Media Frameworks Technical Summary Draft 1

Media Frameworks is a cross platform, cross language open media authoring, logging and playback API that strives towards open standards and simplicity.

Technology Stack

Azure : used to host web services we are currently using node.js,

document database we are currently using mongo DB to store scenes in JSON,

blog storage raw storage for image files

Node.js packages

Socket.io

Many many more node packages I can list them all if you like…

DataLayer

MongoDB holds scenes

HTML Player and GUI target Google Chrome (38 +)

JavaScript

React <https://facebook.github.io/react/>

Socket.io <http://socket.io/>

WebAPIs

Vimeo : video streaming

SoundCloud : audio streaming

MediaFrameworks Asset Store: image storage and parsing of XMP meta data to azure storage blob

<https://github.com/Colum-SMA-Dev/AssetStore>

MediaFrameworks Hub: manage web socket connections and loading scenes

<https://github.com/Colum-SMA-Dev/MediaHub>

<https://github.com/Colum-SMA-Dev/MediaHubAPI>

<https://github.com/Colum-SMA-Dev/MediaPlaybackFramework>

MediaFrameworks Controller: manage media frameworks scene

<https://github.com/Colum-SMA-Dev/html-controller>

Diagram

Pilot Version Phase

Lightweight approach, all browser based implementation. Create the simplest possible testable framework. Focus on images and simple transitions. Images will be tagged with XMP metadata this data will be parses into a MediaFrameworks scene file when it is uploaded.

Scene : collection of media objects and authoring/playback parameters

Parameter : Name value pair that the API understands

Theme : collection of tags (tag set). Themes can use simplified Boolean logic to create sets of tags. We currently support AND, OR and ()

Tag : keyword(s) for a media object

JSON for Scene :

Advantage : Simple syntax

Disadvantage : Possible logical and parsing errors. Maybe need a lint checker

Azure

Advantage : Azure made it very easy to deploy and scale web services, blob storage and mongoDB

Disadvantage: Not free costing us about $100 a month to maintain

Node.js

Advantage:

Disadvantage:

Web Services

Vimeo

Advantage: Open and can be free. Tags can be read without authorization.

Disadvantage: Free version didn't support all of the player feature we wanted we ended up switching to a paid account. We also considered YouTube but YouTube requires authorization to read tags. The YouTube API also supports temporal tagging. When we started Vimeo API might have been better then YouTube but now I think YouTube may be better.

SoundCloud:

Advantage: Lightweight and easy to use API

Disadvantage: web and JavaScript only we will need to build proxy services for native playback outside of a browser

AssetStore:

Advantage: Azure is fast, easy. Our service also translates XMP meta data into scene tags

Disadvantage: No export back to XMP yet. No other web API outside of Azure. Might be nice to support flicker, Instagram or Picasa…

Requested Features

Test group1 Digital City

Zoned Random Images

Video Subclips

Test group2 SonicExperience

FadeIn Fadeout audio

Crossfade audio

Audio Cues

Consist of different sounds

Audio SubClips

Phased Approach

Phase 1 Complete

Strategy

Create the smallest possible API to display basic media types with a simple client. The platform target for phase 1 will be a full HTML stack with an HTML client.

Image : Stored in Azure

Video: Played from Vimeo : uses vimeo tags and descriptions

Audio: Played from SoundCloud : uses SC tags and descriptions

Phase 1 built an HTML MediaEditor, HTMLMedia Player and an HTML GIU Controller for the HTMLMedia Player. Browser requirements for the MediaEditor/Player were Chrome 38+ (we did all of our testing on the most recent version of Chrome currently 39) and FireFox 37+

Phase 2

Create other reference clients. WebGL, Python, C++ OpenGL or DirectX, Unity C#, MaxMSP Plugin, Epic Engine… Creating other clients will help us keep our API simple and flexible as we cross platforms and programming languages.

Phase 3

Add features to API for more types of media with more specialized features/

Audio/Video Sub clips

3D objects

3D Markers and camera control

Temporal Scene Editing

Audio Parameters

Cues

Architecture

Layers for Media Framework

* + MediaPlayers : display media request responding to a MediaPlayerController or MediaHub
    - HTMLCanvasMediaPlayer (priority 1)
    - HTMLWebGLMediaPlater (priority 2)
    - UnityGameEngineMediaPlayer (priority 3)
    - EpicGameEngineMediaPlayer (priority 3)
  + MediaPlayerController : UI (staring with html) for building media scenes for playback on a hub or player
    - HTML interface
    - Saves MediaScene to JSON to local storage for testing or direct real-time control over a media Controller
    - Saves MediaScene to PlayerPlayerController
  + MediaController : interacts with MediaHubs to control MediaPlayers by a MediaPlayerController of files using the MediaControllerAPI
  + MediaHub : Media Players subscribe to a media hub. MediaHub directly control players though a web socket
  + MediaControllerAPI : API used to control MediaPlayers and Hubs
  + ServerPlayerController : Server that plays Media Scenes to Media Controllers

## Phase 1 Test Groups

Digital City : Collaboration between Columbia College Chicago and Beijing Film Academy.

Image and Content Issues

Content from both Chicago and Beijing were uploaded to Vimeo, SoundCloud and our Asset store. By design our AssetStore didn’t limit the size of any uploads (images). We also by design didn’t do any thumb nailing or image scaling/reduction on upload. Student often uploaded large original camera images up to 25Mb per image. This became an issue as the Media Framework editor would often become unresponsive or possibly crash when a scene with many large image was edited. The crashes were due to not enough RAM for the browser to run and some scene needed to be edited on workstation with more RAM as many student laptops with less RAM were crashing. In the future we should either store scaled images that are appropriate for the client or recommend that images be pre scaled down before they are uploaded.

Vimeo and Soundcloud worked very well in Chicago. We did upgrade to a Pro Vimeo Account as this allowed us to better control the UI of the player. Vimeo would occasionally time out in Chicago probably due to the rapid rate of requests in some scenes.

Issues in China

Our content in the AssetStore layer in Azure was distributed use Azure Content Deliver Network (CND). The primary location for the assets in the AssetStore was North America Central (ORD) but we also distributed content to the Azure East Asia. This was done to try to improve the responsiveness of our app in Beijing. We used network monitoring tools to determine the network load time in Beijing could be as long as 1500 – 2000ms which were much larger than the 60-150ms we were used to. After adding the CDN load times were reduced to at best 200 – 300ms but this was very unreliable and varied greatly.

Load times in China were also very long to Vimeo and SoundCloud. These sites were not blocked by the Chinese firewall but they were often very slow and unresponsive. I suspect this was due to bandwidth shaping and network throttling. Our app often makes many many http request in a short period of time and many of these requests are for RTP streams. This type of behavior is often throttled in a low bandwidth or NAP protected network. I suspect this type of throttling also increased latency in China and made MediaFramework unresponsive.

We did experiment with using a VPN connection to our Chicago location. The VPN would allow us to bypass the firewall and any network throttling but just using a VPN connection also increases latency. The China location may have also been traffic shaping VPN connection to it wasn’t much more reliable.

Vimeo and SoundCloud would often time out in the MediaFrameworks player in china. I’m assuming this was due to the network latency.

We used HAR files and analyzers to test latency <https://toolbox.googleapps.com/apps/har_analyzer/>

We also use websites to test for URL connectivity through the Chinese firewall <http://www.greatfirewallofchina.org/>

Sonic Experience

Audio Latency issues were not complained about with this group.

JSON

The proposed Scene specification was simplified and expanded to include some new feature both are included below for reference

Proposed spec

{

"version": "1",

"name": "scene1",

"scene": [

{

"mediaObject": {

"name": "image",

"type": "image",

"mimeType": "image/png",

"url": "<http://someurl.com/image.png>",

"anmiationIn": "default",

"anmiationOut": "default",

"cachePolicy": "default",

"tags": "some,tags"

}

},

{

"mediaObject": {

"name": "video",

"type": "video",

"mimeType": "video/ogg",

"url": "<http://anotherurl.com/video.ogg>",

"anmiationIn": "default",

"anmiationOut": "default",

"cachePolicy": "default",

"subclips": [

{

"in": "00:00:00:00",

"out": "00:00:00:05",

"tags": "some,tags"

}

]

}

}

]

}

Current Implementation Demo Scene

{

"name": "GUIsceneSecurity",

"version": "1",

"maximumOnScreen": {

"image": 3,

"text": 1,

"video": 1,

"audio": 1

},

"displayDuration": 10,

"displayInterval": 3,

"transitionDuration": 1.4,

"themes": {

"MillBlower": "(formant 1khz AND formant 300hz) AND macro dur, (BP 300hz AND high freq amplified) AND macro dur, (HP 2khz AND BP 400hz) AND highpass beats, (BP 2khz AND BP 400hz AND BP 5khz) AND echo, (BP 400hz AND echo AND strong beat AND macro dur), (BP 1khz AND echo AND strong beat AND continuous)",

"SnowPiercer": "highend clear AND (no beat AND timestretch)",

"Squaredancing": "square wave, quantized",

},

"style": {

"backgroundColor": "black"

},

"scene": [

{

"tags": "March, Soldiers, Guards, Beijing, China, Tienanmen Square, Forbidden City, Culture, Tradition, History, Underground, Walkway",

"type": "video",

"volume": 100,

"url": "<https://vimeo.com/126084982>"

},

{

"tags": "sculpture, Abraham Lincoln, Augustus Saint-Gaudens",

"type": "image",

"url": "<https://smaassetstore.blob.core.windows.net/assetstoremattmacbook/404bfedf18ab6a71f11581f5791c374b/DN-080749.jpg>"

},

{

"tags": "hans solo",

"solo": true,

"type": "image",

"url": "<http://img1.wikia.nocookie.net/__cb20100129155042/starwars/images/0/01/Hansoloprofile.jpg>"

},

{

"tags": "demo video",

"type": "video",

"volume": 100,

"url": "<https://vimeo.com/1179590>"

},

{

"tags": "high freq attenuate, BP 300hz, natural sound foreground, periodic echo, complex source, no beat, wheels, squeak, El train, free field, processed source, short dur, formalizedScene",

"type": "audio",

"volume": 100,

"url": "<https://soundcloud.com/insound-3/elsstake09-altoechosegment-single>"

},

{

"tags": "high freq attenuate, BP 300hz, natural sound foreground, periodic echo, complex source, no beat, El train, wheels, squeak, free field, processed source, medium dur, formalizedScene",

"type": "audio",

"volume": 100,

"url": "<https://soundcloud.com/insound-3/elsstake09-altoechosegment>"

},

]

}