

Creative Solutions to HTML5 Limitations as a Gaming Platform

Kevin Moot & Scott Bromander

Interactive Developers, The Nerderly



 @docix

 sbromand@nerdery.com



 @kcmoot

 kmoot@nerdery.com

Outline

- **Gaming on the web**
- **HTML5 Limitations**
 - 3D
 - Audio
 - Fullscreen
 - Pointer Lock
 - Gamepads
 - Local storage
 - Media streams
- **Road to a brighter future**

Gaming on the Web



Flash

- Historically dominated gaming
- Mature dev tools
- Browser-agnostic

Gaming on the Web



Concerns

- Mobile discontinuation
- Minimal support for Windows 8 Apps

Gaming on the Web



HTML5

- No third-party plugins necessary
- Mobile support

Gaming on the Web



Concerns

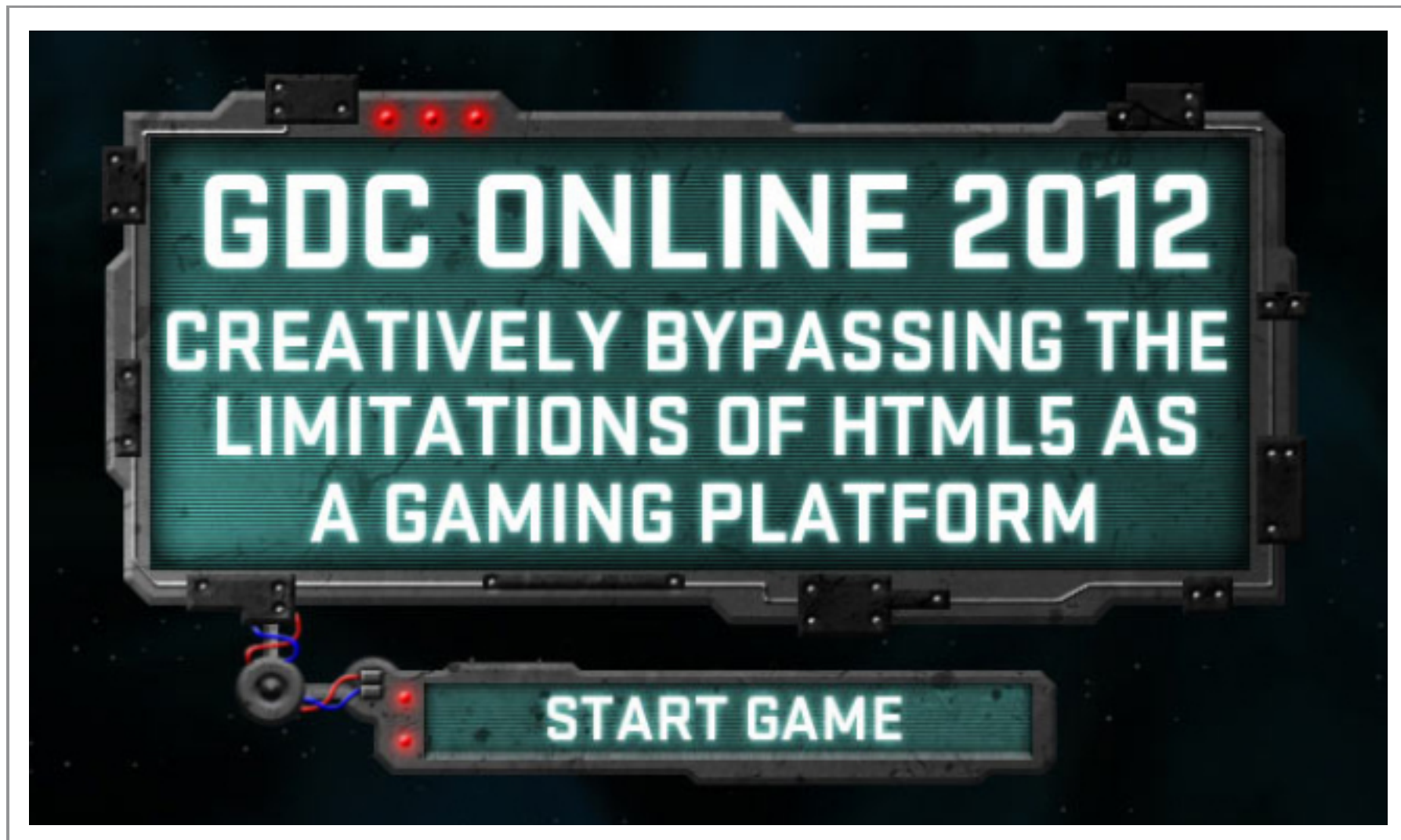
- Immature spec
- Inconsistent feature set
- Security & privacy

Overcoming HTML5 Limitations

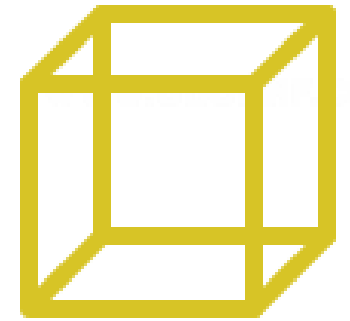


- Creative workarounds
- Proprietary API's
- Polyfill-plugins

The Demo



3D Limitations



Native via:

- WebGL
- 3D CSS transforms

But:

- Incomplete browser support
- Security concerns

WebGL Support

FULL



**PARTIAL/
EXPERIMENTAL**



NOT HAPPENING



Software rendering

Three.js library

- WebGL not required
- Output to 2D canvas
- Broadens compatibility to all modern browsers

Software rendering

Limitations:

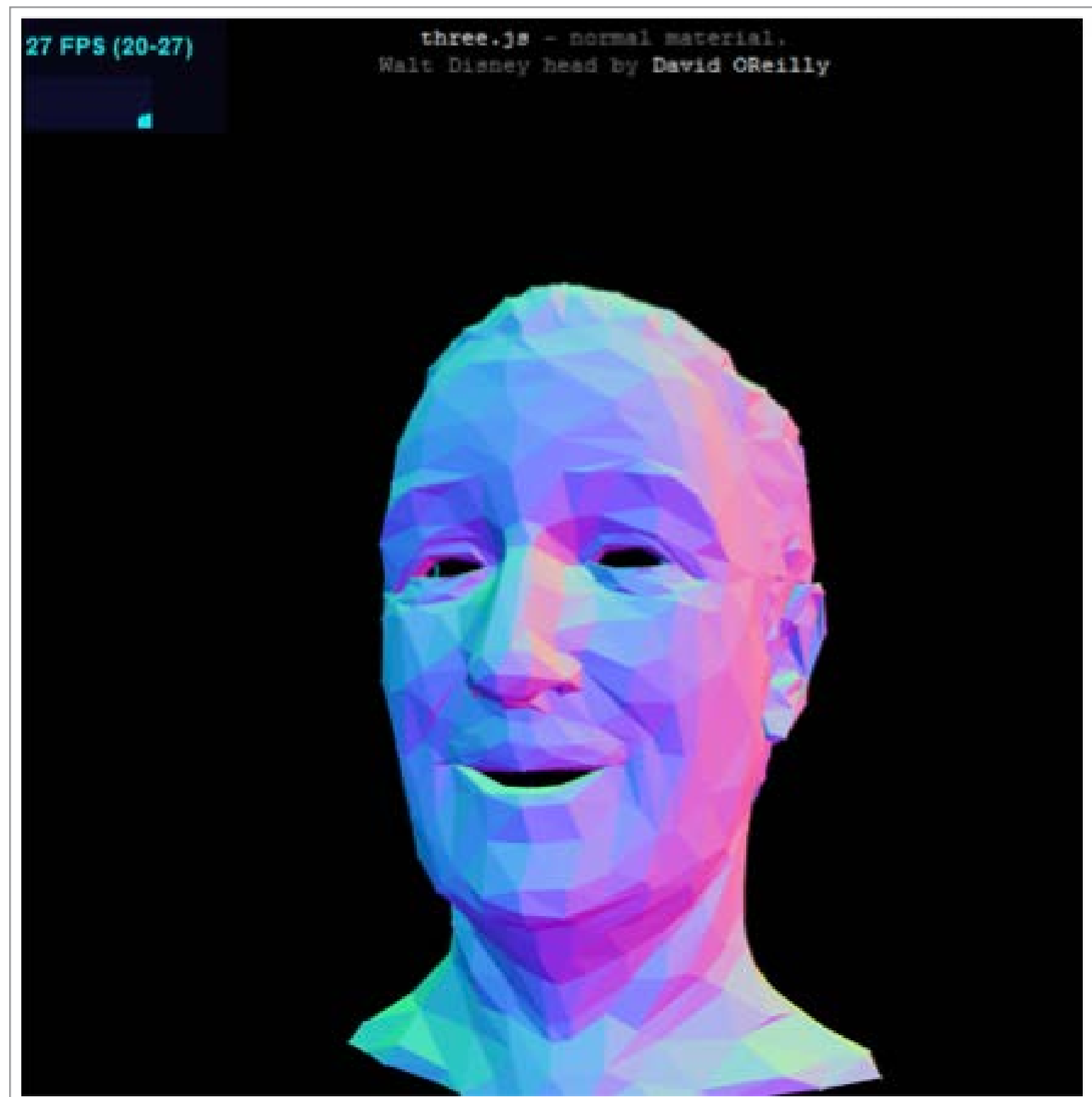
- Less performant than WebGL
- Basic lighting only
- No custom shaders

Suitable for:

- Lo-poly scenes
- Fallback version



Software @27 FPS



WebGL @62 FPS



WebGL in IE



Remember when we said...

NOT HAPPENING



WebGL in IE



IEWebGL extension

- Unendorsed plugin
- IE 6+
- Canvas & video texture limits

Replace <canvas> tag with:

```
<script src="http://iewebgl.com/scripts/webglhelper.js"/>  
<object type="application/x-webgl" />
```

WebGL in IE



Google Chrome frame

- Embedded webkit browser
- Obfuscation concerns
- Limited support (offline application cache)

HTML:

```
<meta http-equiv="X-UA-Compatible" content="chrome=1" />
```

Header:

```
X-UA-Compatible "chrome=1" env=gcf
```


CSS 3D Transforms

Pros

- Wide browser support
- No security concerns

Cons

- Slow
- No concept of scene / camera
- No lighting / shaders

CSS 3D Transforms

Photon 3D lighting engine

- Adds flat lighting effects
- Webkit-only

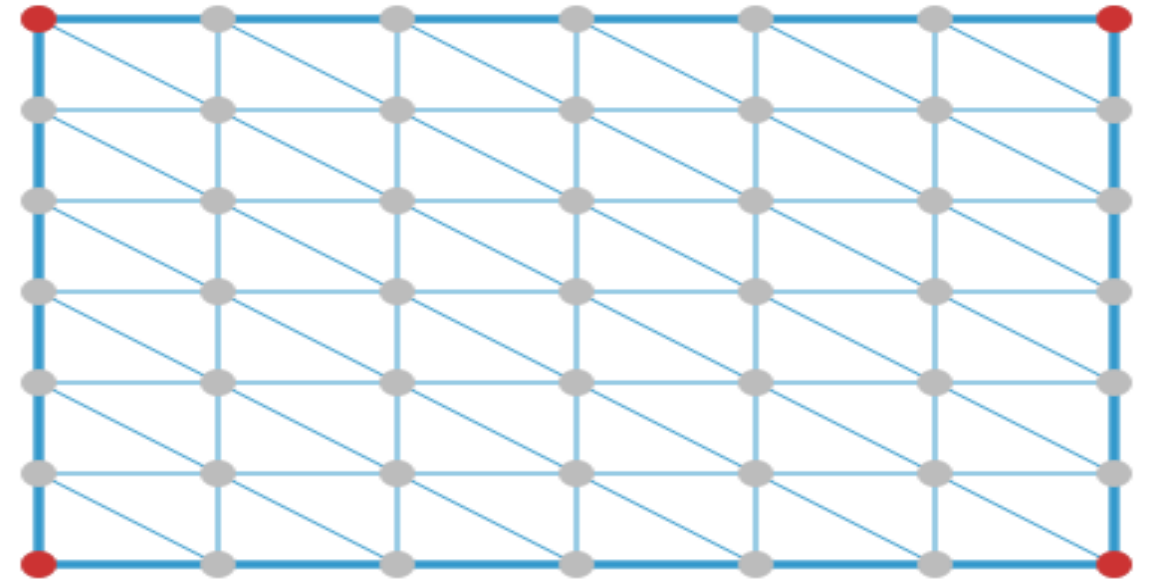


CSS Custom Filters

- Vertex & fragment shaders
- OpenGL ES shading language
- Chrome developer build

```
.my-shader {  
    -webkit-filter: custom(none mix(url(shaders/grayscale.fs)  
    normal source-atop), amount 1);  
}
```

CSS Custom Filters



Wrap Up – 3D

Up Next: Audio

- Offline cache
- Web audio

Audio Limitations



HTML5 Native audio

- Basic play/pause/seek
- Universal support

```
<audio id="my-audio" src="explosion.mp3" />
```

Audio Limitations



Limitations

- High latency (resolution=16ms)
- No processing/effects
- Can't access data

Suitable for

- Sync is not important
- Background music

Overcoming Latency

Offline Application Cache

- Cached on disk rather than network resource
- Eliminates buffer/seek latency
- No IE9 support
- Mobile storage limit
- Mobile video/audio limit

Offline Application Cache

cache.manifest file

```
CACHE MANIFEST
```

```
CACHE:
```

```
/audio/explosion.mp3
```

```
/audio/cannon.mp3
```

index.html

```
<html manifest="cache.manifest">
```

```
<body>
```

```
    <audio src="explosion.mp3" />
```

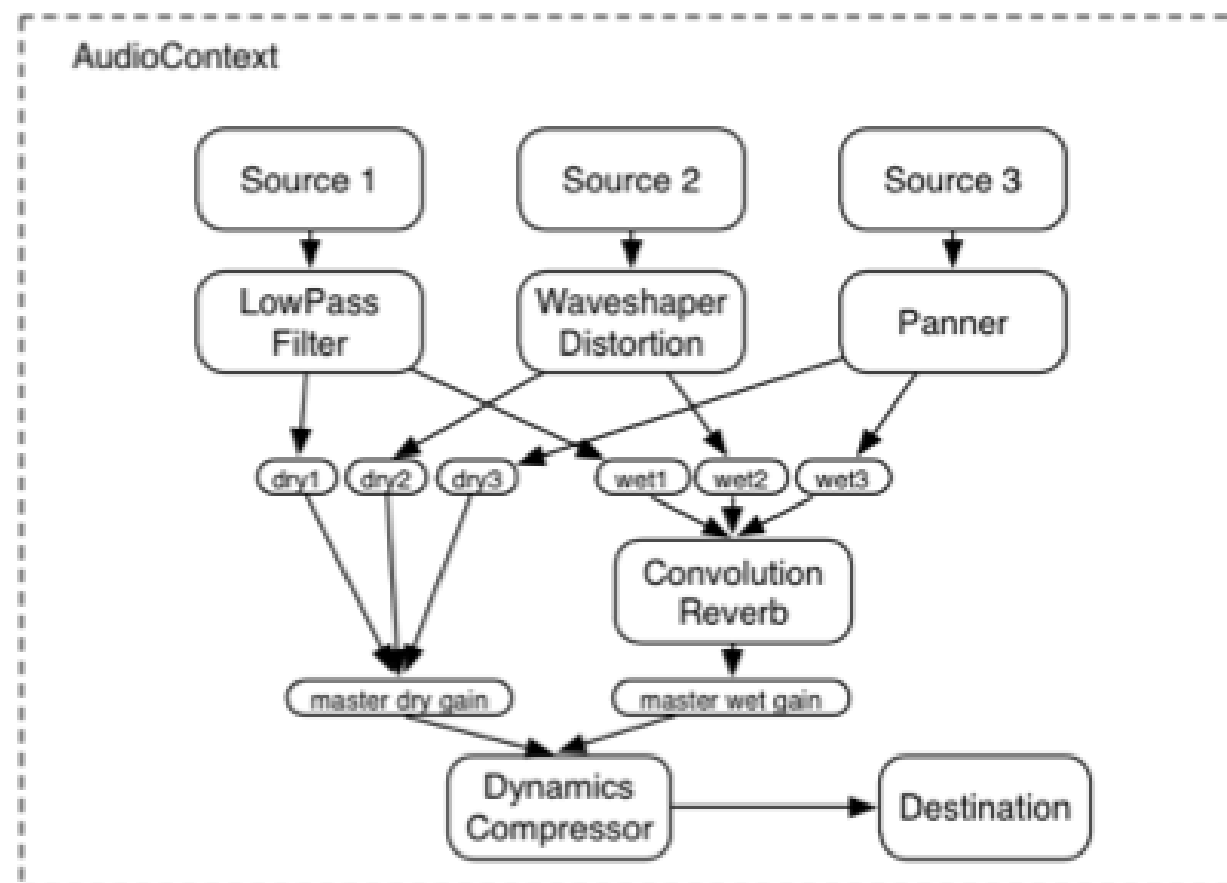
```
    <audio src="cannon.mp3" />
```

```
</body>
```

Web Audio API

Features

- Based on OpenAL
- Low-latency
- Processing/effects



BROWSERS
SUPPORTED



Web Audio API

by [Roby T. Miller](#)

```
var context = new AudioContext();  
var source = context.createBufferSource();  
  
// Buffer contains binary audio (preloaded)  
source.buffer = buffer;  
  
// Destination will be your speakers  
source.connect(context.destination);  
  
// Play sound!  
source.noteOn(0);
```

BROWSERS
SUPPORTED



Web Audio API

Limitations

- Chrome/Safari
- Firefox – deprecated Audio Data API
- iOS6 only
- Legacy iPad & iPod Touch stuck with iOS5
- Audio binary retrieved via AJAX
- AJAX introduces cross-origin (CORS) issue

BROWSERS
SUPPORTED



Mobile Audio Limitations

Limitations

- Mobile Safari audio crippled prior to iOS 6
- Can't harness same capability as native app

Motivation

- Reduce bandwidth & data charges



iOS 5 Limitations

- No autoplay
- No volume control
- One audio stream at a time
- Delay when switching streams



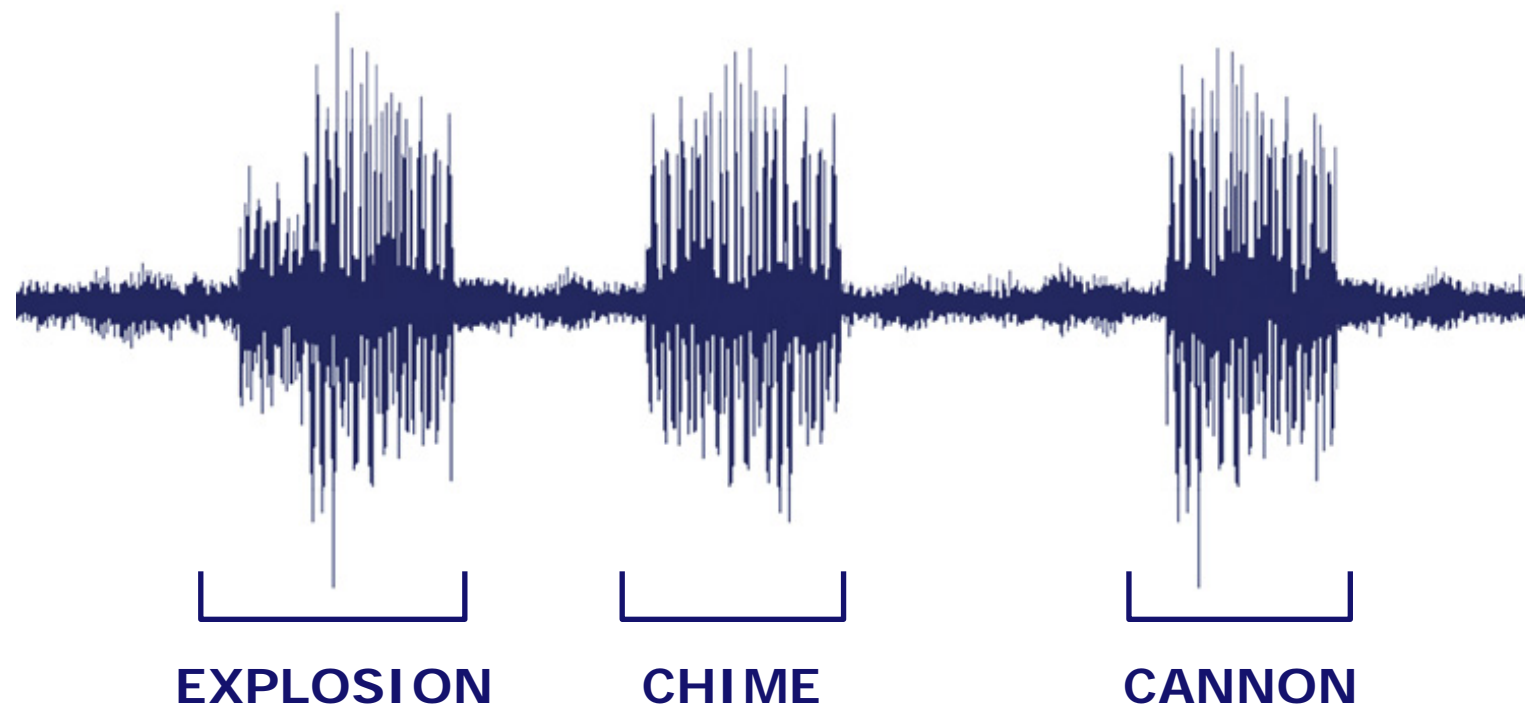
Illustrating Audio Delay

```
var audio1 = document.getElementById('audio1');  
var audio2 = document.getElementById('audio2');  
  
audio1.play();  
audio2.play();  
  
// audio1 immediately stops  
// audio2 plays after a half-second delay
```

Eliminating Delays

Audio Spritesheet

- Multiple sounds in a single audio file
- Seek to desired sound
- Generous blank space between



Eliminating Delays

```
var sprite = {  
    explosion: {  
        start: 0,  
        length: 5  
    },  
    cannon: {  
        start: 8,  
        length: 3.5  
    }  
};
```



```
var audio = document.getElementById('audio');  
audio.addEventListener('timeupdate', onTimeUpdate, false);  
audio.currentTime = sprite.explosion.start;  
audio.play();
```

```
function onTimeUpdate() {  
    var sound = sprite.explosion;  
    if (this.currentTime >= sound.start + sound.length) {  
        this.pause();  
    }  
}
```

Overcoming Mobile Audio Limitations

- Audio loads when triggered by touch event
- Use splash screen to load audio sprite



Mobile Browser Replacements

- Capabilities of native app
- Direct access to graphics/sound hardware
- User must install an app

Examples:

- AppMobi's DirectCanvas
- Ejecta

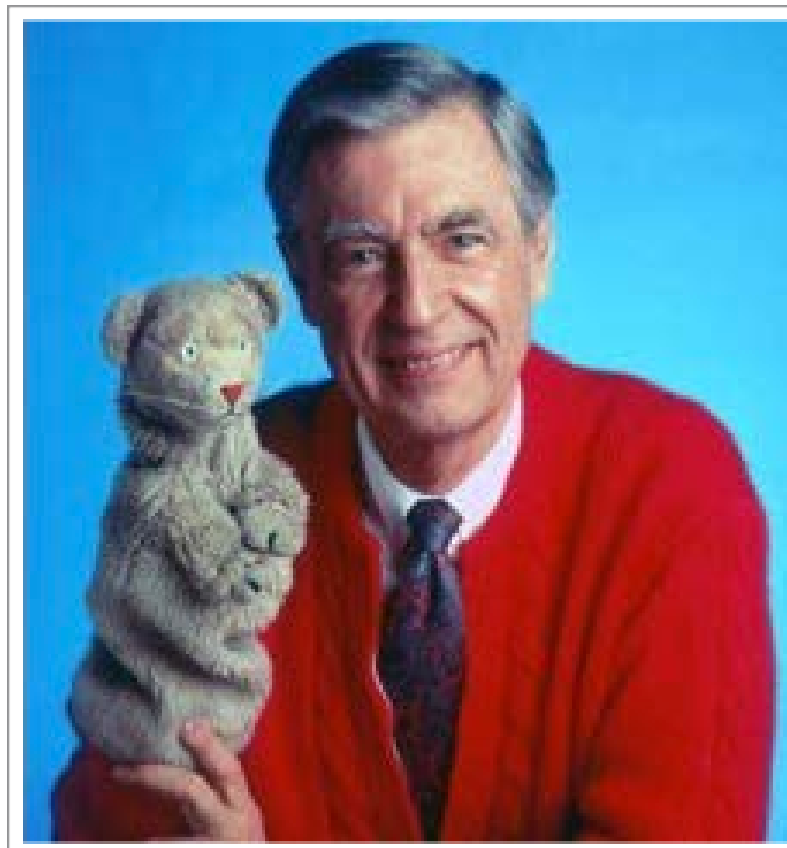
Wrap Up, Audio

- Offline cache
- Web audio

Up Next: Fullscreen limitations

Fullscreen Limitations

- Historically impossible to force fullscreen
- Unneighborly conduct of a good web citizen
- Spoofing concerns



F11

- Low-tech solution
- Instruct users to press 'F11'
- Enters browser's fullscreen mode

Fullscreen API

- Pending final spec
- Fires via user-initiated event (e.g. button click)
- Keyboard limitations
- Only: arrow keys/space/control/alt/shift

```
element.webkitRequestFullscreen(); //Safari & Chrome
```

```
element.mozRequestFullscreen(); //Firefox
```

```
element.requestFullscreen(); //W3C Proposal
```

BROWSERS
SUPPORTED



Pointer Lock Limitations

- Mouse moves outside of browser window
- Can't capture mouse input!
- First-person environments? ☹️



Pointer Lock API

- Hides cursor
- Fullscreen mode required
- New movementX/movementY property

BROWSERS
SUPPORTED

```
element.requestPointerLock();  
  
document.addEventListener('mousemove', function(e) {  
    var deltaX = e.movementX;  
    var deltaY = e.movementY;  
  
}, false);
```



Gamepad Limitations

Browsers limit input to:

- Mouse
- Keyboard
- Touch events



Gamepad API

- Works with some (but not all) gamepads
- Non-standardized spec
 - Chrome: Continuously poll for button/axis state
 - Firefox: Event-driven

BROWSERS
SUPPORTED



Gamepad Mapping

Specific to each gamepad / browser

```
var ps3buttons = [];  
ps3Buttons[12] = 'triangle';  
ps3Buttons[15] = 'square';  
ps3Buttons[14] = 'cross';  
ps3Buttons[13] = 'circle';  
ps3Buttons[4] = 'up';  
ps3Buttons[7] = 'left';  
ps3Buttons[6] = 'down';  
ps3Buttons[5] = 'right';  
ps3Buttons[10] = 'L1';  
ps3Buttons[8] = 'L2';  
ps3Buttons[11] = 'R1';  
ps3Buttons[9] = 'R2';  
ps3Buttons[1] = 'L3';  
ps3Buttons[2] = 'R3';  
ps3Buttons[16] = 'PS';  
ps3Buttons[0] = 'select';  
ps3Buttons[3] = 'start';
```

Example mapping:
PS3 controller in Firefox

Gamepad API

```
window.addEventListener('MozGamepadButtonDown', buttonDown);  
window.addEventListener('MozGamepadButtonUp', buttonUp);  
window.addEventListener('MozGamepadAxisMove', moveAxis);
```

```
function buttonDown(e) {  
    var button = e.button;  
}
```

```
function buttonUp(e) {  
    var button = e.button;  
}
```

```
function moveAxis(e) {  
    var axis = e.axis; //which axis  
    var value = e.value; //magnitude of movement  
}
```



Offline Storage Limitations

Key/value strings:

- Cookies - 4K limit
- localStorage ~5MB limit

Javascript objects:

- IndexedDB ~50MB limit

Transient

- Wiping browser history
- Exiting incognito session



Cloud Storage

- Permanent
- Javascript API
- Save data or realtime broadcasting

```
<script type='text/javascript'
src='http://static.firebase.com/demo/firebase.js'>
</script>

var db = new Firebase('http://demo.firebase.com/my-game');

db.set({
  name: 'Dave',
  score: 42412
});
```



BROWSERS
SUPPORTED



FileWriter API

- Chrome-only
- Sandboxed file system
- Store large binaries

BROWSERS
SUPPORTED



FileWriter API

Cache large game assets

- Download single tarball
- Expand into local directory
- Reference assets via local URL

BROWSERS
SUPPORTED



FileWriter API

```
window.requestFileSystem(  
    window.PERSISTENT, 1024*1024, gotFileSystem  
);  
  
function gotFileSystem(fs) {  
    fs.root.getFile(  
        '/audio/explosion.mp3',  
        null,  
        function(file) {  
            document.getElementById('audio').src = file.toURL();  
        }  
    );  
}
```

BROWSERS
SUPPORTED



Media Stream Limitations

How can we access:

- Webcam
- Microphone

Historically not possible without Flash



getUserMedia API

- Access user's audio/video stream
- Part of proposed WebRTC spec (Real-Time Communications)
- www.webrtc.org

BROWSERS
SUPPORTED



Video Stream

- Capture frames
- Record video
- Realtime effects
- Movement recognition

```
navigator.getUserMedia({video:true}, gotVideo);  
  
function gotVideo(stream) {  
    var webcam = window.URL.createObjectURL(stream);  
  
    document.getElementById('video').src = webcam;  
}
```

BROWSERS
SUPPORTED



Audio Stream

- Input from microphone or webcam audio
- Not yet integrated with Web Audio API
- Future potential:
 - Peer-to-peer conferencing
 - Voice chat

```
navigator.getUserMedia({audio:true}, gotAudio);
```

BROWSERS
SUPPORTED



Road to a Brighter Future

- Immaturity of native HTML5 gaming
- Potential is growing
- Demand universal, consistent implementations
- Browser vendors & W3C



References

Browser compatibility charts

www.caniuse.com

Three.js

<http://mrdoob.github.com/three.js>

IWebGL

<http://iwebgl.com>

Google Chrome frame

<http://www.google.com/chromeframe>

Photon 3D lighting engine

<http://photon.attasi.com>

AppMobi DirectCanvas

<http://www.appmobi.com/gamedev>

Ejecta

<http://impactjs.com/ejecta>

Firebase

<http://www.firebase.com>

WebRTC spec

<http://www.webrtc.org>

Questions?

Presented by NERDERY.COM



@docix



sbromand@nerdery.com



@kcmoot



kmoot@nerdery.com