

AR/VR Workshop

Part 4: **A-Frame components**

Phillip G. Bradford
University of Connecticut

Learning plan

Introduction	Overview – outline goals Setting up google cardboard with glitch.com system
A-frame basics	Simple 3D a-frame examples Work with glitch.com
Foundations	JavaScript, DOM, events, Web-Components
A-frame components	Defining A-frame components
Three.JS and A-frame	Basics of Three.js for A-frame
Entity component architecture (ECA)	Three.js and ECA with standard OO paradigm
A-frame and planets	Complex 3D a-frame, work with complex a-frame detail and basic planetary math; illustrate ECA, geometries, controls, etc.
A-frame and animations	Goal: show how to do basic animation
Conclusion	Goal: review our learning

Outline

Cameras

Parts of A-frame components

Defining a new a-frame component

Cameras

Default

```
<a-entity camera="active: true"  
  look-controls wasd-controls  
  position="0 1.6 0"  
  data-aframe-default-camera></a-entity>
```

Basic of an a-frame component

- 1.Name:** a unique name
- 2.Schema:** Defines the data or properties. It sets the types, default values, and other characteristics of the data
- 3.Init:** This function is called when a component initialized.
- 4.Update:** Called when the component's data changes
- 5.Tick:** Called on every render loop or "frame" of the scene
- 6.Remove:** Called when the component is removed
- 7.Pause & Play:** Called when the scene or entity is paused or played

Component schema

```
AFRAME.registerComponent('my-box', {  
  schema: {  
    width: {type: 'number', default: 1},  
    height: {type: 'number', default: 1},  
    depth: {type: 'number', default: 1},  
    color: {type: 'color', default: '#AAA'}  
  }  
});
```

A component

```
AFRAME.registerComponent('mmap', {
```

```
  init: function() {  
    console.log("HELLO");  
  },
```

```
  tick: function() {  
    console.log("click");  
  },
```

```
  remove: function() {  
    console.log("Goodnight moon!");  
  }
```

```
});
```

Methods in action – put `<script>` in `<body>` For `AFRAME.registerComponent`

```
<body>
  <script>
    AFRAME.registerComponent('say-hello', {
      init: function() {
        console.log("HELLO");
      },
      tick: function() {
        console.log("click");
      } });
  </script>
  <a-scene>
    <a-entity say-hello></a-entity>
  </a-scene>
</body>
```


Another Example

```
<body>
  <script>

    AFRAME.registerComponent('simple-comp', {
      init: function () { console.log("Yes: simple-comp"); } });
    AFRAME.registerComponent('other-comp', {
      schema: {
        message: {type: 'string',default: "Good day!" }
      },
      init: function () { console.log(this.data.message); },
      multiple: true });
  </script>
</body>
```

Another Example, cont

```
<a-scene>  
  <a-entity simple-comp></a-entity>  
  <a-entity other-comp></a-entity>  
  <a-entity other-comp="message: Hello, Metaverse!"></a-entity>  
  <a-entity other-comp__helloworld="message: Hello, World! A great day!"  
    other-comp__metaverse="message: Hello, Metaverse AGAIN!">  
    </a-entity>  
  </a-scene>  
</body>  
</html>
```

Attaching a new component to a scene

```
AFRAME.registerComponent('new-comp', {  
  init: function () {  
    var sceneE1 = this.e1;  
  }  
});
```