Extend the Viewer

Petr Broz

Forge Developer Advocate

Denis Grigor

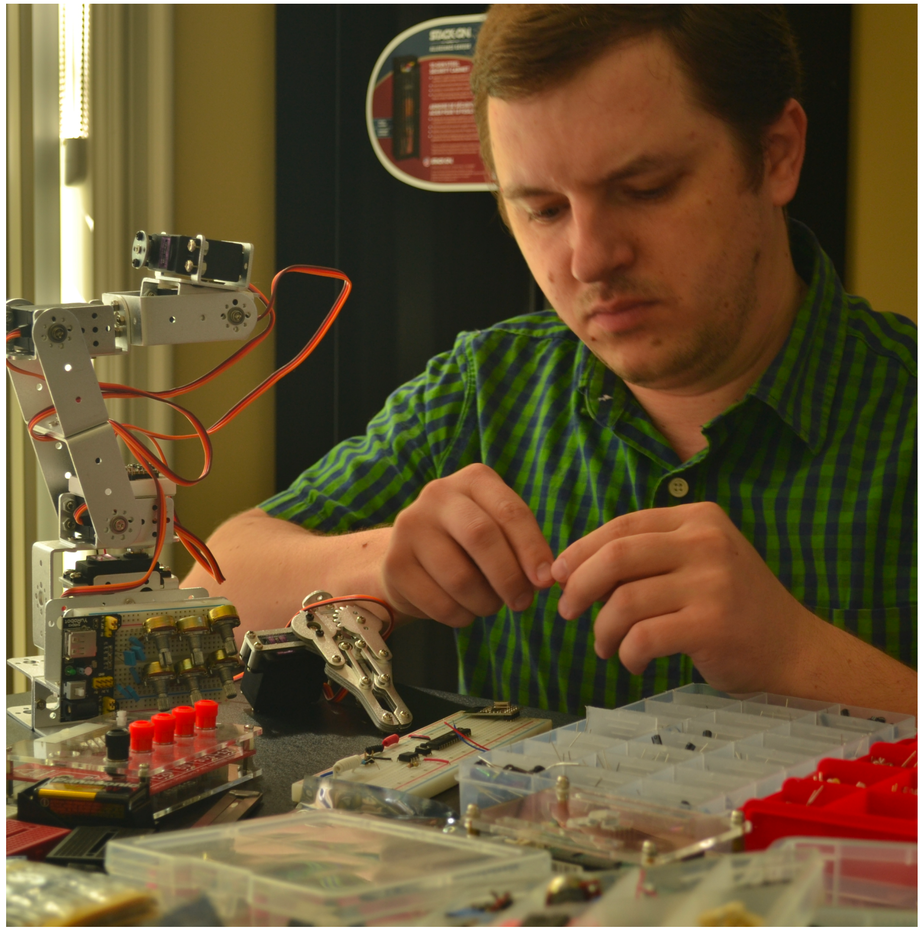
Forge Developer Advocate

AUTODESK. UNIVERSITY

About the speaker

Petr Broz

Petr is a senior developer advocate at Autodesk. He has been developing some of the company's web applications and services since 2011, and now he aims to help others build amazing, creative solutions using these tools, especially using Autodesk Forge APIs. Petr's technical focus areas include everything around web and cloud, 3D graphics, and all kinds of "non-real reality".



About the co-speaker

Denis Grigor

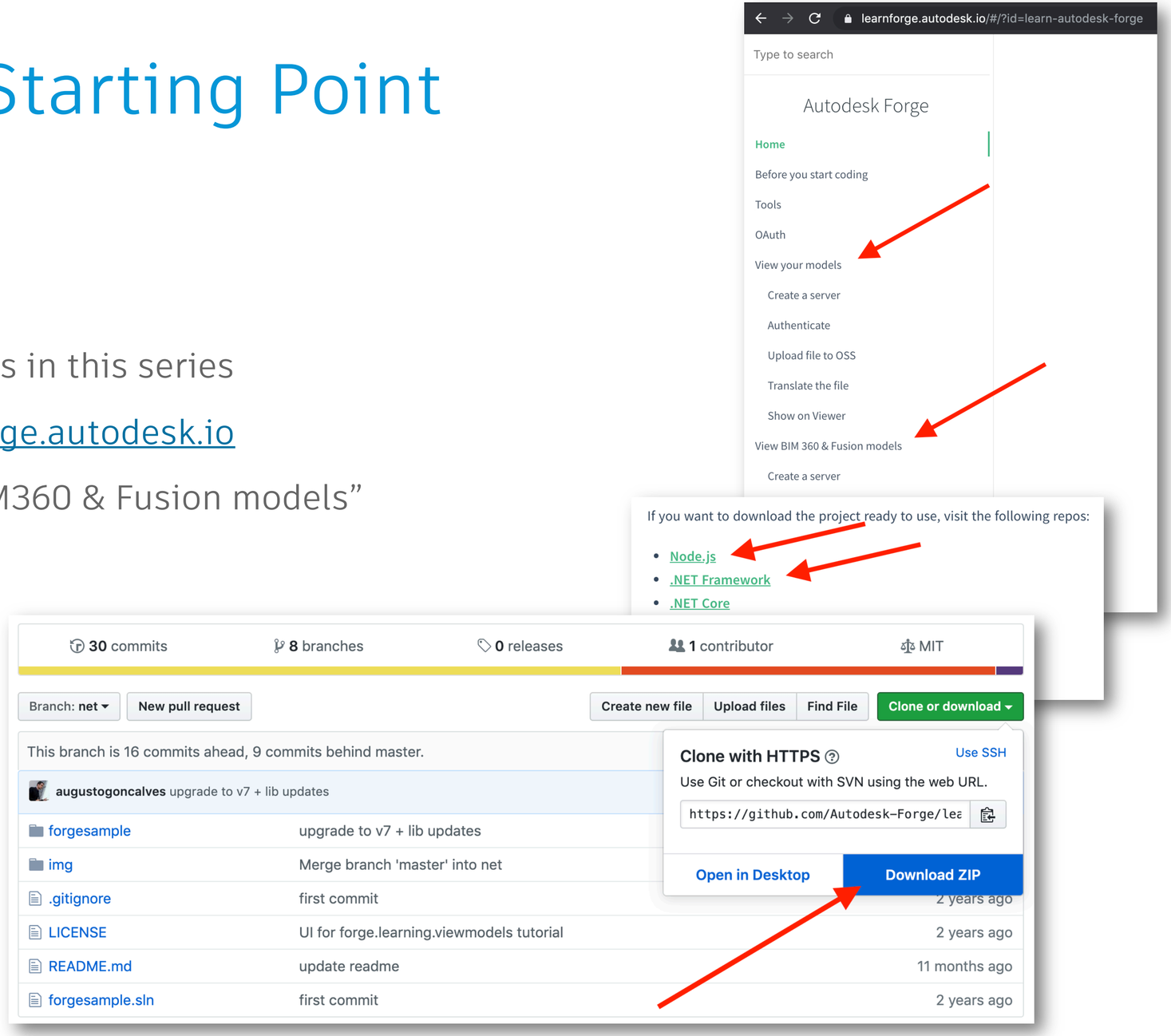
I like to know how everything works under the hood, and I am not afraid of low-level stuff like bits, buffers, pointers, stack, heap, threads, shaders and of course Math. I am interested in 3D for Web, from raw WebGL to libraries and frameworks with different levels of abstractions, as well as how virtual entities from 3D world can be linked to things from real world. In the end, "For an artificial mind, all reality is virtual" [The Animatrix]. To achieve my goals, I like to speak C/C++ mostly with modern dialect, Go, Python, and I have to speak Javascript/Node.js.

Before We Start

Starting Point

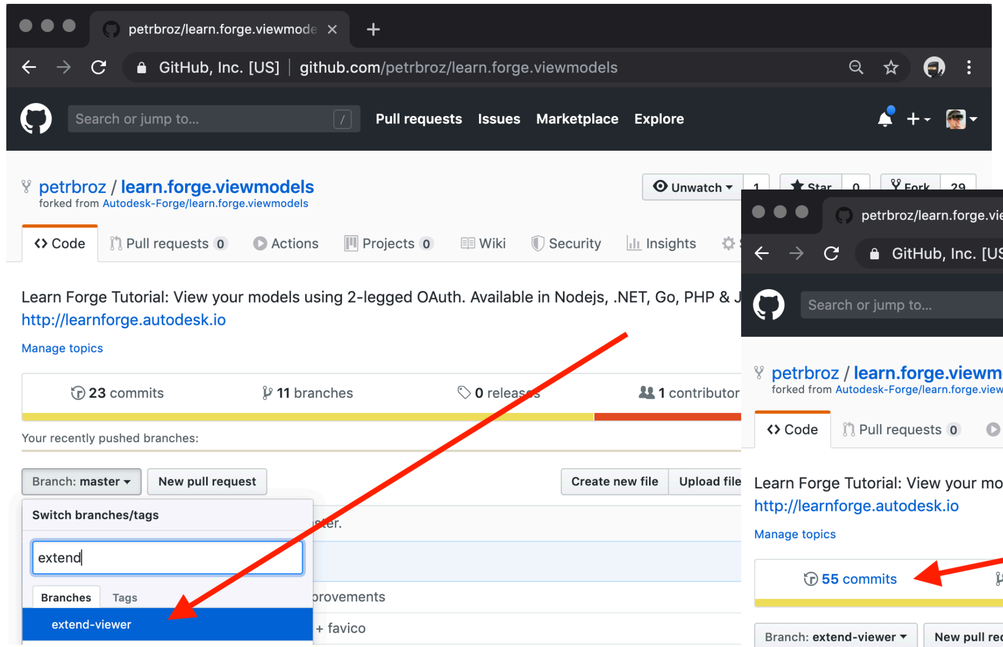
OPTIONS

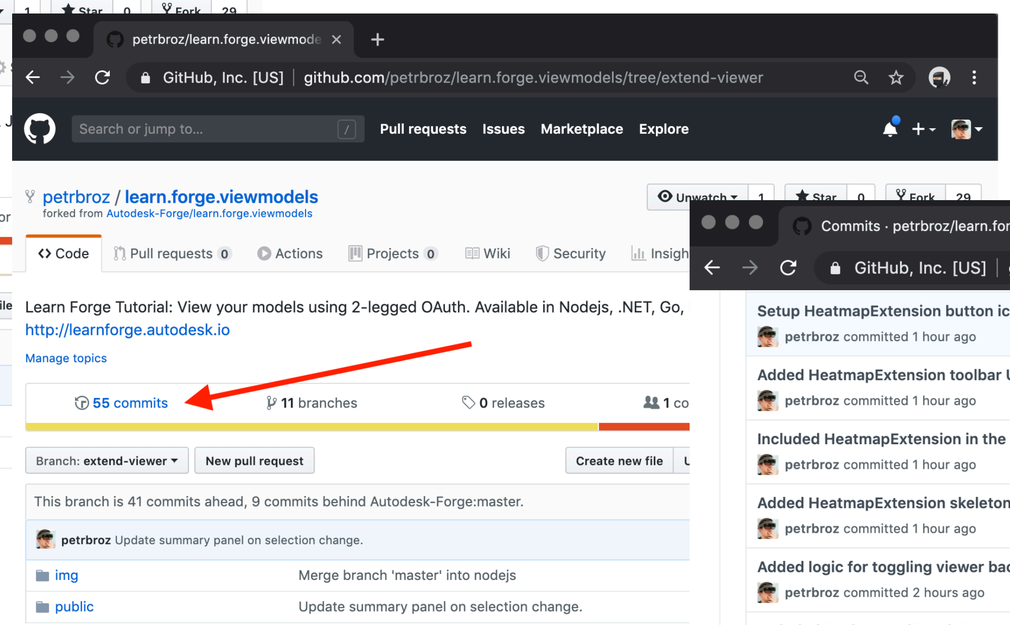
- Use your project from previous labs in this series
- Use a sample from https://learnforge.autodesk.io
  - “View your models” or “View BIM360 & Fusion models”
  - Node.js or .NET
  - Download and extract
  - Follow steps in README

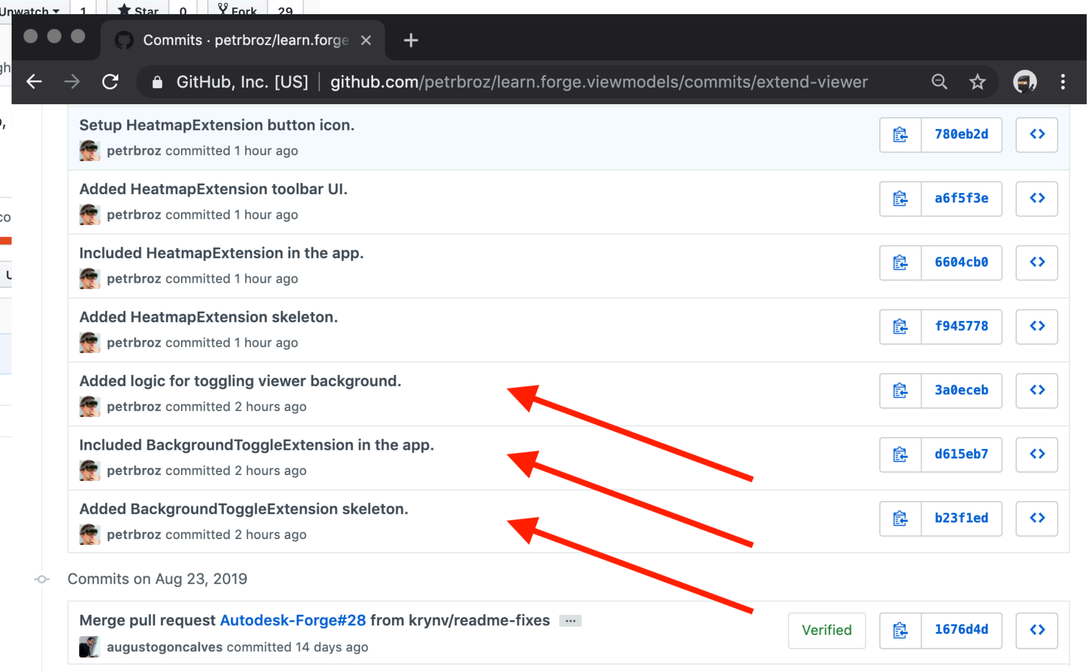


If You Get Lost

- All steps in this presentation are available as git commits
  https://github.com/petrbroz/learn.forge.viewmodels/tree/extend-viewer



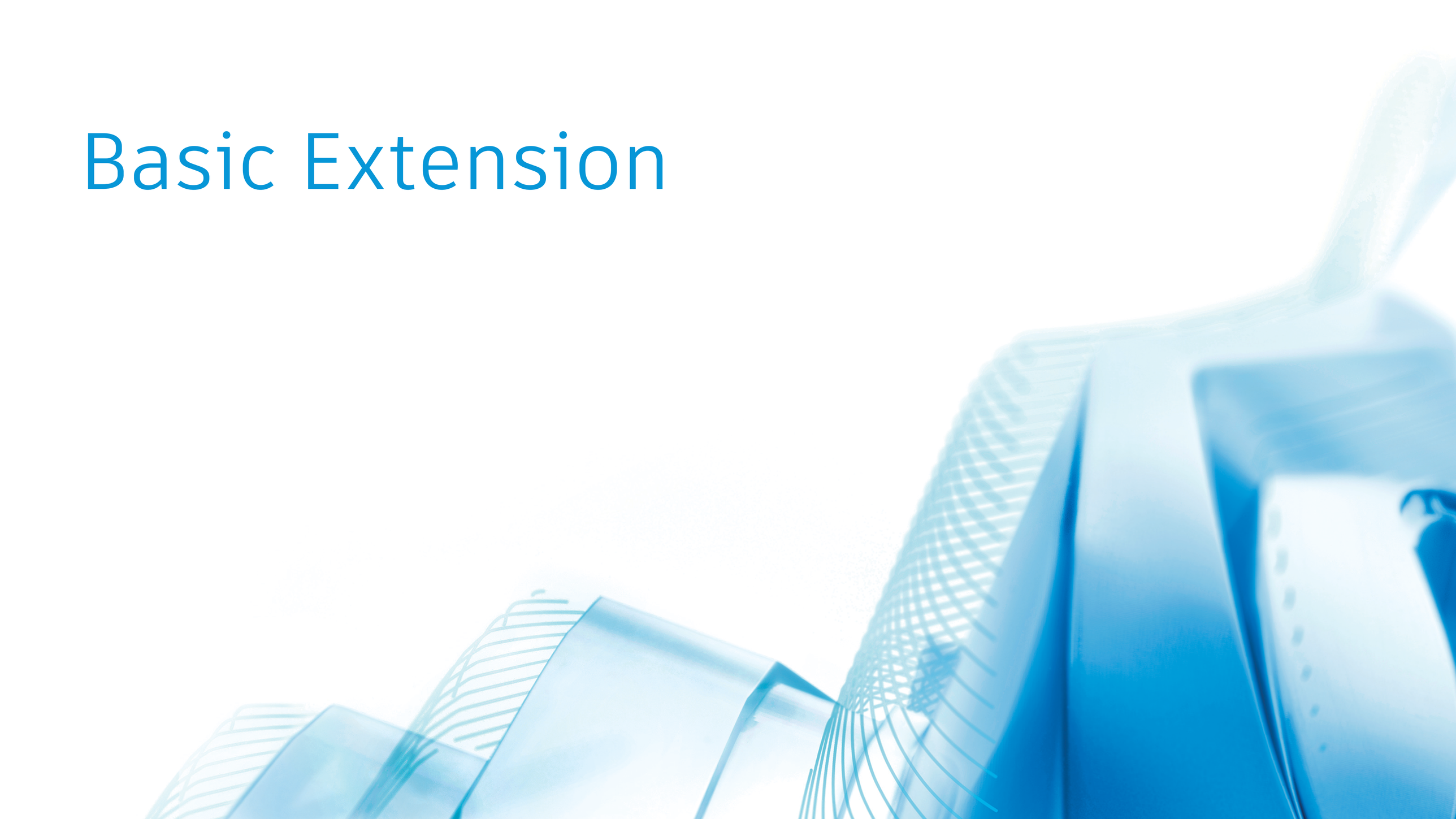




Agenda

- Example #1: Basic Extension
  - No UI
  - Changing viewer settings
- Example #2: Object Tree & Properties
  - Included in toolbar
  - Accessing model scene and properties
- Homework: Custom UI & Events
  - Custom panel
  - Reacting to viewer events

Basic Extension



Extension Skeleton

- Create new file:
  public/js/BackgroundToggleExtension.js
- Add **BackgroundToggleExtension** class extending **Autodesk.Viewing.Extension**
- Add **constructor**
  - Accepting two params:
    - **viewer** (actual instance of **GuiViewer3D**)
    - **options** (viewer configuration)
  - Pass both params to the parent-class
- Add **load** & **unload** methods
  - Called when extension is loaded/unloaded
  - Should return **true** when successful
- Register extension with the extension manager

18 public/js/BackgroundToggleExtension.js

```
@@ -0,0 +1,18 @@
+ class BackgroundToggleExtension extends Autodesk.Viewing.Extension {
+     constructor(viewer, options) {
+         super(viewer, options);
+     }
+
+     load() {
+         console.log('BackgroundToggleExtension loaded.');
+         return true;
+     }
+
+     unload() {
+         console.log('BackgroundToggleExtension unloaded.');
+         return true;
+     }
+ }
+
+ Autodesk.Viewing.theExtensionManager.registerExtension(
+     'BackgroundToggleExtension', BackgroundToggleExtension);
```

Include Extension

- In *public/index.html*:
  - Add **<script>** tag to **<head>** with the new extension script (**js/BackgroundToggleExtension.js**)

- In *public/js/ForgeViewer.js*:
  - Pass config object as 2nd param to **GuiViewer3D**
  - Include list of extension names as **extensions** property in the config

1 public/index.html

```
@@ -18,6 +18,7 @@
18 18      <link href="css/main.css" rel="stylesheet" />
19 19      <script src="js/ForgeTree.js"></script>
20 20      <script src="js/ForgeViewer.js"></script>
   21 +    <script src="js/BackgroundToggleExtension.js"></script>
21 22    </head>
22 23
23 24    <body>
```

5 public/js/ForgeViewer.js

```
@@ -25,7 +25,10 @@ function launchViewer(urn) {
25 25      };
26 26
27 27      Autodesk.Viewing.Initializer(options, () => {
28    -      viewer = new Autodesk.Viewing.GuiViewer3D(document.getElementById('forgeViewer'));
   28 +      const config = {
   29 +        extensions: ['BackgroundToggleExtension']
   30 +      };
   31 +      viewer = new Autodesk.Viewing.GuiViewer3D(document.getElementById('forgeViewer'), config);
29 32        viewer.start();
30 33        var documentId = 'urn:' + urn;
31 34        Autodesk.Viewing.Document.load(documentId, onDocumentLoadSuccess, onDocumentLoadFailure);
```

Background Toggle Logic

- In *public/js/BackgroundToggleExtension.js*:
  - Add **onKeyPress** method
  - Bind **onKeyPress** method to **'keypress'** event
  - If **ev.key** is number:
    - change viewer background:

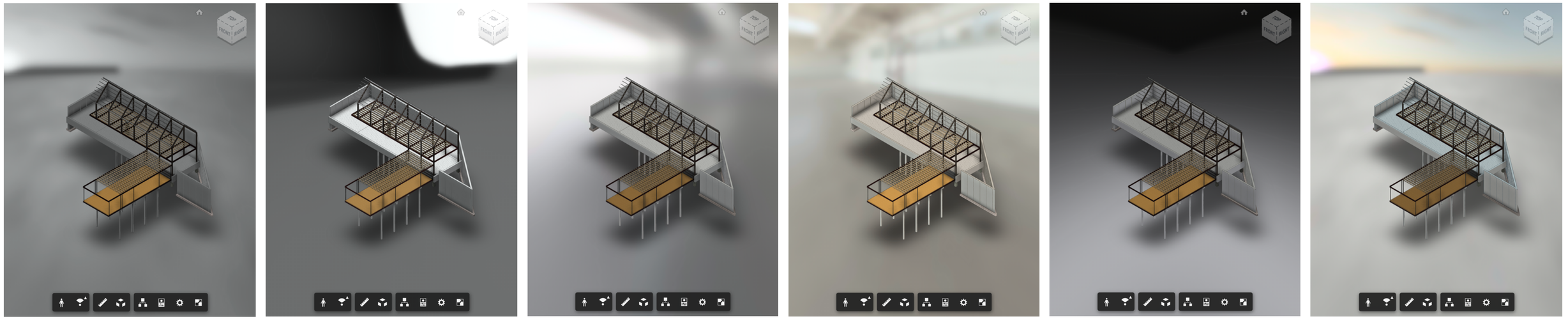
      viewer.setLightPreset(number)

7 public/js/BackgroundToggleExtension.js

```
@@ -4,6 +4,7 @@ class BackgroundToggleExtension extends Autodesk.Viewing.Extension {
4  4          }
5  5
6  6          load() {
   7  +           window.addEventListener('keypress', this.onKeyPress.bind(this));
7  8              console.log('BackgroundToggleExtension loaded.');
8  9              return true;
9  10         }
@@ -12,6 +13,12 @@ class BackgroundToggleExtension extends Autodesk.Viewing.Extension {
12 13             console.log('BackgroundToggleExtension unloaded.');
13 14             return true;
14 15         }
   16 +
   17 +     onKeyPress(ev) {
   18 +         if (/^\d$/.test(ev.key)) {
   19 +             this.viewer.setLightPreset(parseInt(ev.key));
   20 +         }
   21 +     }
15 22     }
16 23
17 24     Autodesk.Viewing.theExtensionManager.registerExtension(
```

Test Extension

- Run the application & open a model
- Toggle between environments using numerical keys



Object Tree & Properties

Extension Skeleton

- Create new file: *public/js/HeatmapExtension.js*
- Add **HeatmapExtension** class extending **Autodesk.Viewing.Extension**
- Same structure as **BackgroundToggleExtension**

18 public/js/HeatmapExtension.js

```
...  ...  @@ -0,0 +1,18 @@
     1  + class HeatmapExtension extends Autodesk.Viewing.Extension {
     2  +     constructor(viewer, options) {
     3  +         super(viewer, options);
     4  +     }
     5  +
     6  +     load() {
     7  +         console.log('HeatmapExtension loaded.');
     8  +         return true;
     9  +     }
    10  +
    11  +     unload() {
    12  +         console.log('HeatmapExtension unloaded.');
    13  +         return true;
    14  +     }
    15  + }
    16  +
    17  + Autodesk.Viewing.theExtensionManager.registerExtension(
    18  +     'HeatmapExtension', HeatmapExtension);
```

Include Extension

- In *public/index.html*:
  - Add **<script>** tag to **<head>** with the new extension script (**js/HeatmapExtension.js**)

- In *public/js/ForgeViewer.js*:
  - Add extension name to viewer config

1 public/index.html

```
@@ -19,6 +19,7 @@
19 19     <script src="js/ForgeTree.js"></script>
20 20     <script src="js/ForgeViewer.js"></script>
21 21     <script src="js/BackgroundToggleExtension.js"></script>
   22 +   <script src="js/HeatmapExtension.js"></script>
22 23   </head>
23 24
24 25   <body>
```

2 public/js/ForgeViewer.js

```
@@ -26,7 +26,7 @@ function launchViewer(urn) {
26 26
27 27     Autodesk.Viewing.Initializer(options, () => {
28 28       const config = {
29    -        extensions: ['BackgroundToggleExtension']
   29 +        extensions: ['BackgroundToggleExtension', 'HeatmapExtension']
30 30       };
31 31       viewer = new Autodesk.Viewing.GuiViewer3D(document.getElementById('forgeViewer'), config);
32 32       viewer.start();
```

Create Toolbar UI

- In *public/js/HeatmapExtension.js*:
  - Add **onToolbarCreated()** method
    - Will be called when viewer toolbar is ready
  - Check existence of control using **viewer.toolbar.getControl(name)**
  - Create new toolbar group
    - **new Autodesk.Viewing.UI.ControlGroup(name)**
    - **viewer.toolbar.addControl(group)**
  - Add new button to toolbar group
    - **new Autodesk.Viewing.UI.Button(name)**
    - **button.setToolTip(tooltip)**
    - **button.addClass(className)**
    - **button.onClick = (event) => {}**
    - **group.addControl(button)**

16 public/js/HeatmapExtension.js

```
@@ -12,6 +12,22 @@ class HeatmapExtension extends Autodesk.Viewing.Extension {
12 12             console.log('HeatmapExtension unloaded.');
13 13             return true;
14 14         }
   15 +
   16 +     onToolbarCreated() {
   17 +         this._group = this.viewer.toolbar.getControl('myToolbarGroup');
   18 +         if (!this._group) {
   19 +             this._group = new Autodesk.Viewing.UI.ControlGroup('myToolbarGroup');
   20 +             this.viewer.toolbar.addControl(this._group);
   21 +         }
   22 +
   23 +         this._button = new Autodesk.Viewing.UI.Button('heatmapButton');
   24 +         this._button.onClick = (ev) => {
   25 +             alert('Hello World!');
   26 +         };
   27 +         this._button.setToolTip('Heatmap');
   28 +         this._button.addClass('heatmapButtonIcon');
   29 +         this._group.addControl(this._button);
   30 +     }
15 31 }
16 32
17 33 Autodesk.Viewing.theExtensionManager.registerExtension(
```

Toolbar Button Icon

- In *public/css/main.css*:
  - Add rule for class defined in previous slide (**.heatmapButtonIcon**)
  - Specify background image URL, size 24px, no-repeat, and centered
  - For example: https://img.icons8.com/office/30/000000/fire-element.png

7 public/css/main.css

```
@@ -21,3 +21,10 @@ body {
   #forgeViewer {
     width: 100%;
   }
+
+ .heatmapButtonIcon {
+   background-image: url(https://img.icons8.com/office/30/000000/fire-element.png);
+   background-size: 24px;
+   background-repeat: no-repeat;
+   background-position: center;
+ }
```

Iterate Objects

- In *public/js/HeatmapExtension.js*:
  - Add **getLeafNodes()** method
  - Retrieve object tree: **viewer.getObjectTree(callback)**
  - Retrieve root node ID: **tree.getRootId()**
  - Enumerate children of specific node: **tree.enumNodeChildren(parentId, callback, recursive)**
  - Count child nodes: **tree.getChildCount(id)**

14 public/js/HeatmapExtension.js

```
@@ -28,6 +28,20 @@ class HeatmapExtension extends Autodesk.Viewing.Extension {
28 28           this._button.addClass('heatmapButtonIcon');
29 29           this._group.addControl(this._button);
30 30       }
   31 +
   32 +     getLeafNodes() {
   33 +         return new Promise((resolve, reject) => {
   34 +             this.viewer.getObjectTree((tree) => {
   35 +                 let dbids = [];
   36 +                 tree.enumNodeChildren(tree.getRootId(), (dbid) => {
   37 +                     if (tree.getChildCount(dbid) === 0) {
   38 +                         dbids.push(dbid);
   39 +                     }
   40 +                 }, true);
   41 +                 resolve(dbids);
   42 +             });
   43 +         });
   44 +     }
31 45   }
32 46
33 47   Autodesk.Viewing.theExtensionManager.registerExtension(
```

Color Objects

- In *public/js/HeatmapExtension.js*:
  - Add **colorNodes()** method
  - List properties of specific objects: **viewer.model.getBulkProperties(ids, filterProps, callback)**
    - 1st callback argument a list of property DB records
    - Each record containing **dbId** and **properties**
  - Set theming color based on chosen property (in our case, “Area”): **viewer.setThemingColor(id, rgbaVector)**

14 public/js/HeatmapExtension.js

```
@@ -42,6 +42,20 @@ class HeatmapExtension extends Autodesk.Viewing.Extension {
42 42                 });
43 43             });
44 44         }
   45 +
   46 +     colorNodes(ids) {
   47 +         const filterProps = ['Area'];
   48 +         const MaxArea = 100.0;
   49 +         this.viewer.model.getBulkProperties(ids, filterProps, (items) => {
   50 +             for (const item of items) {
   51 +                 const areaProp = item.properties[0];
   52 +                 const normalizedArea = Math.min(1.0, parseFloat(areaProp.displayValue) / MaxArea);
   53 +                 const color = new THREE.Color();
   54 +                 color.setHSL(normalizedArea * 0.33, 1.0, 0.5);
   55 +                 this.viewer.setThemingColor(item.dbId, new THREE.Vector4(color.r, color.g, color.b, 0.5));
   56 +             }
   57 +         });
   58 +     }
45 59   }
46 60
47 61   Autodesk.Viewing.theExtensionManager.registerExtension(
```

Trigger on Button Click

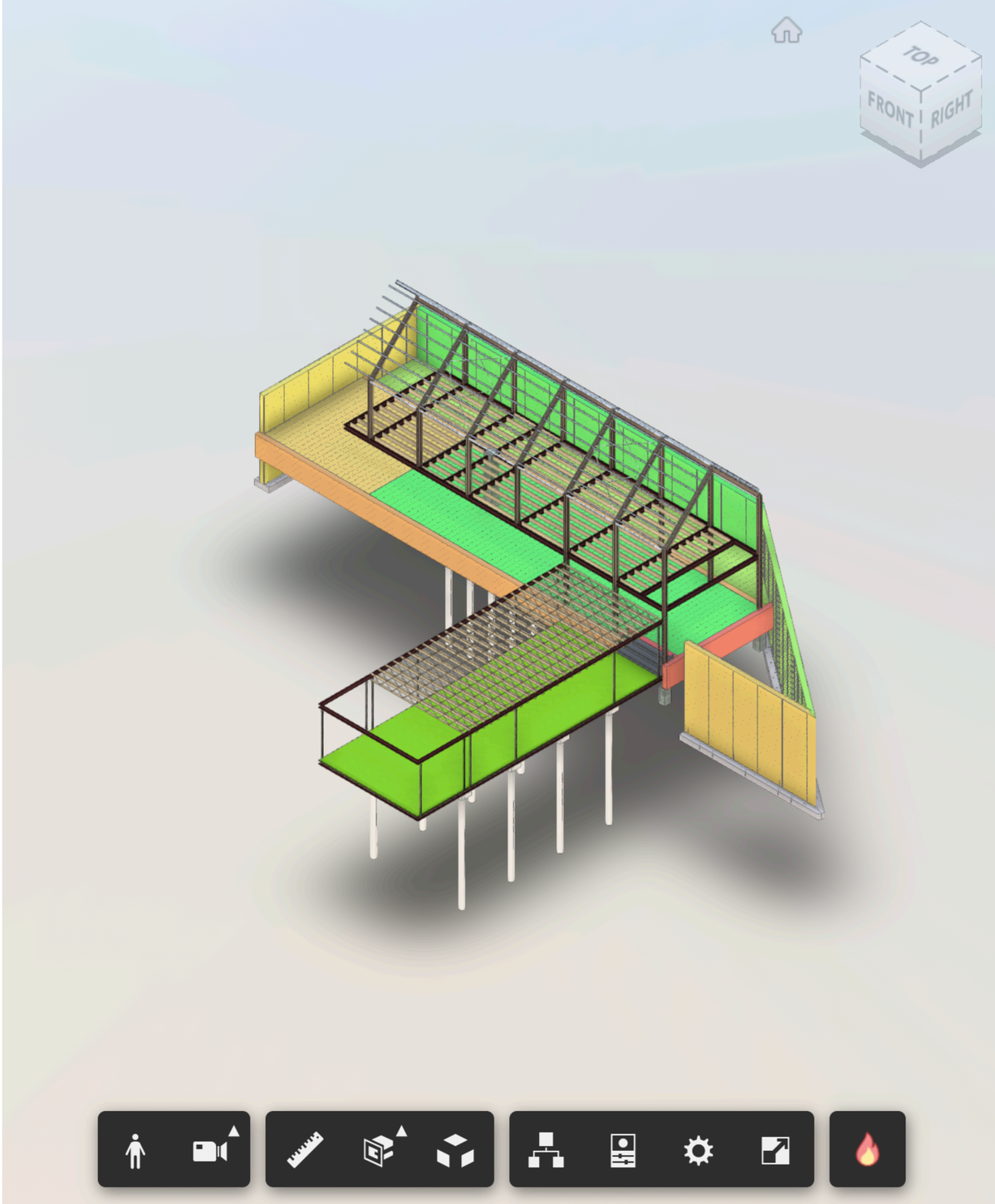
- In *public/js/HeatmapExtension.js*:
  - Update button click handler
  - Toggle **_enabled** state
  - If enabled:
    - Color all leaf nodes
  - If disabled:
    - Clear all colors: **viewer.clearThemingColors()**
  - Don’t forget **async** and **await**

11 public/js/HeatmapExtension.js

```
@@ -20,9 +20,16 @@ class HeatmapExtension extends Autodesk.Viewing.Extension {
20  20              this.viewer.toolbar.addControl(this._group);
21  21          }
22  22
    23  +       this._enabled = false;
23  24          this._button = new Autodesk.Viewing.UI.Button('heatmapButton');
24      -       this._button.onClick = (ev) => {
25      -           alert('Hello World!');
    25  +       this._button.onClick = async (ev) => {
    26  +           this._enabled = !this._enabled;
    27  +           if (this._enabled) {
    28  +               const ids = await this.getLeafNodes();
    29  +               this.colorNodes(ids);
    30  +           } else {
    31  +               this.viewer.clearThemingColors();
    32  +           }
26  33          };
27  34          this._button.setToolTip('Heatmap');
28  35          this._button.addClass('heatmapButtonIcon');
```

Test Extension

- Run the application & open a model
- Click the new toolbar button



You did it!

Resources

- Learning
  - Viewer reference: https://forge.autodesk.com/en/docs/viewer/v7/reference
    - Events: https://forge.autodesk.com/en/docs/viewer/v7/reference/Viewing/#events
    - Viewer3D API: https://forge.autodesk.com/en/docs/viewer/v7/reference/Viewing/Viewer3D
  - Learn Forge tutorial: https://learnforge.autodesk.io

- Questions
  - Stack Overflow (tags “autodesk-forge”, “autodesk-viewer”, “autodesk-model-derivative”, etc.)
  - https://forge.autodesk.com/en/support/get-help
  - forge.help@autodesk.com

Come code with us!

- Forge Accelerator events in 2020: http://autodeskcloudaccelerator.com/
  - Jan 13 – Jan 17 – San Francisco
  - Feb 3 – Feb 7 – Bogota
  - Feb 17 – Feb 21 – London
  - Mar 9 – Mar 13 – Sydney
  - Apr 6 – Apr 10 – São Paolo
  - Apr 27 – May 1 – Boston
  - Jun 1 – Jun 5 – Denver
  - Jun 8 – Jun 12 – Barcelona

AUTODESK®

Make anything™

Homework: Custom UI & Events

Extension Skeleton

- Create new file: *public/js/SummaryExtension.js*
- Add **SummaryExtension** class extending **Autodesk.Viewing.Extension**
- Same structure as before

18 public/js/SummaryExtension.js

```
@@ -0,0 +1,18 @@
+ class SummaryExtension extends Autodesk.Viewing.Extension {
+     constructor(viewer, options) {
+         super(viewer, options);
+     }
+
+     load() {
+         console.log('SummaryExtension loaded.');
+         return true;
+     }
+
+     unload() {
+         console.log('SummaryExtension unloaded.');
+         return true;
+     }
+ }
+
+ Autodesk.Viewing.theExtensionManager.registerExtension(
+     'SummaryExtension', SummaryExtension);
```

Include Extension

- In *public/index.html*:
  - Add **<script>** tag to **<head>** with the new extension script (**js/SummaryExtension.js**)

- In *public/js/ForgeViewer.js*:
  - Add extension name to viewer config

1 public/index.html

```
@@ -20,6 +20,7 @@
20 20     <script src="js/ForgeViewer.js"></script>
21 21     <script src="js/BackgroundToggleExtension.js"></script>
22 22     <script src="js/HeatmapExtension.js"></script>
   23 +   <script src="js/SummaryExtension.js"></script>
23 24   </head>
24 25
25 26   <body>
```

2 public/js/ForgeViewer.js

```
@@ -26,7 +26,7 @@ function launchViewer(urn) {
26 26
27 27     Autodesk.Viewing.Initializer(options, () => {
28 28       const config = {
29    -       extensions: ['BackgroundToggleExtension', 'HeatmapExtension']
   29 +       extensions: ['BackgroundToggleExtension', 'HeatmapExtension', 'SummaryExtension']
30 30       };
31 31       viewer = new Autodesk.Viewing.GuiViewer3D(document.getElementById('forgeViewer'), config);
32 32       viewer.start();
```

Create Toolbar UI

- In *public/js/SummaryExtension.js*:
  - Add **onToolbarCreated()** method
  - Add new button
  - Same structure as before

16 public/js/SummaryExtension.js

```
@@ -12,6 +12,22 @@ class SummaryExtension extends Autodesk.Viewing.Extension {
            console.log('SummaryExtension unloaded.');
            return true;
        }
+
+    onToolbarCreated() {
+        this._group = this.viewer.toolbar.getControl('myToolbarGroup');
+        if (!this._group) {
+            this._group = new Autodesk.Viewing.UI.ControlGroup('myToolbarGroup');
+            this.viewer.toolbar.addControl(this._group);
+        }
+
+        this._button = new Autodesk.Viewing.UI.Button('summaryButton');
+        this._button.onClick = async (ev) => {
+            alert('Hello World!');
+        };
+        this._button.setToolTip('Summary');
+        this._button.addClass('summaryButtonIcon');
+        this._group.addControl(this._button);
+    }
}

Autodesk.Viewing.theExtensionManager.registerExtension(
```

Toolbar Button Icon

- In *public/css/main.css*:
  - Add rule for class defined in previous slide (**.summaryButtonIcon**)
  - Specify background image URL, size 24px, no-repeat, and centered

7 ■■■■■ public/css/main.css

```
@@ -28,3 +28,10 @@ body {
28  28      background-repeat: no-repeat;
29  29      background-position: center;
30  30    }
    31  +
    32  + .summaryButtonIcon {
    33  +   background-image: url(https://img.icons8.com/color/48/000000/calculator.png);
    34  +   background-size: 24px;
    35  +   background-repeat: no-repeat;
    36  +   background-position: center;
    37  + }
```

New Panel Class

- In *public/js/SummaryExtension.js*:
  - Add new **SummaryPanel** class extending **Autodesk.Viewing.UI.PropertyPanel**
  - Add constructor with params:
    - **container** (HMTL container for the panel)
    - **id** (unique ID of the panel)
    - **title** (panel title)
    - **options** (additional options)

6 public/js/SummaryExtension.js

```
@@ -30,5 +30,11 @@ class SummaryExtension extends Autodesk.Viewing.Extension {
30  30          }
31  31      }
32  32
    33  +   class SummaryPanel extends Autodesk.Viewing.UI.PropertyPanel {
    34  +       constructor(container, id, title, options) {
    35  +           super(container, id, title, options);
    36  +       }
    37  +   }
    38  +
33  39      Autodesk.Viewing.theExtensionManager.registerExtension(
34  40          'SummaryExtension', SummaryExtension);
```

Show Panel on Button Click

- In *public/js/SummaryExtension.js*:
  - Create panel when extension is loaded
  - Update the button click handler
    - Toggle panel visibility:
      panel.setVisible(bool), **panel.isVisible()**

3 public/js/SummaryExtension.js

```
@@ -4,6 +4,7 @@ class SummaryExtension extends Autodesk.Viewing.Extension {
4  4          }
5  5
6  6          load() {
   7  +           this._panel = new SummaryPanel(this.viewer.container, 'summaryPanel', 'Summary Panel');
7  8           console.log('SummaryExtension loaded.');
8  9           return true;
9  10         }
@@ -22,7 +23,7 @@ class SummaryExtension extends Autodesk.Viewing.Extension {
22 23
23 24          this._button = new Autodesk.Viewing.UI.Button('summaryButton');
24 25          this._button.onClick = async (ev) => {
25    -             alert('Hello World!');
   26 +             this._panel.setVisible(!this._panel.isVisible());
26 27          };
27 28          this._button.setToolTip('Summary');
28 29          this._button.addClass('summaryButtonIcon');
```

Update Panel Content

- In *public/js/SummaryExtension.js*:
  - Add **onSelectionChanged()** method
  - Bind the method to **Autodesk.Viewing.SELECTION_CHANGED_EVENT**
  - When selection changes:
    - Get IDs of selected objects: **viewer.getSelection()**
    - Get properties of specific objects: **viewer.model.getBulkProperties(ids, filterProps, callback)**
    - Clear panel: **panel.removeAllProperties()**
    - Add new properties with computed results: **panel.addProperty(propName, propVal, propCategory)**

29 public/js/SummaryExtension.js

```
@@ -5,6 +5,8 @@ class SummaryExtension extends Autodesk.Viewing.Extension {
5  5
6  6       load() {
7  7           this._panel = new SummaryPanel(this.viewer.container, 'summaryPanel', 'Summary Panel');
   8 +         this.viewer.addEventListener(Autodesk.Viewing.SELECTION_CHANGED_EVENT,
   9 +             this.onSelectionChanged.bind(this));
8  10          console.log('SummaryExtension loaded.');
9  11          return true;
10 12      }
@@ -29,6 +31,33 @@ class SummaryExtension extends Autodesk.Viewing.Extension {
29 31          this._button.addClass('summaryButtonIcon');
30 32          this._group.addControl(this._button);
31 33      }
   34 +
   35 +     onSelectionChanged() {
   36 +         const selectedIds = this.viewer.getSelection();
   37 +         if (selectedIds.length > 0) {
   38 +             const filterProps = ['Area', 'Volume'];
   39 +             this.viewer.model.getBulkProperties(selectedIds, filterProps, (items) => {
   40 +                 let totalArea = 0.0, totalVolume = 0.0;
   41 +                 for (const item of items) {
   42 +                     const areaProp = item.properties.find(prop => prop.displayName === 'Area');
   43 +                     if (areaProp) {
   44 +                         totalArea += parseFloat(areaProp.displayValue);
   45 +                     }
   46 +                     const volumeProp = item.properties.find(prop => prop.displayName === 'Volume');
   47 +                     if (volumeProp) {
   48 +                         totalVolume += parseFloat(volumeProp.displayValue);
   49 +                     }
   50 +                 }
   51 +                 this._panel.removeAllProperties();
   52 +                 this._panel.addProperty('Total Area', totalArea.toFixed(2), 'Measurements');
   53 +                 this._panel.addProperty('Total Volume', totalVolume.toFixed(2), 'Measurements');
   54 +             });
   55 +         } else {
   56 +             this._panel.removeAllProperties();
   57 +             this._panel.addProperty('Total Area', '0.00', 'Measurements');
   58 +             this._panel.addProperty('Total Volume', '0.00', 'Measurements');
   59 +         }
   60 +     }
32 61  }
33 62
34 63  class SummaryPanel extends Autodesk.Viewing.UI.PropertyPanel {
```

Test Extension

- Run the application & open a model
- Click the new toolbar button
- Select multiple objects (shift+click)