3D Visualization with the ArcGIS API for JavaScript

Kristian Ekenes
Raluca Nicola
Kristian Ekenes
Product Engineer JS API
Data visualization, Smart Mapping, Arcade
Redlands, California

Raluca Nicola
Product Engineer JS API
Data visualization, 3D
Zurich, Switzerland
Discover how to visualize geodata in a meaningful way in 3D using ArcGIS API for JavaScript
Points of interest in a city
Urban development
Visualization API - basics -

Three use cases - hands on demos -

Symbols & renderers - general overview -

Sneak peak into the future

5 min 20 min 10 min 5 min
Visualization API

Basic patterns
var layer = new FeatureLayer({
    renderer: new SimpleRenderer(
        symbol: new PointSymbol3D({
            symbolLayers: [new IconSymbol3DLayer({})]
        })
    ),
    elevationInfo: {
        mode: "relative-to-scene"
    },
    labelingInfo: [
        {symbol: new LabelSymbol3D({
            symbolLayers: [new TextSymbol3DLayer({})]
        })
    ]
});
```javascript
var layer = new FeatureLayer({
    renderer: new SimpleRenderer({
        symbol: new PointSymbol3D({
            symbolLayers: [new IconSymbol3DLayer({...})]
        }),
    }),
    visualVariables: [{
        type: "size",
        field: "TOTPOP_CY",
        normalizationField: "SQMI",
        stops: [{
            value: 4000,
            size: 6
        }, {
            value: 23000,
            size: 40
        }]
    }]
});
```

for numeric continuous visualizations
Data-driven visualization

- Renderers & Visual Variables

1. Field value(s)
   - TOTAL_POP

2. Arcade expression
   - Round( ($feature.BACHELOR + $feature.MASTER) / $feature.POP_25UP) );

3. JavaScript function
   - function (graphic){
     return graphic.attributes.POP_DENSITY;
   }
```javascript
var layer = new FeatureLayer({
  renderer: new SimpleRenderer({
    symbol: new PointSymbol3D({
      symbolLayers: [new IconSymbol3DLayer({})]
    })
  }),
  elevationInfo: {
    mode: "relative-to-scene"
  },
  labelingInfo: [{
    symbol: new LabelSymbol3D({
      symbolLayers: [new TextSymbol3DLayer({})]
    })
  }]
});
```
layer

elevationInfo

mode

relative to ground

relative to scene

absolute height

on the ground

relative to ground
var layer = new FeatureLayer({
    renderer: new SimpleRenderer({
        symbol: new PointSymbol3D({
            symbolLayers: [new IconSymbol3DLayer({})]
        })
    }),
    elevationInfo: {
        mode: "relative-to-scene"
    },
    labelingInfo: [{
        symbol: new LabelSymbol3D({
            symbolLayers: [new TextSymbol3DLayer({})]
        })
    }]
});
Three use cases
Hands on demos
Points of interest in a city

Demo time
Smart mapping

Demo time
Symbols and Renderers

General overview
Symbols

- **PointSymbol3D**
- **LineSymbol3D**
- **PolygonSymbol3D**
- **IconSymbol3DLayer**
- **LineSymbol3DLayer**
- **FillSymbol3DLayer**
- **ObjectSymbol3DLayer**
- **PathSymbol3DLayer**
- **ExtrudeSymbol3DLayer**

**Flat**
- Screen size units (pt, px)
- Graphics are in screen space

**Volumetric**
- Real world units (feet, meter)
- Graphics are in real world space

Real world units (feet, meter)
- Graphics are in real world space
Symbols

MeshSymbol3D

FillSymbol3DLayer
Renderers

SimpleRenderer

UniqueValueRenderer

ClassBreaksRenderer
Multivariate mapping with visual variables

visualVariables - only for continuous numerical visualization

Color

Size

Opacity

Rotation
Sneak peak into the future
Edge rendering – already in 4.7
Lighting conditions