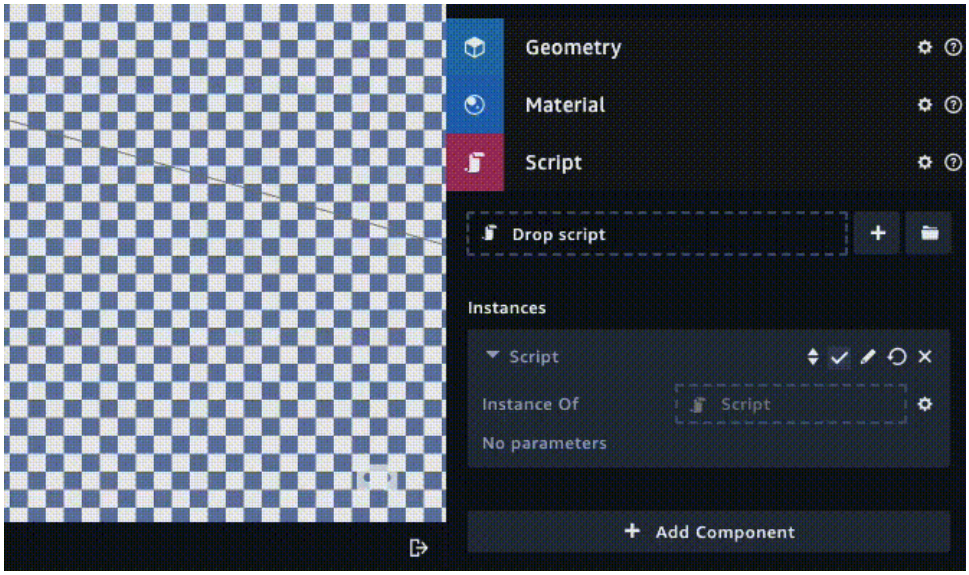
ADD VIDEO SCRIPT

- With the videoQuad still selected, choose Add Component.
- Add a Script component.
- Use the + button to add a new Custom (Legacy Format) script.
- Open the Script editor by pressing the pencil in the Scirpt window



- By default, the script is named Script. Rename it to VideoScript. You can copy and paste the following code into the script.

```
'use strict';

var parameters = [
{type: 'string', key: 'videoUrl', 'default': '', description: 'URL
to Video'},
];

// Set up a <video> element and use it as a source for a texture
function setup(args, ctx) {
```

```
ctx.worldData.playVideo = event => {
// Attempt to play video. This may be blocked by the 'autoplay'
policy of
// the browser. If so, we need to capture a gesture from the user
before
// attempting to play(). We do so by emitting a message which a
Behavior will
// listen for to obtain that gesture.
const playPromise = ctx.worldData.video.play();
if (playPromise !== undefined) {
  playPromise.then( () => {
    // Autoplay started
    console.log(`play was successful on
${ctx.worldData.video.src}`);
    ctx.worldData.videoMat = new
sumerian.Material(sumerian.ShaderLib.textured);

    ctx.worldData.videoTexture = new sumerian.Texture(null, {
      generateMipmaps: false,
      minFilter: 'BilinearNoMipMaps'
    });

    ctx.worldData.videoTexture.updateCallback = function () {
      return ctx.worldData.videoTexture.image &&
!ctx.worldData.videoTexture.image.paused;
    };

    ctx.worldData.videoTexture.readyCallback = function() {
      return ctx.worldData.videoTexture.image &&
ctx.worldData.videoTexture.image.readyState === 4;
    };

    ctx.worldData.videoTexture.setImage(ctx.worldData.video);
    ctx.worldData.videoMat.setTexture('DIFFUSE_MAP',
ctx.worldData.videoTexture);
    ctx.entity.meshRendererComponent.materials[0] =
ctx.worldData.videoMat;
  }).catch(error => {
    // Autoplay was prevented. Emit a signal to obtain a user
gesture
    console.log('video Autoplay blocked - emitting
"VideoAutoplayBlocked" signal. Respond to this message by obtaining
user gesture and then emitting a "PlayVideo" signal.');
    sumerian.SystemBus.emit('VideoAutoplayBlocked');
  });
}
};

sumerian.SystemBus.addListener('PlayVideo',
ctx.worldData.playVideo);
```

```
ctx.worldData.video = makeVideo(ctx, args.videoUrl);
}

function cleanup(args, ctx) {
ctx.worldData.video.pause();
ctx.worldData.video = null;
ctx.worldData.videoMat = null;
ctx.worldData.videoTexture = null;
sumerian.SystemBus.removeListener('PlayVideo',
ctx.worldData.playVideo);
}

// Creates a <video> tag in the HTML DOM
function makeVideo(parentCtx, url) {
let video = document.createElement('video');
video.crossOrigin = 'anonymous';
video.src = url;
video.muted = false;
video.autoplay = true;
video.playsInline = true;       // iPhone requires this to prevent
fullscreen video playback
video.oncanplay = function(ctx) {
    video.width = video.videoWidth;
    video.height = video.videoHeight;
    sumerian.SystemBus.emit('PlayVideo');
};
return video;
}
```