Research

3D Technology for Web

There are two main competitors in the area of interactive 3D for web browsers as of now: Flash and WebGL.

Flash

Adobe Flash is a multimedia and software platform widely used throughout Internet.

Flash uses vector graphics, static or animated, supports streaming of videos, and offers other multimedia related features. The language typically associated with online Flash is ActionScript, although Haxe can be complied into Flash applications as well. Flash is a proprietary platform of Adobe. Freeware editing software exists, but arguably inferior to licensed Adobe Flash Professional.

Flash requires a plug-in (Flash Player) to run in web browsers, which is enough of a reason to reject many competing technologies. However, the Flash Player is so prevalent today that most users have it installed regardless of our application. Adobe claimed to have 99% penetration on desktop computers in 2011 [101]. That makes the necessity of a plug-in a much smaller issue for our project.

Being plug-in based still means a lot, though. The advantage is that the application will look the same on every device and doesn’t need to be tweaked for different browsers. The reason behind that is that the application is executed by Flash Player, not the browser. The disadvantage is that it is not fully integrated in the web page.

Since version 11.2 (2012), Flash Player allows advanced mouse control such as scrolling or right click [107], which was one of the main features missing in previous versions. Nevertheless, user input might be clashing with the rest of the HTML page, resulting in poor user experience.

Flash Player 11 also introduced Scene3D, an API for development of hardware accelerated 3D content. Scene3D is one of the major candidates for our project’s 3D API. Previous Flash technologies relied on CPU rendering and were significantly slower than GPU accelerated alternatives. Such slower technologies were used in previous implementations of the atlas.

Historically, Flash had been one step ahead of traditional web development. When HTML didn’t allow an easy way to create highly interactive, animated web pages, Flash did. But the time has come when web development started to catch up. HTML5 enables development of rich and sophisticated websites, parsed directly by the browsers without the drawbacks of external plug-ins. Flash is no longer the unparalleled champion of rich Internet applications.

For several years, numerous writers and developers have claimed that Flash is insecure [105], a dying technology [103] and a developmental dead end. Despite these voices, Flash is still alive in 2014 and here to stay for some time. Although still widely used, the future of Flash is not all bright. According to statistics at builtwith.com, the usage of Flash on major websites has been on the decline lately [104].

Adobe Flash is currently available on vast majority of desktop devices [111], although the development for Linux has been discontinued since Flash Player 11.2 (outside of Google Chrome).

Flash Player for mobile browsers availability is controversial. It is not available on iOS devices (iPhone, iPad…), but it was officially available on Android 2.2-4.3 and BlackBerry (Tablet OS, BB10). However, in 2011 Adobe personnel announced that future development of Flash will not focus on mobile browsing and admits that HTML5 is the best solution to deploying browser content across mobile platforms [102]. In 2012, Adobe also announced that Flash wouldn’t be installed on new Android devices and Flash Player for Android will no longer be updated [108]. Apple’s (and formerly Steve Job’s) attitude and lack of support also speaks against mobile Flash [106].

Despite the lack of native support, there are ways and workarounds to view Flash content on most mobile platforms including iOS [109] and Android 4.4 [110].

WebGL

WebGL (Web Graphics Library) is a JavaScript API for development of interactive 3D scenes for the web. It uses underlying low-level graphics API, typically OpenGL ES 2.0, to make full use of GPU acceleration. It is a royalty-free, cross platform standard maintained by the non-profit Khronos Group.

WebGL uses HTML5 Canvas to display its content and doesn’t require the use of a plug-in. It will work on any browser with WebGL support without additional software (although it might require appropriate GPU drivers). It is fully integrated into the webpage and the application’s execution is managed by the browser, which might result in slight variations across browsers. Even the use of underlying API depends on the environment, many browsers use ANGLE [116] to utilize DirectX over OpenGL on Windows.

The required features such as advanced mouse controls and 3D acceleration are all present and comparable to those of Flash with possible minor benefits (such as consistency of user controls throughout the page).

WebGL is a new technology compared to Flash. WebGL 1.0 standard was issued in 2011 [112] and in the following years, support by vendors and developers has been growing. Firefox, Google Chrome and Safari have supported WebGL for some time [113]. Microsoft support starts with Internet Explorer 11 and applies to all IE11 devices [114]. Apple announced support on their new iOS 8 [115]. Default Android browser doesn’t support it, but Firefox and Chrome for Android do.

Other options

Other technologies exist but were quickly dismissed as they seem to be inferior to both WebGL and Flash. These technologies mostly consist of XML based markup languages such as X3D (and its predecessor, VRML) or 3DMLW. They require additional plug-ins to be displayed in browsers and their popularity and support doesn’t compare to WebGL and Flash.

Performance Comparison

A simple performance test was performed on a laptop (link to specs? list them?) to compare Flash and WebGL. A demo application with a various number of semi-transparent cubes was used, implemented once using Flash Scene3D and once using WebGL [117]. The implementation wasn’t studied in detail and might be a source of a bias; the performance might be affected by hardware usage fluctuations.

Results for Internet Explorer 11:

<insert graph>

Results for Firefox 32:

<insert graph>

Measurements in IE11 show similar results for both technologies, going slightly in favor of WebGL overall.

Measurements in FF32 show a great difference in FPS in favor of WebGL. However, it is important to note that although the reported FPS was high, the application stopped redrawing as expected at 1000 cubes and up, only redrawing upon user interaction in WebGL. Flash on the other hand worked properly but slowly.

Conclusion

Both Adobe Flash and WebGL offer us what we need: GPU acceleration, sufficient interactivity and quality of user controls. Both options seem to be perfectly viable for our purpose.

<http://webglsamples.googlecode.com/hg/aquarium/aquarium.html>

<http://www.researchgate.net/post/Do_you_consider_WebGL_or_Stage_3D_flash_more_suitable_for_creating_a_web_application_for_3D_interactive_contact>

<http://codeflow.org/entries/2013/feb/02/why-you-should-use-webgl/>

<http://www.irrlicht3d.org/pivot/entry.php?id=1255>

<http://www.cescg.org/CESCG-2013/papers/Bahor-HTML5WebGL_vs_Flash_in_3D_Visualisation.pdf>

<http://www.streetwisetech.com/2014/09/will-html5-finally-takeover-adobe-flash-player/173806.html>

101 <http://www.adobe.com/cz/products/flashplatformruntimes/statistics.html>

103 <http://archive.wired.com/insights/2014/05/flash-dead-long-live-webgl/>

102 <http://blogs.adobe.com/conversations/2011/11/flash-focus.html>

104 <http://trends.builtwith.com/framework/Shockwave-Flash-Embed>

105 <http://www.infoworld.com/article/2610420/adobe-flash/adobe-flash--insecure--outdated--and-here-to-stay.html>

106 <http://en.wikipedia.org/wiki/Apple_and_Adobe_Flash_controversy>

107 <http://tomkrcha.com/?p=2621>

108 <http://www.theguardian.com/technology/blog/2012/jun/29/flash-android-dies-adobe>

109 <http://thefusejoplin.com/2014/04/adobe-flash-player-12-download-iphone-free/>

110 <http://wccftech.com/install-flash-player-in-android-4-4/>

111 <http://en.wikipedia.org/wiki/Adobe_Flash#User_experience>

112 <http://www.khronos.org/news/press/khronos-releases-final-webgl-1.0-specification>

113 <http://caniuse.com/#feat=webgl>

114 <http://msdn.microsoft.com/en-us/library/ie/bg182648%28v=vs.85%29.aspx>

115 <http://www.cnet.com/news/ios-8-brings-big-boost-for-web-programmers/>

116 <https://code.google.com/p/angleproject/>

117 <http://www.airtightinteractive.com/2011/10/stage3d-vs-webgl-performance/>