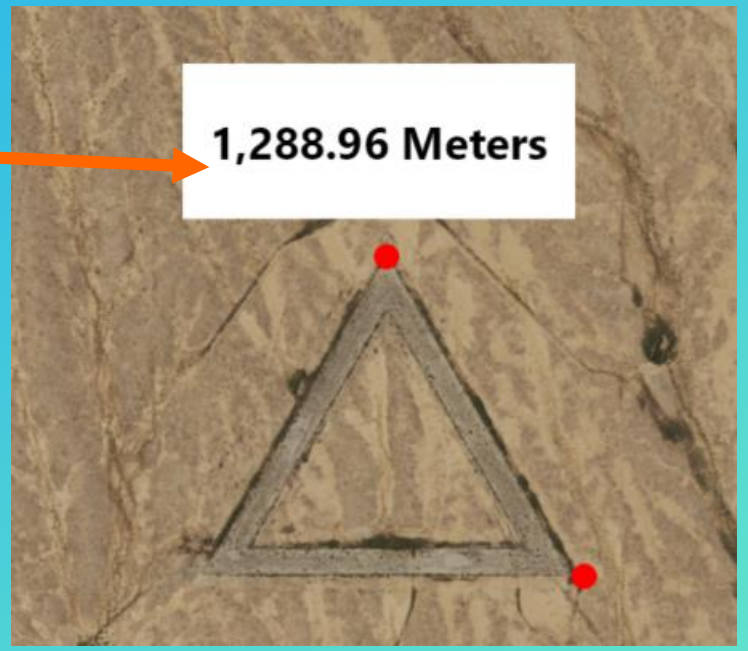




Plus many more…

- Label point

```
GeometryEngine.labelPoint(myPoly)
```



Geoprocessing

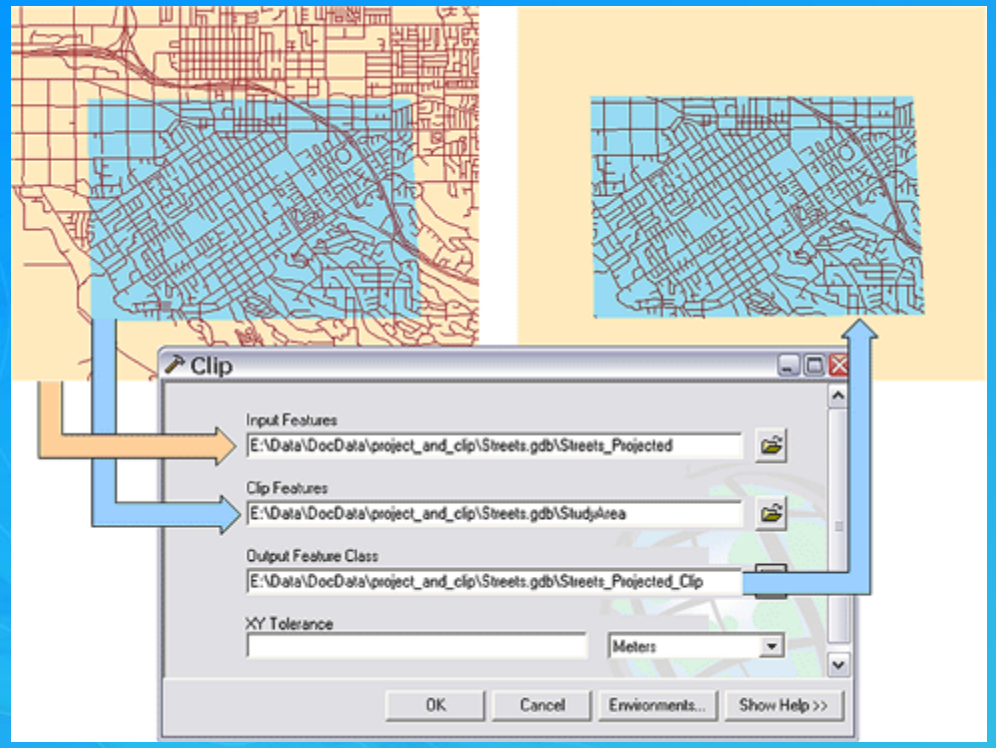
Mike Branscomb

Geoprocessing: Overview

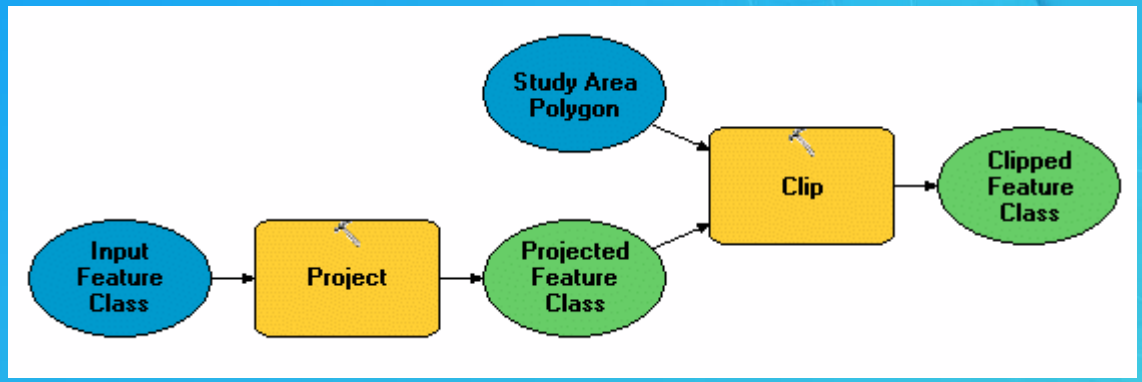
- Framework and tools for processing spatial data
  - Spatial analysis e.g. Buffer, Intersect, Viewshed…
  - Data management e.g. Create Feature Class, Add Field, Add Domain…
- >1200 built-in tools installed with ArcGIS Desktop
- Can be combined into custom tools/workflows
  - Model Builder
  - Python scripts
- Published to ArcGIS Enterprise as services
- Provides REST endpoint for service execution
- ArcGIS Runtime executes geoprocessing tools via services
- Services are designed in ArcGIS Desktop

Geoprocessing: Framework

- ArcGIS tools



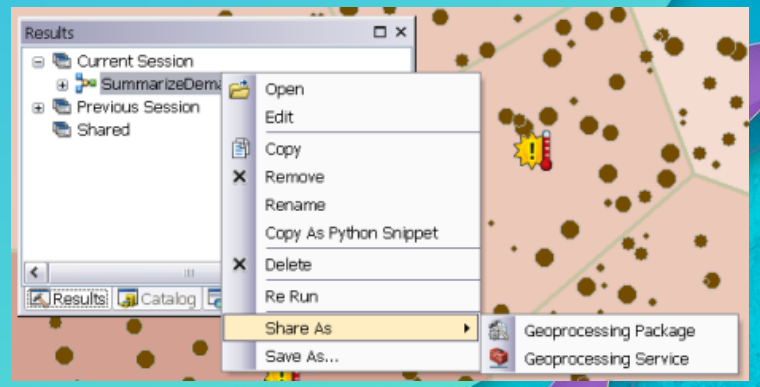
- Custom tools




```
import os
import arcpy

# Set the workspace for ListFeatureClasses
arcpy.env.workspace = "c:/base"

# Use the ListFeatureClasses function to return a list of
#  shapefiles.
featureclasses = arcpy.ListFeatureClasses()

# Copy shapefiles to a file geodatabase
for fc in featureclasses:
    arcpy.CopyFeatures_management(
        fc, os.path.join("c:/base/output.gdb",
                         os.path.splitext(fc)[0]))
```

Geoprocessing: Services

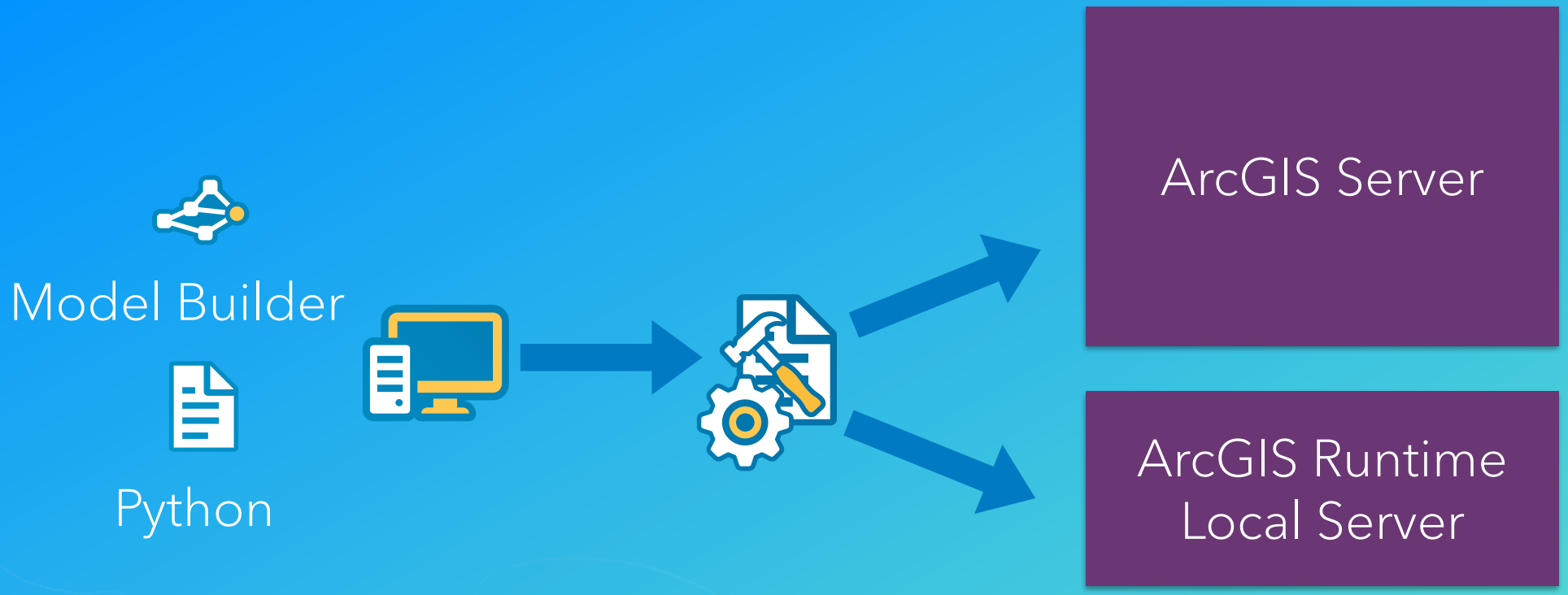
- Must run tool in ArcGIS Desktop before publishing
  - Validates inputs, processing and outputs
- Publish result to ArcGIS Enterprise
  - Result includes input data, output data, project data
- Runtime API provides types to represent geoprocessing service (task) and specific requests to run (jobs)
- Build parameter objects, send to service, wait for results
- Process output parameters
  - e.g. features, tables, raster datasets, values



Geoprocessing: Local Services

- ArcGIS Runtime Local Server includes subset of tools
  - Approx. 300 tools (data management, analysis, extensions)
- Result must be packaged
  - Geoprocessing Package (.gpk)
- Package includes all data required for execution of tool as a service
  - Input data, project data, output data
- LocalServer API includes administration
  - e.g. LocalGeoprocessingService.StartAsync(ServiceType.Asynchronous)

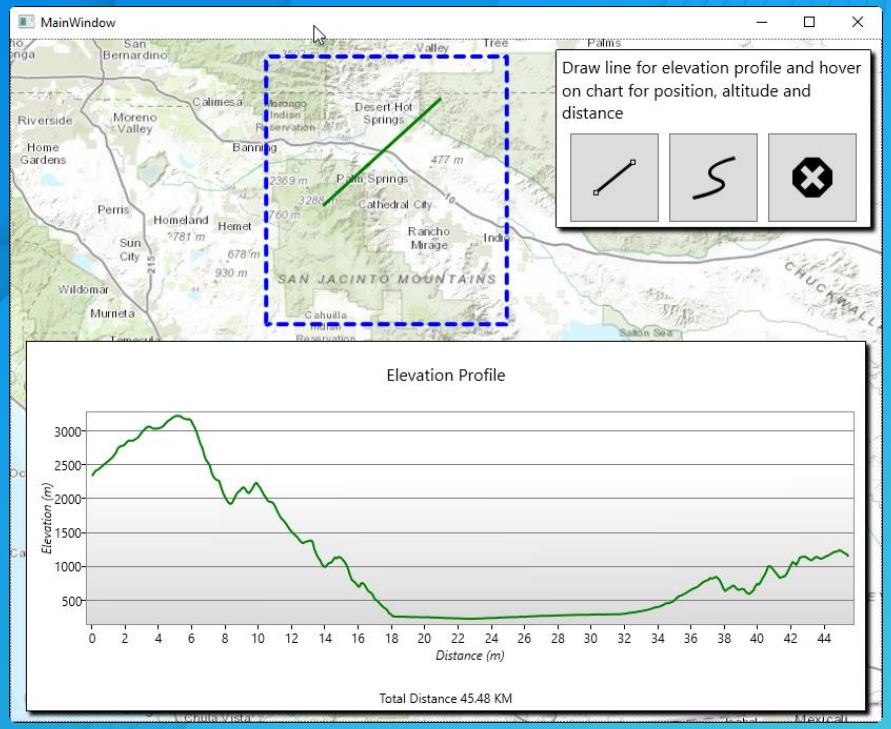
Publishing Geoprocessing services



Geoprocessing: Tips for building geoprocessing tools

- Keep input data and output data small
  - Determines size of the GPK or Service Definition
- Use local data to ArcGIS Server / LocalServer
- Write intermediate data to memory
- Add attribute and spatial indexes
- Avoid unnecessary coordinate transformations
- Reduce data size

Server Help > Performance Tips for Geoprocessing Services



Demo: Geoprocessing

Mike Branscomb

Visual Analysis

Lucas Danzinger

Visual Analysis

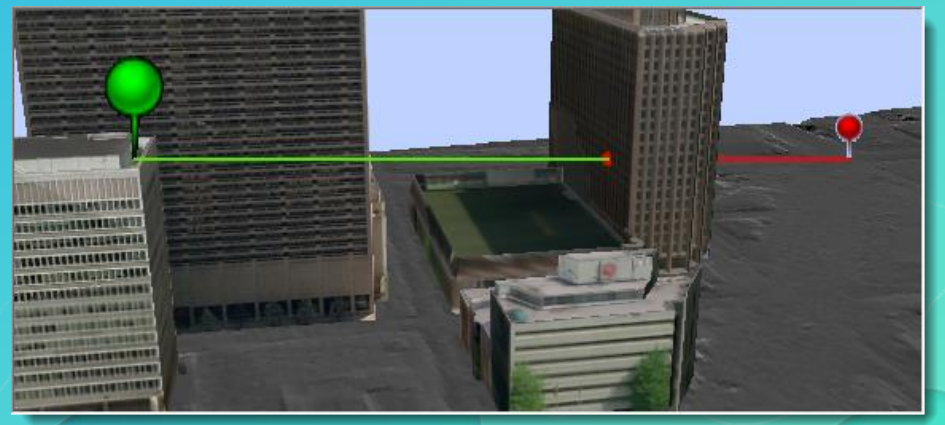
- Visually detect patterns, trends, outliers
- A form of discovery
- Vector and raster data
- 2D and 3D

Exploratory 3D Visual Analysis

- Available analyses:
  - Line of Sight
  - Viewshed
- Fast performance – GPU driven
- Results displayed on Scene, not persisted
  - *differs from traditional GeoProcessing
- Acts on current resolution of data

Exploratory 3D Visual Analysis

- Line of Sight
  - Determines the visible and not visible portions of an imaginary line between a target and an observer
  - Output is a line graphic where visible areas and obstructed areas have different colors
  - 2 varieties:
    - LocationLineOfSight – calculates against 2 points
    - GeoElementLineOfSight – calculates between 2 GeoElements
      - Can attach to 2 GeoElements to auto update
  - Events triggered when target is visible



Exploratory 3D Visual Analysis

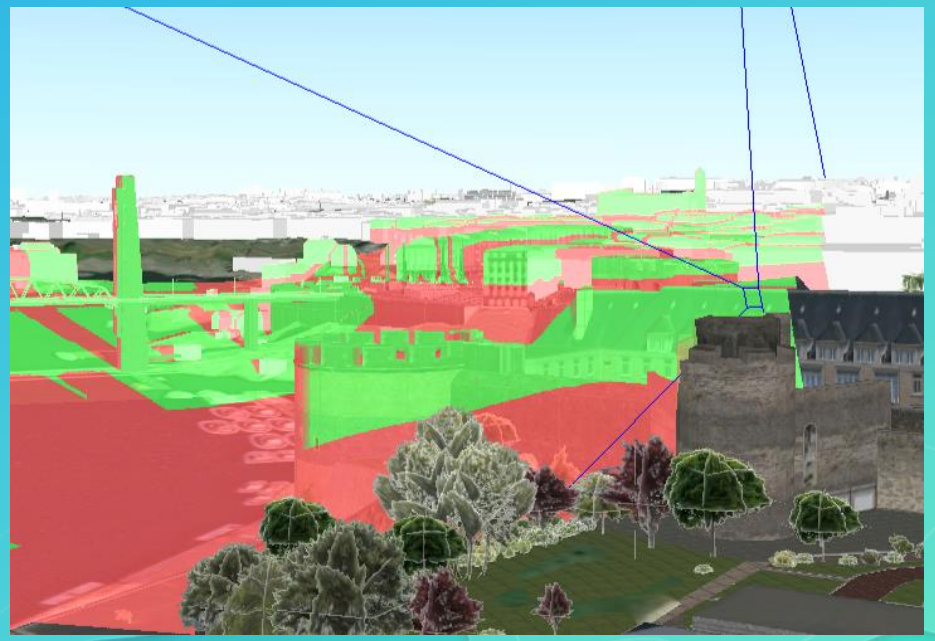
- Line of Sight

Qt C++ sample

```
void LineOfSightLocation::createLineOfSight()
{
  // create the observer/target points
  const Point observerPt(-73.06958032962375, -49.253112971555446, 2000, SpatialReference::wgs84());
  const Point targetPt(-73.079266999709162, -49.300457676730559, 1312, SpatialReference::wgs84());

  // create the line of sight
  m_lineOfSight = new LocationLineOfSight(observerPt, targetPt, this);
  m_analysisOverlay->analyses()->append(m_lineOfSight);

  // configure the LoS color and width
  LineOfSight::setVisibleColor(QColor("cyan"));
  LineOfSight::setObstructedColor(QColor("magenta"));
  LineOfSight::setLineWidth(2.0f);
}
```

Exploratory 3D Visual Analysis

- Viewshed
  - Determines the visible and not visible portions of a surface based on an observer point
  - Output is a raster where visible areas and obstructed areas have different colors
  - 3 varieties:
    - LocationViewshed (Camera)
    - LocationViewshed (Point)
    - GeoElementViewshed (attach to a GeoElement)



Exploratory 3D Visual Analysis

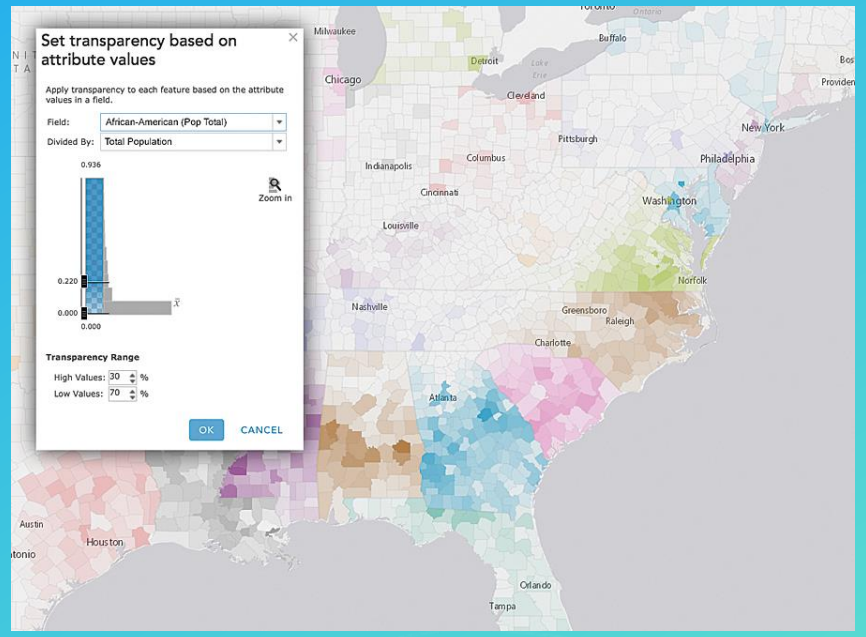
- Viewshed

Qt C++ sample

```
void ViewshedLocation::createViewshed(double x, double y)
{
  const Point pt = m_sceneView->screenToBaseSurface(x, y);

  // Create the Location Viewshed
  m_locationViewshed = new LocationViewshed(pt, m_heading, m_pitch,
                                            m_horizontalAngle, m_veriticalAngle,
                                            m_minDistance, m_maxDistance, this);
  m_locationViewshed->setVisible(m_viewshedVisible);

  // Add the Viewshed to the Analysis Overlay
  m_analysisOverlay->analyses()->append(m_locationViewshed);
}
```

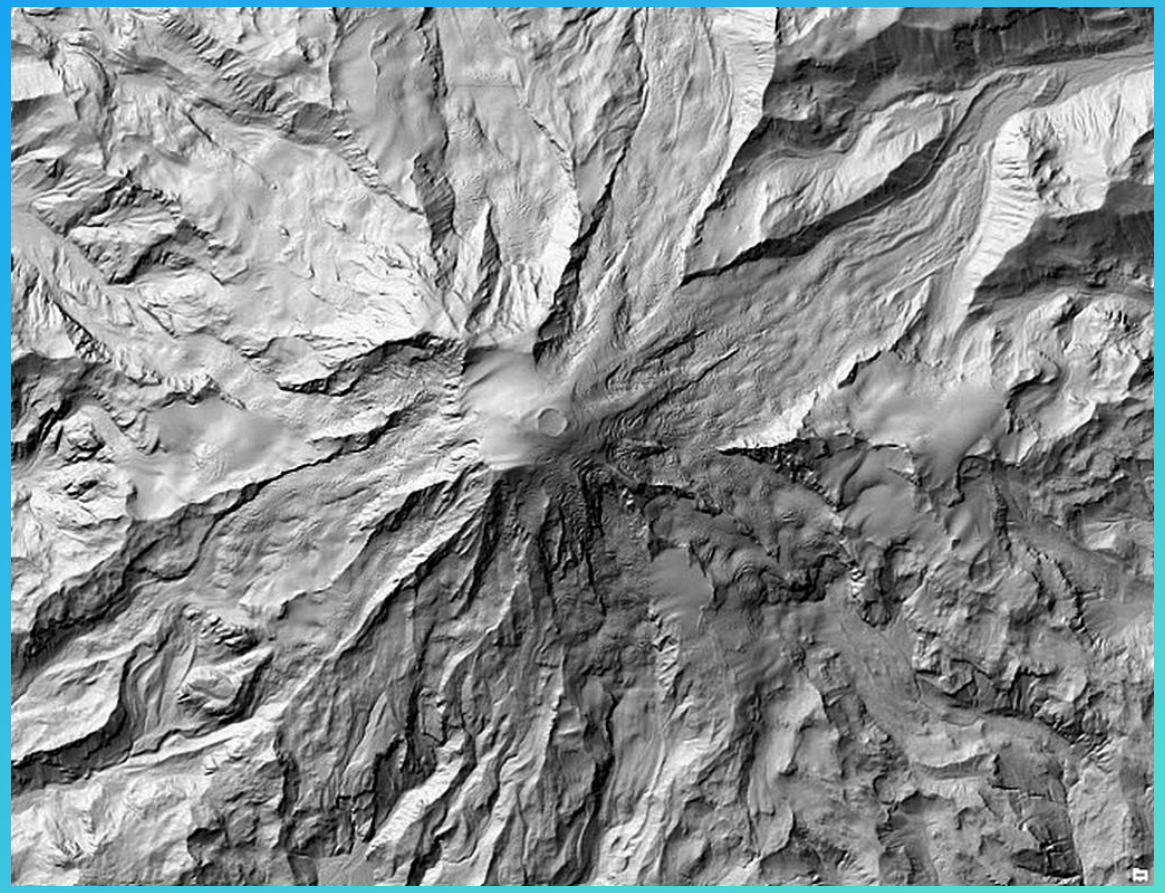
Visualizing vector data

- Apply symbols based on rules
  - Feature attribute data values
- Unique Value Renderer
  - Different symbol per unique attribute value
  - Ex: A state's favorite sports team
- Class Breaks Renderer
  - Different symbol per range of attribute values
  - Ex: Median income
- Smart Mapping Renderers
  - More and more being added to Runtime



Visualizing raster data

- Hillshade renderer
- Blend renderer
- Stretch renderer
- RGB renderer
- Colormap renderer



Summary & Roadmap

Mike Branscomb

Summary

- ArcGIS Runtime enables spatial analysis in your native apps
- Common to combine approaches
  - Geometry Engine > Geoprocessing > Visualization
- Local to the device
  - Offline workflows
- In your enterprise or the cloud
  - Online service-based workflows

Roadmap for analysis in ArcGIS Runtime 100.x (Dev Summit 2017)

- ✓ Geoprocessing additional tools for Local Server
- ✓ Renderers and additional support for smart mapping
- ✓ 3D on-the-fly visual analysis e.g. viewshed, line of sight
- ✓ Network analysis:
  - Closest Facility (online and offline)
  - Service Area (online and offline)

Roadmap for analysis in ArcGIS Runtime

- Geoprocessing additional tools for Local Server
- More 3D exploratory analysis tools e.g. 3D distance
- Local Server support for ArcGIS Pro packages
- Local Server support for ArcMap 10.6 packages

Thank you

Questions?

esri

THE SCIENCE OF WHERE