

X3DOM

Getting declarative (X)3D into HTML



WebGL Camp2

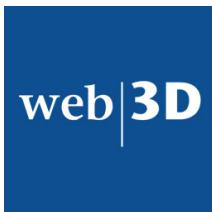
Johannes Behr & Yvonne Jung

Fraunhofer IGD

johannes.behr@igd.fraunhofer.de



Fraunhofer
IGD



X3DOM Project



X3DOM (pronounced X-Freedom) is an experimental open source framework and runtime to support the **ongoing discussion** in the **Web3D and W3C communities** how an integration of **HTML5** and **declarative 3D** content could look like.

Ok fine. But how relates this to WebGL?

Declarative (X)3D in HTML

Completes todays graphics technologies



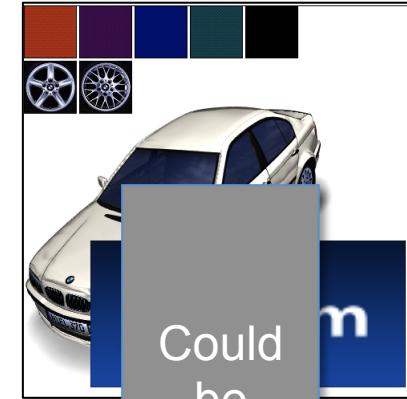
Declarative

Scenegraph
Part of HTML-document
DOM Integration
CSS/ Events

2D
(Final HTML5 spec)

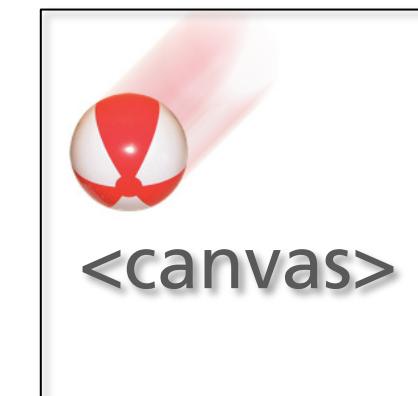


3D
(No W3C spec yet)



Imperative

Procedural API
Drawing context



Declarative (X)3D in HTML

Powerful abstraction for Web-apps



Efficiency: Scene interaction, picking, culling and rendering can be implemented in native code. Utilizes all (battery) resources as efficient as possible.

Concepts: Provides HTML/DOM concepts for application developer. Does not force HTML Developer to deal with GLSL and 4x4 matrices.

Adaptability: Declarative material abstraction allows shading adoption per client hardware (shader, raytracer, ...)

Flexibility: Dynamic scene-partitioning supports client and server based rendering

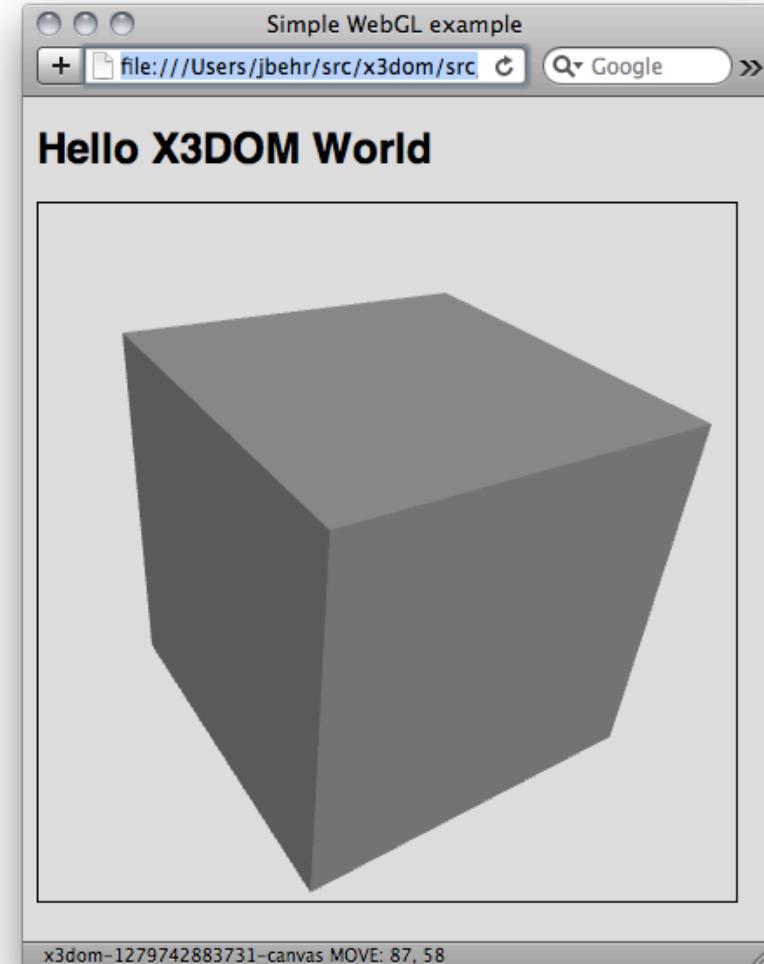
Metadata: How to index and search “content” in WebGL-Apps?

Declarative (X)3D in HTML

Embed a live scenegraph in the DOM



```
<!DOCTYPE html >
<html >
  <body>
    <h1>Hello X3DOM World</h1>
    <x3d xmlns='...' profile='HTML' >
      <scene>
        <shape>
          <box></box>
        </shape>
      </scene>
    </x3d>
  </body>
</html>
```



History of declarative (X)3D in HTML



X3D: ISO Standard since 2004; Plugin integration model

HTML5 Specification: <http://www.w3.org/TR/html5/no.html#declarative-3d-scenes>

13.2 **Declarative 3D scenes**

*Embedding 3D imagery into XHTML documents is the domain of X3D, or technologies based on **X3D** that are namespace aware.*

2007: First experiment by Philip Taylor, W3C: <http://philip.html5.org>

X3D in DOM, no DOM manipulation (Canvas3D for rendering)

2009: x3dom by Fraunhofer IGD, Based on code of Taylor: www.x3dom.org

Full DOM integration. Native, X3D-Plugin or WebGL for rendering

W3C TPAC 2009 in Santa Clara, USA

<http://web3d.org/x3d/presentations/X3D+HTML5.W3cTpac-20091106.pdf>

2010: HTML/X3D InterestGroup / Web3D consortium

Develops HTML/X3D integration model based on x3dom

W3C TPAC 2010 in Lyon, France

2011: W3C “Declarative 3D” Incubator Group

Should produce real world requirements for “Declarative 3D” in HTML

X3D Standard

State of the current integration model



Pro:

ISO Standard developed by non-profit Web3D Consortium

Specification includes abstract **IDL** and **JavaScript-binding**

10 year W3C member and contributor

Event and runtime model build for the Web

Con:

Plugin integration model

Simple Shape/Material assign mechanism. **No cascading** Material system

Full Standard is **to complex**, Some concepts duplicate W3C techniques

HTML/DOM Profile

Reduce X3D to 3D visualization component for HTML5



General Goal:

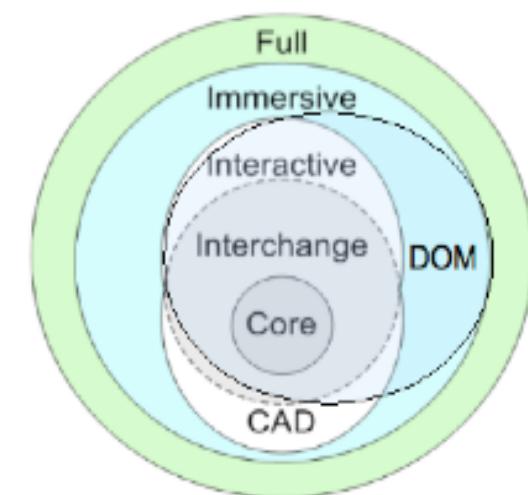
Utilizes **HTML/JS/CSS** for **scripting** and **interaction**
Reduced complexity and **implementation effort**

“**HTML**”-Profile (Extends X3D “Interchange”):

Full **runtime** with anim., navigation and
asynchronous data fetching

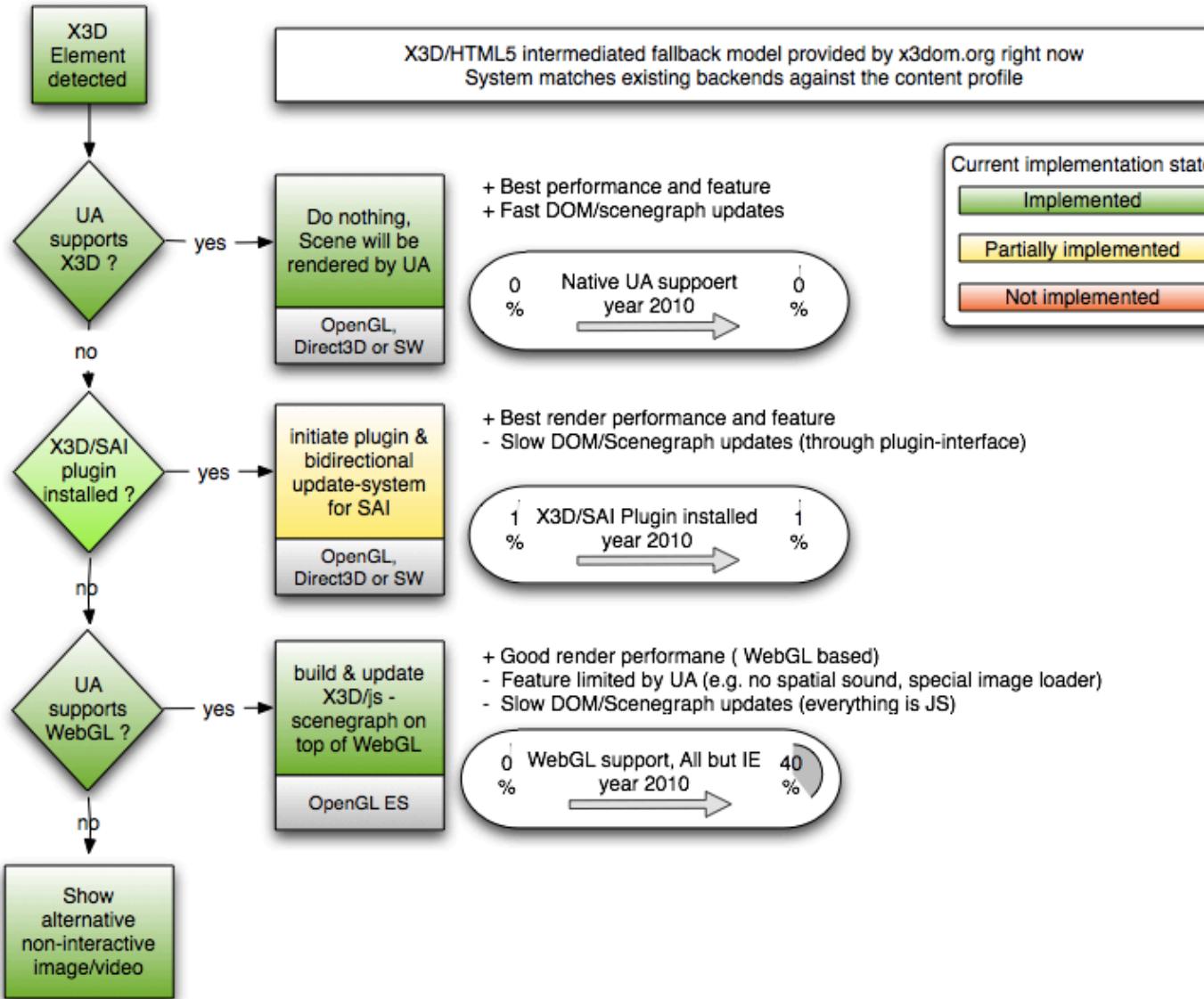
No X3D-Script, Proto, High-Level Sensors

Declarative and explicit shader material



Implementation

JS-Layer: Supports **native**, **X3D/SAI-Plugin** or **WebGL**



x3dom.org/x3dom/release/x3dom.js

JavaScript-based layer



```
<!DOCTYPE html >
<html >
  <head>
    <link rel="stylesheet" type="text/css" href="x3dom.css" >
    <script type="text/javascript" src="x3dom.js"></script>
  </head>
  <body>
    <h1>HTML5 Hello World</h1>
    <x3d xmlns="..." profile='...' backend='...' >
      <scene>
        ...
      </scene>
    </x3d>
  </body>
</html>
```

HTML Profile (X3D subset)

Basic scene-graph nodes as DOM elements



group, transform, lod, switch, billboard

inline (loads additional parts asynchronously)

viewpoint (standard perspective camera)

Single and Multi-index **Geometry** types with support for **vertex properties**

pointLight, spotLight and **directionalLight**

appearance with flexible shader-selection

technique for explicit and parametric shader (e.g. CommonSurfaceShader proposal)

text

sound (spatialized emitter)

background (skybox background)

follower and interpolator for animation

Standard **navigation** (Walk, Fly, Examine, ...)

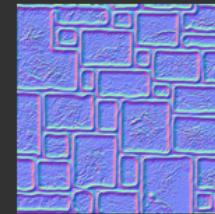
Direct support for HTML5 media elements

texture: , <video> and <canvas>

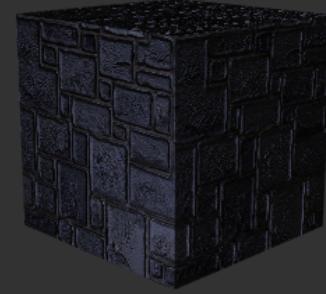
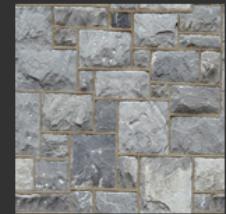
sound: <audio>

Common Surface Shader

Normal Map



Diffuse Map



DOM Manipulation

Node appending and removal



HTML/X3D code:

```
...
<group id='root'></group>
```

HTML-Script to add nodes:

```
trans = document.createElement('Transform');
trans.setAttribute('translation', '1 2 3' );
document.getElementById('root').appendChild(trans);
```

HTML-Script to remove nodes:

```
document.getElementById('root').removeChild(trans);
```

Application show-case

Facebook Friendgraph in 3D



DOM Manipulation

Field updates with setAttribute() or SAI-Field interfaces



HTML/X3D code:

```
...
<material id='mat'></material>
...
<coordinate id='coord' point='5.6 3 87, 8.8 8.4 3.2, ...' ></coordinate>
...
...
```

Generic HTML-Script with **setAttribute()**: also useful for libs like **jQuery**

```
document.getElementById('mat').setAttribute('diffuseColor','red');
```

HTML-Script using SAI-Field interface: X3D JS-binding for more efficiency

```
var saiField = document.getElementById('coord').getField('point');
saiField[4711].x = 0.815;
```

Application show-case

Application specific navigation: Carousel menu



HTML Events

User Interaction through DOM Events



Supports interaction with standard HTML-Events. Supports **ancient** (Netscape2) and **addEventListener()** interfaces.

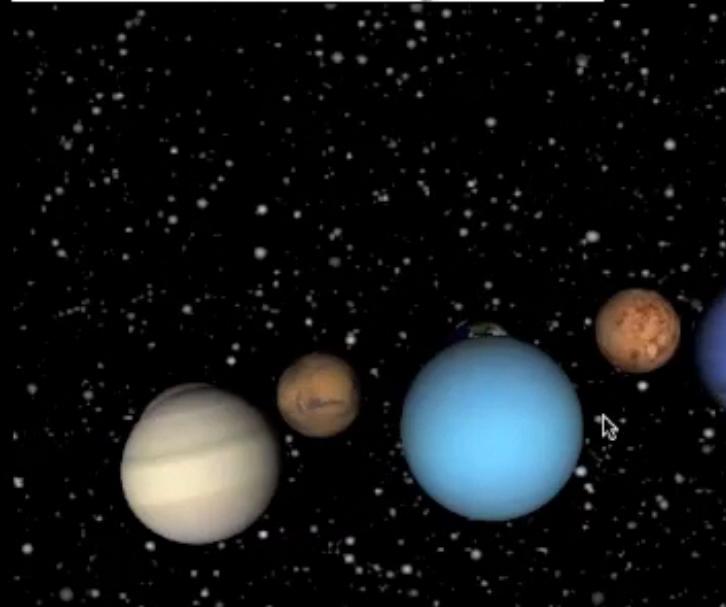
```
<x3d xmlns="...">
<Scene>
  <Shape>
    <Appearance>
      <Material id='mat' diffuseColor='red' />
    </Appearance>
    <Box onclick="document.getElementById('mat').diffuseColor='green'" />
  </Shape>
</Scene>
</x3d>
```

Application show-case

Application specific navigation: Carousel menu



Planets of the solar system



Carousel menu: the virtual planetarium

A simple effect, driven by a powerful engine: With X3DOM an JavaScript you can run and develop beautiful effects with strong integration of 2D and 3D elements.

3D Navigation through 2D "mousemove" event

Moving the mouse over the scene causes the system to rotate. A standard mousemove event on the x3d element is used to get the mouse pointer's position values to calculate the planet's rotation.

Pick 3D objects using a standard "click" event

Clicking on a planet reveals additional information. Likewise, JavaScript's click event applied upon a planet's geometry is used as a trigger.

HTML Events

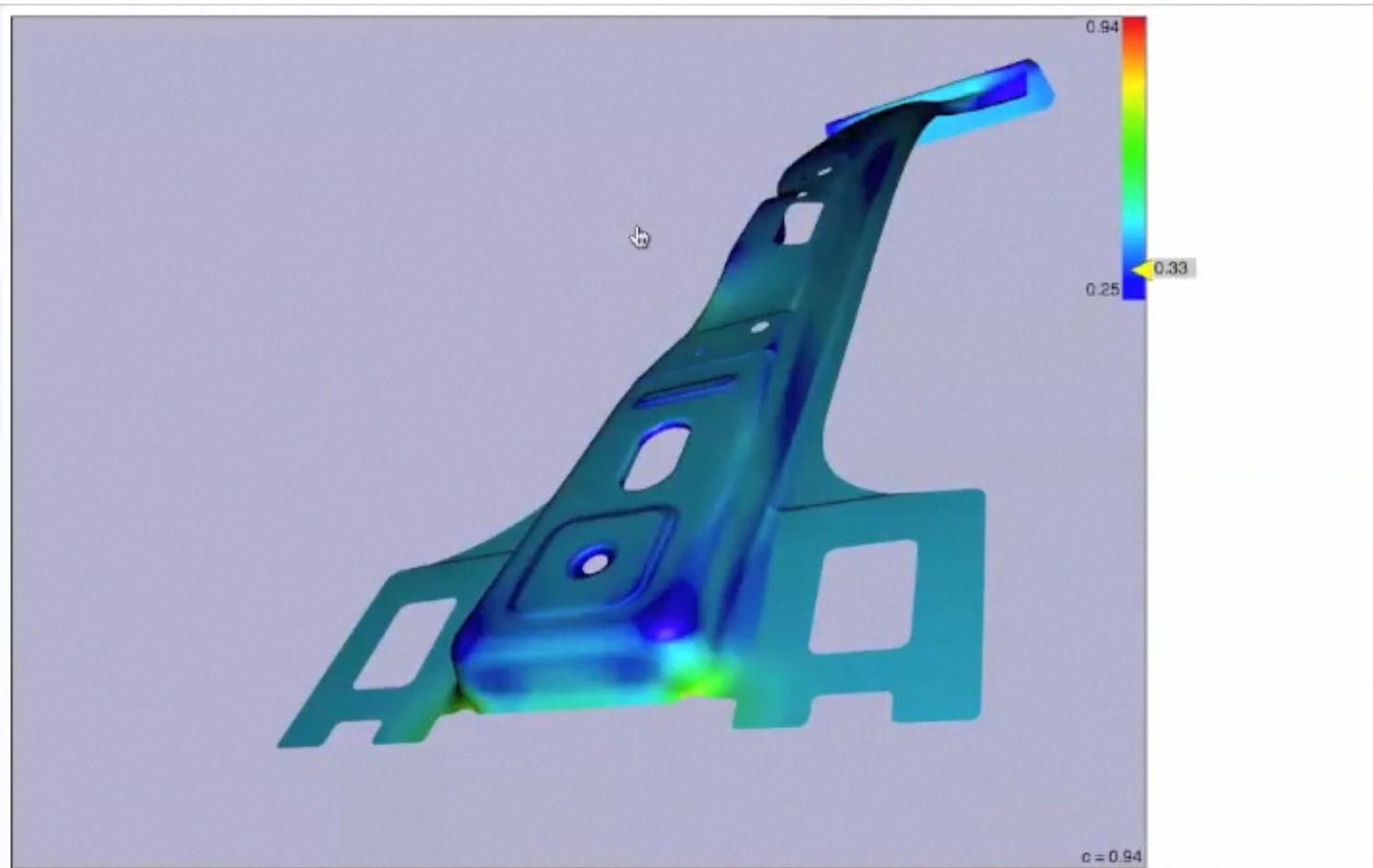
3DPickEvent extends DOM Level 3 MouseEvent



```
interface 3DPickEvent : MouseEvent {  
    readonly attribute float worldX;           // 3d world coordinates  
    readonly attribute float worldY;  
    readonly attribute float worldZ;  
    readonly attribute float localX;           // element local 3d coordinates  
    readonly attribute float localY;  
    readonly attribute float localZ;  
    readonly attribute float normalX;          // surface normal  
    readonly attribute float normalY;  
    readonly attribute float normalZ;  
    readonly attribute float colorRed;          // surface color (with alpha)  
    readonly attribute float colorGreen;  
    readonly attribute float colorBlue;  
    readonly attribute float colorAlpha;  
    readonly attribute float texCoordS;          // surface texture coordinates  
    readonly attribute float texCoordT;  
    readonly attribute float texCoordR;  
    object               getMeshPickData (in DOMString vertexProperty);  
};
```

Application show-case

CAE: Data visualization for Volkswagen, IFX



Choose transferfunction window with slider:

Choose scalar threshold c with slider:

HTML Events



Additional 3D Events extend Dom Level 3 UIEvent

Inspired by the X3D-Sensors Environment Sensors

// Visibility of sub-trees

```
interface 3DVisibilityEvent : UIEvent {  
    ...;  
}
```

// Object-Camera Transformation sensor

```
interface 3DProximityEvent : UIEvent {  
    ...;  
}
```

// Object-Transformation

```
Interface 3DTransformEvent : Event {  
    ...;  
}
```

Application show-case

Thematic data on climate change in cities



DOM Manipulation

CSS 3D Transforms



CSS 3D Transforms Module Level 3; W3C Draft

Utilized to transform and update **<transform>** nodes

```
<style type="text/css">
  #trans {
    -webkit-animation: spin 8s infinite linear;
  }
  @-webkit-keyframes spin {
    from { -webkit-transform: rotateY(0); }
    to   { -webkit-transform: rotateY(-360deg); }
  }
</style>
...
<transform id="trans" >
  <transform style="-webkit-transform: rotateY(45deg);">
  ...

```

Application show-case

Geometry and animation export



HTML Events

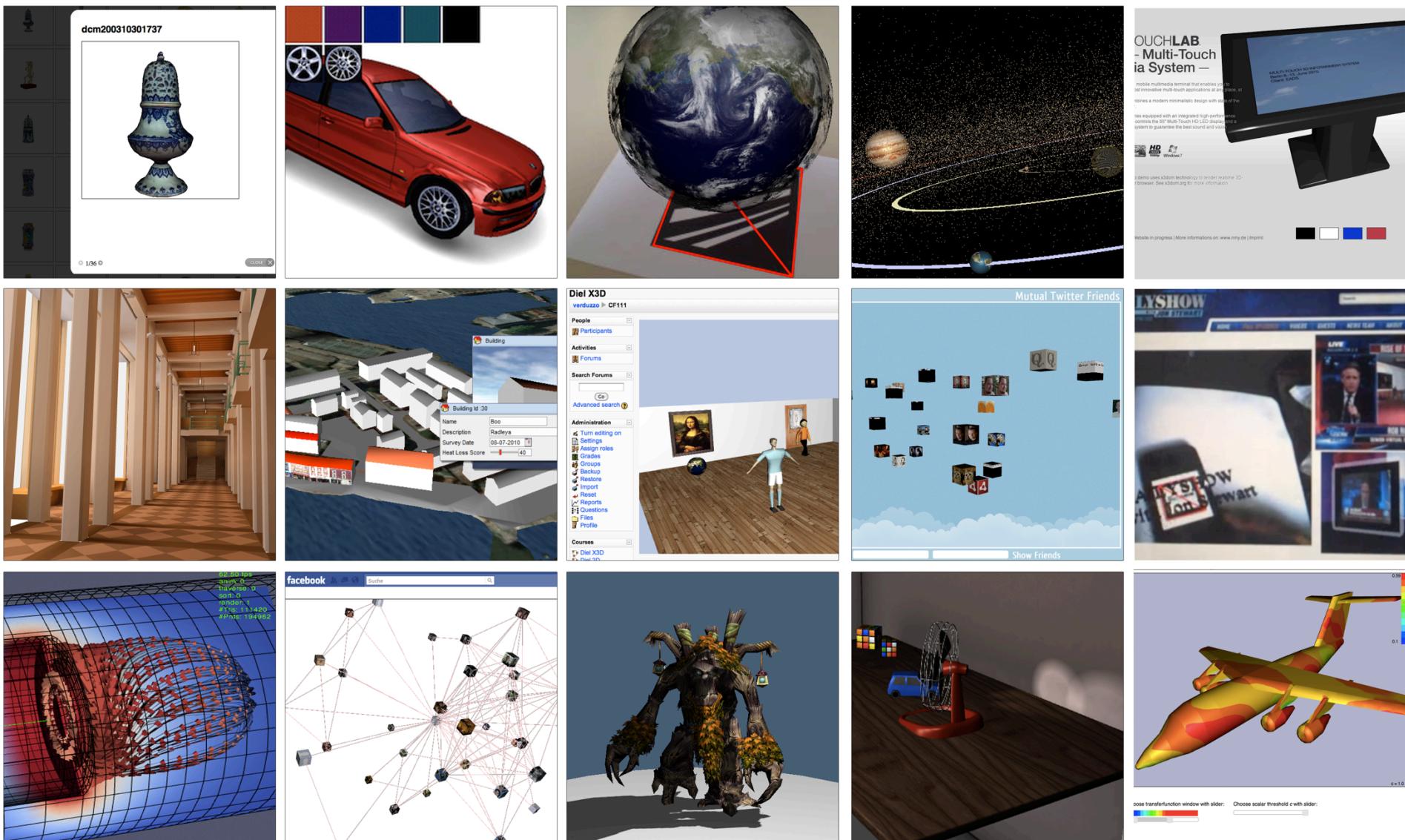
Events from the X3D subsystem



```
<x3d xmlns="http://www.web3d.org/specifications/x3d-3.0.xsd">
  <Viewpoint id='cam01' />
  <Viewpoint id='cam02' />
  <Scene>
    <Shape><Box size="4 4 4" /></Shape>
  </Scene>
</x3d>
<script type="text/javascript">
  var cam = document.getElementById('cam01');
  cam.addEventListener("active",
    function() { alert("Viewpoint 01 is active!"); }, false);
</script>
```

Show-cases and Third-party application

Developed during the last 12 month, (TPAC 2009 – 2010)



Project Status



Current release 1.1: October 22nd, 2010

Unified HTML/XHTML encoding (with and without namespace)
HTML5 <canvas>, and <video> as texture element supported
Better support for CSS 3D Transforms
multiple lights and support for Spot-, Point- and DirectionalLight
Fog
Support for large meshes
Improved normal generation
Follower component

Next Release 2.0: January, 2011 (After WebGL 1.0 spec release)

Internal Shader composition framework -> CommonSurfaceShader
Complete set of HTML Events on X3D-element
CSS support for X3D element
Better Navigation handler, Support for all standard navigation types
Automatic camera path animation



**Thanks!
Questions?**

www.x3dom.org