



# Building Analysis Into Your Command and Control Systems

Kerry Robinson

Parker Hornstein

An abstract 3D architectural graphic on the right side of the slide. It features various geometric shapes in shades of blue, orange, and green, some with topographic contour lines. The shapes are layered and angled, creating a sense of depth and perspective. The overall aesthetic is modern and technical.

**GIS  
INSPIRING  
WHAT'S  
NEXT**

# agenda



Kerry Robinson

Geoprocessing, aka “Conclusive” Analysis

[krobinson@esri.com](mailto:krobinson@esri.com)



Parker Hornstein

Visual Analysis, aka “Exploratory” Analysis

[phornstein@esri.com](mailto:phornstein@esri.com)

# Geoprocessing

“Conclusive” Analysis

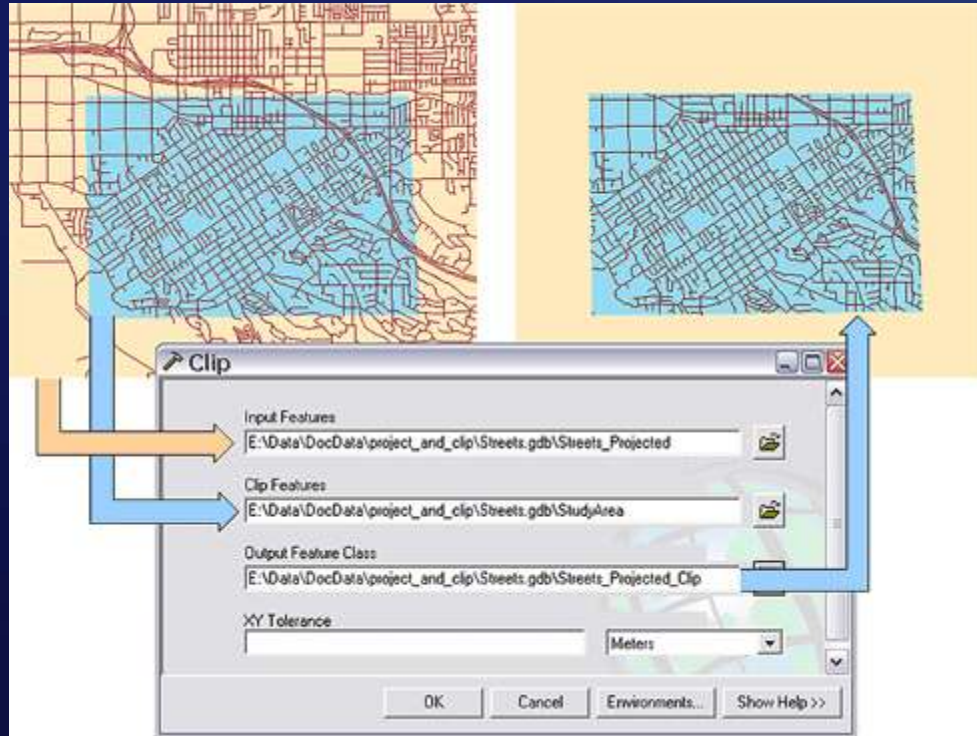
# Geoprocessing: Overview

- Framework and tools for processing spatial data in ArcGIS
  - Spatial analysis e.g. Buffer, Intersect, Viewshed...
  - Data management e.g. Create Feature Class, Add Field, Add Domain...
- ArcGIS Desktop
  - >1200 built-in tools installed
  - Author custom tools/workflows using Model Builder and/or Python scripts
- Publish to ArcGIS Enterprise
  - Geoprocessing Services designed in ArcGIS Desktop
  - Provides a REST endpoint for service execution
- ArcGIS Clients/APIs
  - Execute geoprocessing tools via services (Enterprise or Local)

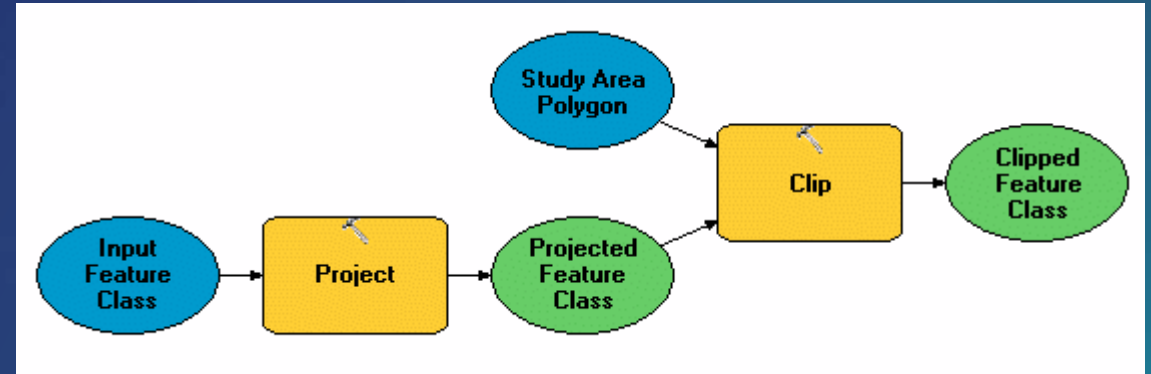


# Geoprocessing: Framework

- ArcGIS tools



- Custom tools



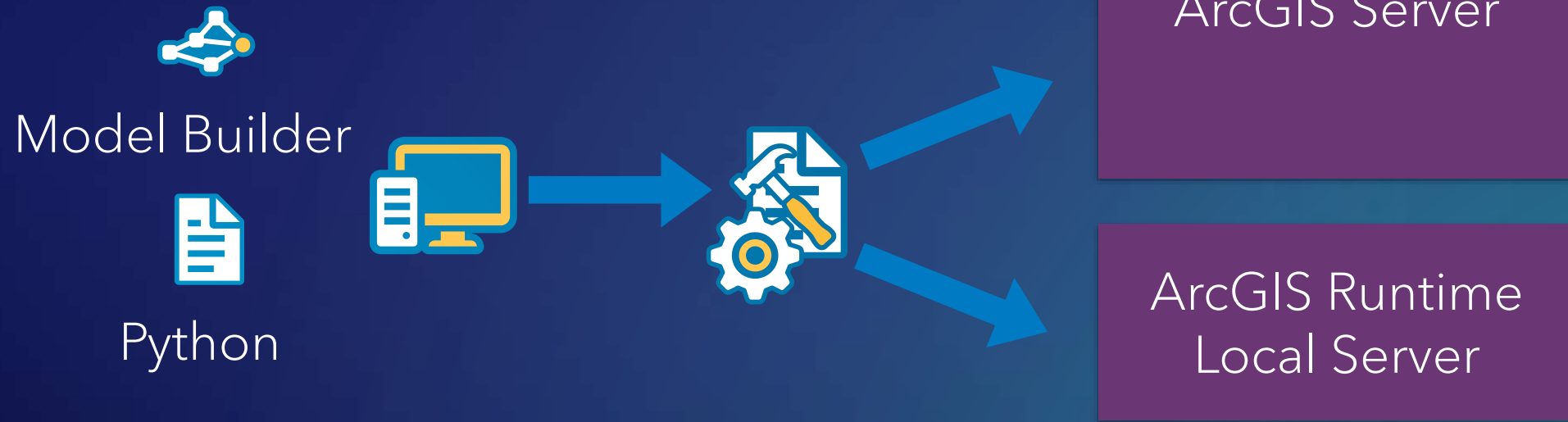
```
import os
import arcpy

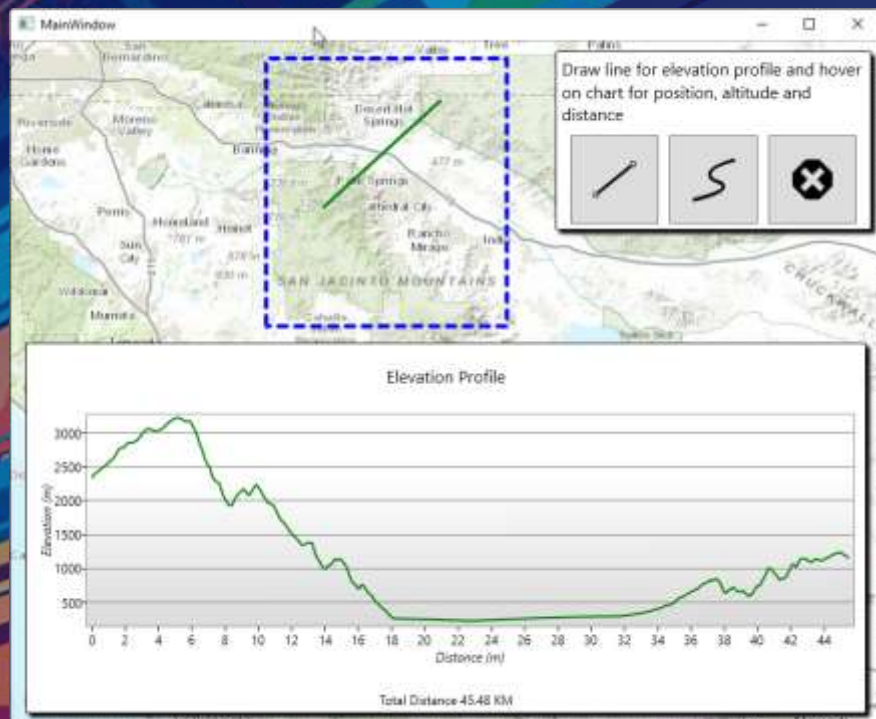
# Set the workspace for ListFeatureClasses
arcpy.env.workspace = "c:/base"

# 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]))
```

# Publishing Geoprocessing services





# Demo

## GP in Runtime

# Geoprocessing: Services

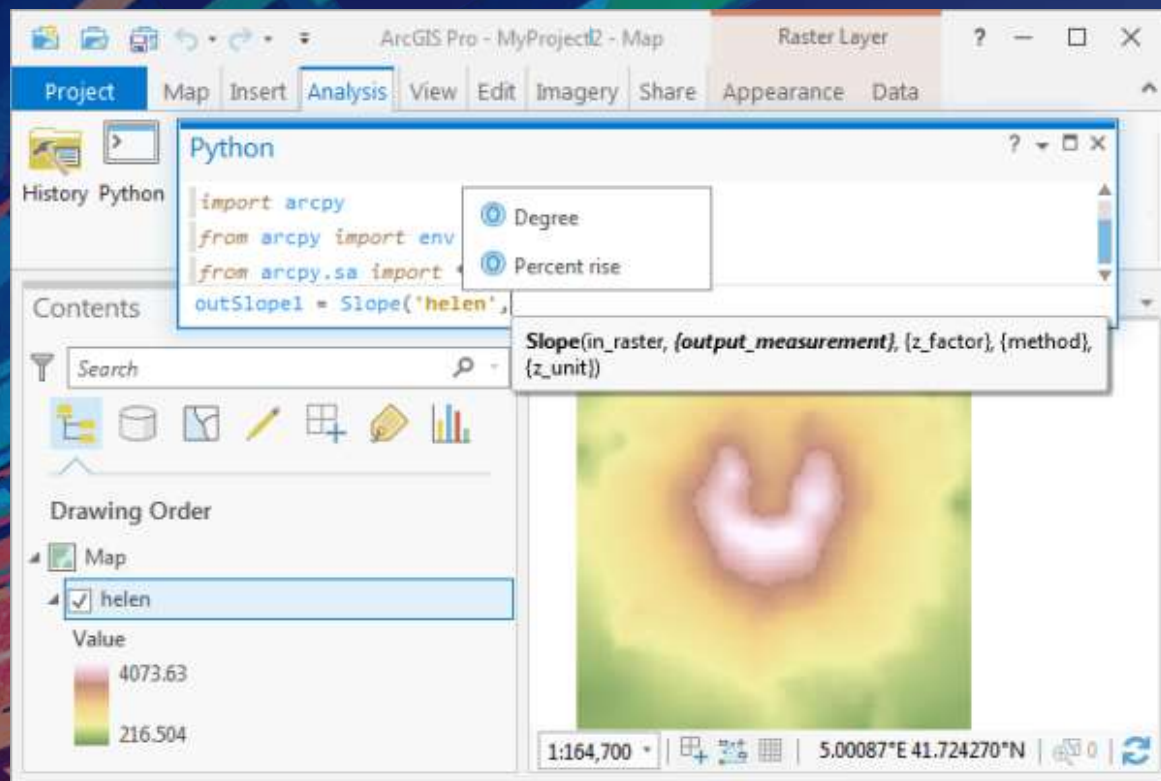
- Must run tool in ArcGIS Desktop before publishing
  - Validates inputs, processing and outputs
- Publish *result* to ArcGIS Enterprise
  - Result includes input data, output data, project data
- Runtime API provides types to represent geoprocessing service (task) and specific requests to run (jobs)
- 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/.gpkx)
- 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)`



# Demo

Authoring & Publishing

# 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](#)

# Visual Analysis

“Exploratory”



# 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 resolution of data in display



# 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());

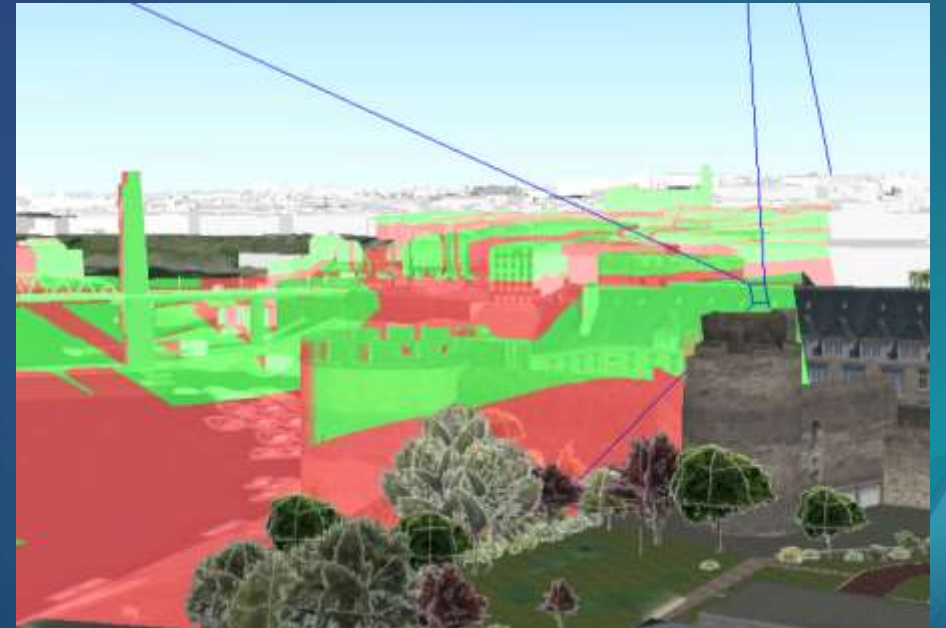
    // create the line of sight
    m_lineOfSight = new LocationLineOfSight(observerPt, targetPt, this);
    m_analysisOverlay->analyses()->append(m_lineOfSight);

    // 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 displays visible areas and obstructed areas using different colors
- 3 varieties:
  - LocationViewshed (Camera)
  - LocationViewshed (Point)
  - GeoElementViewshed (attach to a GeoElement)





# Exploratory 3D Visual Analysis

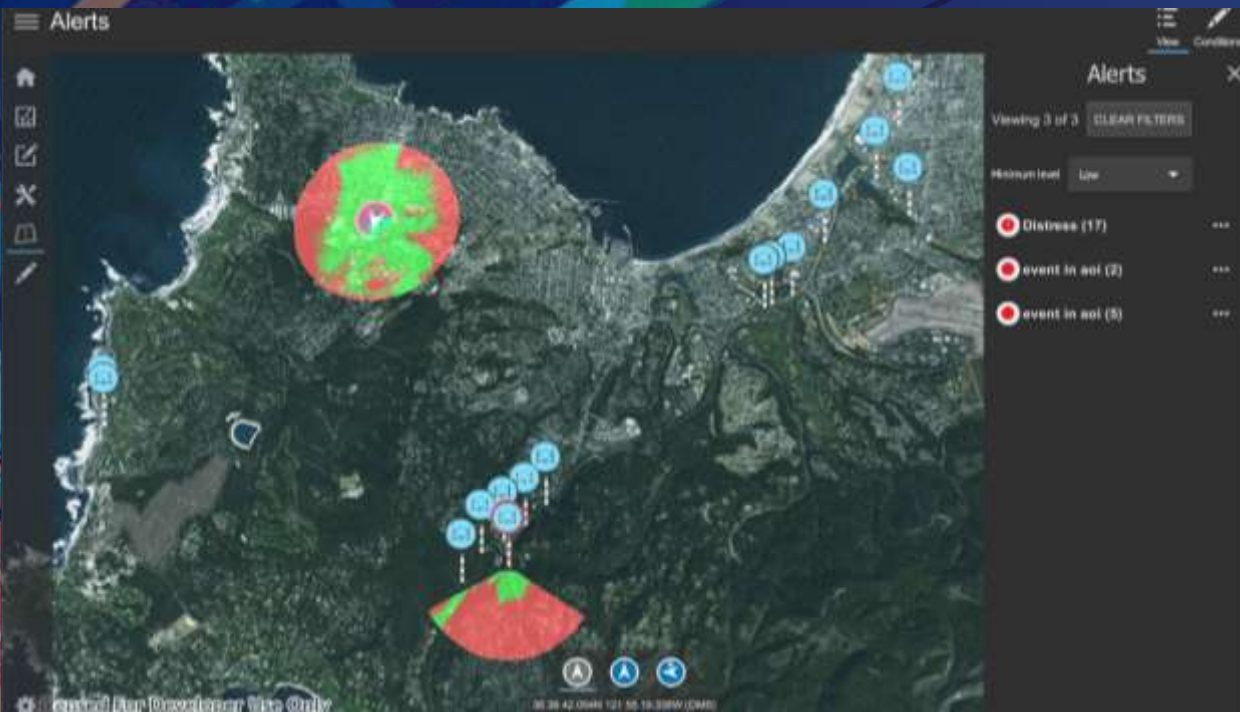
- Viewshed

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);
}
```



Demo

# Summary

- ArcGIS Runtime enables several ways to perform spatial analysis in your native apps
  - Conclusive Analysis (Geoprocessing, Geometry Engine)
  - Exploratory Analysis (Visibility)
- Common to combine approaches
  - Geometry Engine > Geoprocessing > Visualization
- Local to the device
  - Offline workflows
- In your enterprise or the cloud
  - Online service-based workflows

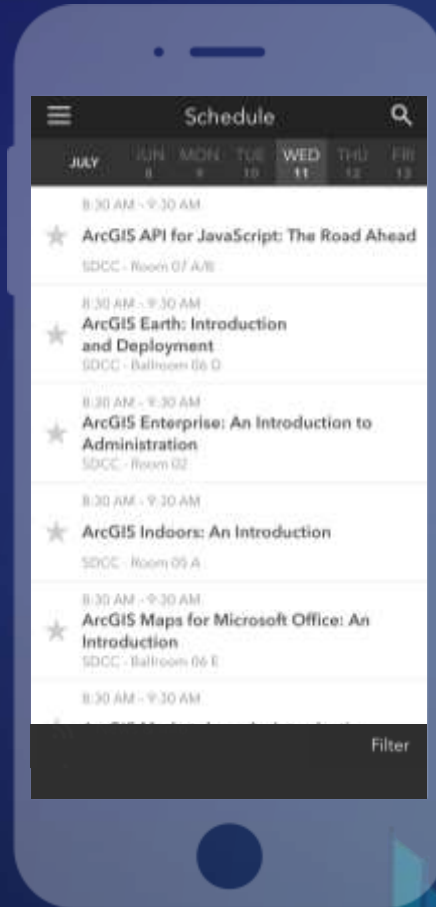


# Please Take Our Survey on the App

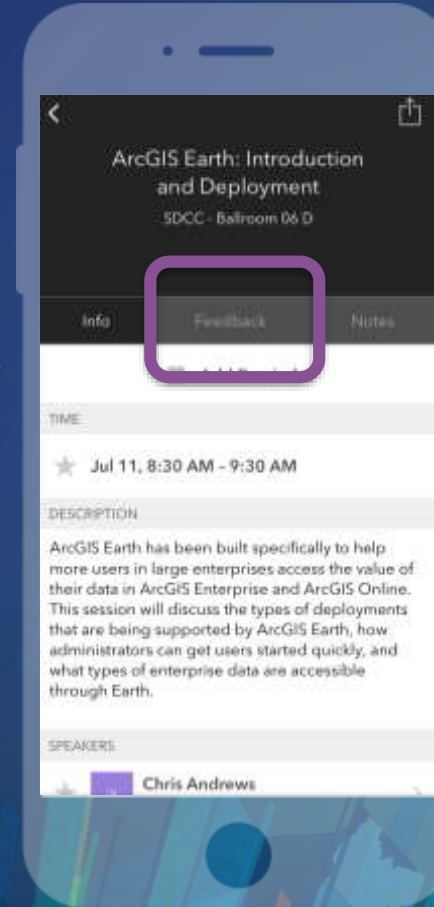
Download the Esri Events app and find your event



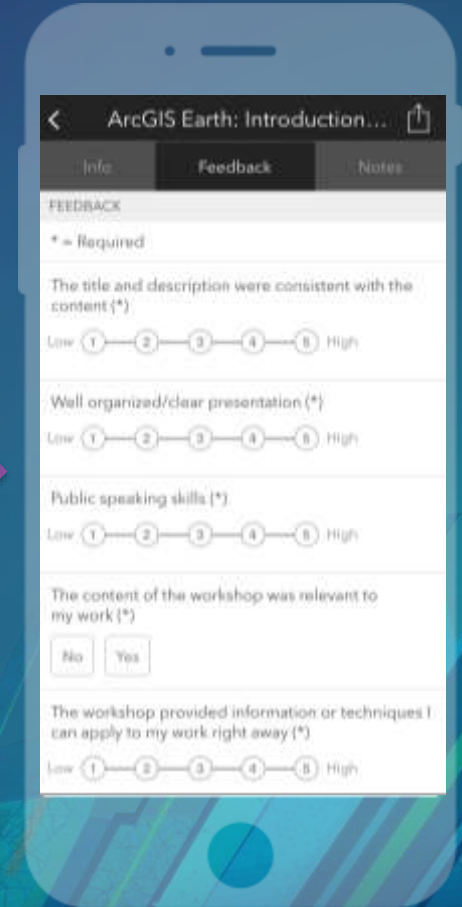
Select the session you attended



Scroll down to find the feedback section



Complete answers and select "Submit"







**esri**

**THE  
SCIENCE  
OF  
WHERE**