International Journal of Cloud Computing and Services Science (IJ-CLOSER)
Vol.1, No.1, March 2012, pp. 26~35
ISSN: 2089-3337

The Future of Web and Mobile Game Development

Kevin Curran, Ciaran George
University of Ulster, School of Computing and Intelligent Systems
email: kj.curran@ulster.ac.uk

Article Info

Article history:

Received December 11th, 2011
Accepted Jan 26th, 2012

Keyword:

Web Programming
HTML 5

ABSTRACT

Since HTML5 has become open to the public many developers have been experimenting with the new possibilities for web development, the aim of this research paper is to give an overview of what this means to the game development community. This research paper will evaluate new HTML5 elements and JavaScript features. We highlight WebGL, Canvas and WebSockets, which have given developers the opportunity to flaunt their creativity by manipulating images, creating 3D environments and providing real-time interaction.

Corresponding Author:

Kevin Curran,
University of Ulster,
School of Computing and Intelligent Systems,
Northland Road, Londonderry, N. Ireland
Email: kj.curran@ulster.ac.uk

1. INTRODUCTION

In 1997 a new and exciting version of the Web's most predominant mark-up language was officially announced as the new W3C recommendation, this was HTML 4.0. It was vastly different to its predecessors and offered an abundance of new features such as the support of Cascading Style Sheets. Before the release of HTML 4.0 the web was largely limited to text and static images, with which many of the websites we use today could not have been created. W3C believed the future of the web lay with XML and thus froze their development of HTML after 4.01 and the release of XHTML. With the support of Apple the WHATWG led by Ian Hickson began work on a specification they called Web Applications 1.0, this was due to a disagreement of XML being the future of web development. Web Applications 1.0 was then used as the spec for the basis of W3C's new HTML version after they admitted being wrong in basing the future on XML. With HTML5 now in a working draft form and offering stacks of new exciting features developers have already begun to create the future of the web.

With the release of HTML5 to simplify developing for the web many new elements were added, along with the addition of several new JavaScript API's. Previous to HTML 5 playing media on a browser required a plug-in or an application installation. For example playing a Flash game requires Adobe shockwave player but now thanks to HTML5 and its new multimedia elements audio, video and canvas that could be a thing of the past. Although HTML5 is not without faults most see it as a huge leap forward for web development. Its release plays a vital role how web games are developed. Recently there has been a massive expansion of casual 'pick up and play' games which are frequently played on social networking sites, smart phones, tablets and web sites this has led to a major shift in the game development industry. The

market for casual games is still continuously growing with games like angry birds having achieved over 350 million download's [1]. The Canvas element provides a way for developers to draw and manipulate 2D images using HTML and JavaScript to implement movement of a canvas element the image must be deleted and redrawn continuously by the web OS. With the canvas element 2D games can be easily implemented and with the use of vector based images they can be manipulated without losing any quality. Websites such as FaceBook have already begun providing an abundance of games showing the effectiveness of HTML5 canvas.

The WebGL API is another exciting and innovative feature that allows the customisation of 3D objects by allowing JavaScript to communication with the users GPU. WebGL provides an API that allows 3D graphics to be used with canvas. This could potentially make web browsers a valid gaming platform, which would revolutionise game development. WebSocket technology provides full-duplex bi-directional communication channels over a single TCP socket in both web browsers and web servers [2]. The WebSocket API provides features in JavaScript that allow the implementation of real-time interactions on applications. For example web based games will be able to provide instant interactions with game objects (such as the canvas images) and provide in game chat functionalities. Combining Canvas and WebSockets allows for the development of a multiplayer web browser based games framework [3]. This paper will evaluate the effectiveness of these new HTML5 features and provide information on how these technologies can and are being used. The main focus will be on previous technologies that may now become deprecated and what the future holds for browser based game development.

2. MOBILE GAME DEVELOPMENT TECHNOLOGIES

The HTML 5 draft brings about substantial changes to the current web standards, from supporting static websites to highly interactive visual websites. Although some features such as WebSockets are not strictly a part of HTML5, they are standardised alongside features such as canvas. Most research will still refer to WebSockets as an aspect of HTML5. Many developers have already begun using canvas to manipulate or create 2d images.

2.1 Canvas and SVG

The HTML5 feature Canvas allows the developer to manipulate bitmap images with pixel operations; however this means that the image will lose its quality if a scaling operation is required. Although this does mean that image size will have no effect on performance, which is advantageous. Graphical manipulation can also be done using the XML based file format SVG (Scalable Vector Graphics), as a predecessor one would believe it to now be vastly inferior. This however is not really the case with SVG handling particular aspects much more effectively such as cross platform resolution scalability. As the name suggests SVG is used to manipulate vector graphics, everything drawn using SVG is created within a separate DOM element. Canvas differs from this in that everything is drawn within one element using JavaScript. The advantages of drawing to DOM are substantial; the images drawn to DOM can be manipulated after they have been drawn. However with canvas if movement is required for an image the entire image must be redrawn to the canvas in the required position to illustrate movement. You can see how SVG appears to be advantageous in this aspect of game development, having to redraw every frame is a costly process for both performance and development. Using DOM also provides SVG with more clever recognition of objects, for example with SVG if a red square were to be drawn on a green screen SVG remembers that behind the red square there is still a green screen. Canvas differs in this aspect; it would perceive the area under the red circle to be non-existent similar to that of Microsoft paint. This might sound like a huge problem but due to the previous point that canvas scenes must be redrawn it does not have as much of an impact. At this point it would seem that SVG is the obvious choice for game development, with its increased performance and easily manipulated images. Due to some recent web browser advancements canvas games are now running much faster, this is from browser introductions of hardware acceleration. Canvas hardware acceleration tasks the graphics card (GPU) with the drawing processes thus freeing up the CPU, this provides a massive improvement. These improvements can be seen in canvas games all over the web, comparing recent versions of web browsers with older versions lacking hardware acceleration will display this. As most web based games are currently aimed at casual gamers this would suggest that the most relevant technology would be the one which creates the better casual games. The majority of casual games are now being developed on canvas; this is because casual games contain a small amount of objects so large amounts of manipulation are not necessary. Games created mostly on the client side face some security issues. Client Side code can be easily obtained and altered by other developers; this could lead to programming code being stolen or manipulated. A solution to this is to

obfuscate the JavaScript in your game; this would encode a portion of the Client Side and make the information much more difficult to reproduce. When used in the right scenarios canvas and SVG are both appropriate technologies for game development.

2.2 2D Adobe Flash

2D games are extremely popular on web browsers, a large proportion of these games on sites such as Facebook are created using Adobe Flash Professional. Adobe Flash Professional is a multimedia authoring program which can be used to create games, movies and more. The main disadvantage of flash is that it requires plug-ins to work, with HTML5 games and videos can be run without the need for any plug-ins. Websites such as youtube have already begun to offer html5 capabilities [4] with the only 'disadvantage' being that ads are not yet supported. Flash has been used for the creation of some massive titles such as Zyngas Farmville, which at its peak reached around 80 million users [5]. It shows that flash is one of the largest gaming platforms in the world, with more users in Farmville alone that any recent console. HTML5 has been replicating much of the features provided in flash to offer a "plug-in free web". Moving to HTML5 would not only create a plug-in free web but would also provide games to more platforms, the biggest of these platforms would be those on the iOS. Apple has refused to support flash for years with the companies frequently clashing. Ex-Apple CEO Steve Jobs felt that Flash programs took up too much processing power and caused frequent crashes, he also believed this would become a huge drain on handheld devices batteries. These were the reasons he refused to support flash. HTML5 would finally offer a way to reach the apple user base [6]. Although HTML5 and JavaScript seem like a solution to flash and its plug-in requirements this is not the case at all. Flash is and always will be an attractive design tool; it reaches a wide audience due to its artistic qualities and designing features. Not all game designers want to write endless amounts of code and this is the niche market that will not be able to move to HTML5, with adobes latest software release this niche market can get all the benefits of flash design for HTML5.

Adobe Edge is the latest release from the Adobe labs, this exciting tool allows the creation of animations (such as those created in Flash) in a HTML5 environment. Its features appear to be much more future-proof than that of its predecessor flash, such as support for SVG, JavaScript and CSS. Most importantly though Adobe have acknowledged the massive, rapid expansion of handheld and tablet devices with Adobe Edge supporting all of the main Operating Systems. At present Adobe Edge does not yet offer anything close to what Adobe Flash offered, but adobe have said that it will be quickly and frequently updated. However it does not enter this market alone, another animation tool called "Hype" was introduced to the Mac OS market bragging "creates slick animations in HTML5". Hype is another potential design tool for game creation, with user friendly animation and simple drag and drop graphics importation. Although hype has stated it offers animations in HTML5 what it actually offers is simply HTML4, there is no evidence of support for any of the new HTML5 elements such as canvas. This problem alone shows that hype is much less future proof than the up and coming Adobe Edge with Adobe having over 15 years of animation software experience. Adobe are offering Edge for free to any interested designers, with its flash animation for HTML5 it would certainly be something to look into. The current version is more of a preview with additional features being added quite regularly. Adobe have also recently unveiled their new fully 3D API "Stage3D", this will be a huge competitor in the race for becoming the dominating 3D web software. This will be discussed in more detail later in this paper.

2.3 WebGL

HTML5 Canvas does offer other substantial benefits over other technologies; one of the most exciting features added is WebGL. WebGL is a JavaScript library that allows for the creation of 3D games, the WebGL API is based on OpenGL ES 2.0 and uses canvas elements. OpenGL is an API widely used for the development of 2D and 3D applications, within games development it is the main competitor for Microsoft's Direct X. To provide efficient 3D graphics there must be communication with the GPU, OpenGL acts like a translator for multiple programming languages enabling the developer to have achieve to this communication. WebGL offers this communication to JavaScript thus enabling the creation of web based 3D games. WebGL provides cross platform capabilities allowing the development of 3D games on most platforms, the only problem with this is that plug-ins are required to support this on small devices [7]. With hardware acceleration added to web browsers this allows the 3D games to run quite well. Drawing primitives directly to the graphics card is a difficult task; using WebGL could prove to be a steep learning curve for a lot of developers. Developers of Google have been showing off the potential for WebGL in games by displaying several projects that reveal outstanding potential. Figure 1 shows an old Nintendo 64 game "Quake 2" that

has been recreated to run on several browsers, but sadly it only ran at full potential on a customised version of Firefox called "Pale Moon". Games must be an immersive experience for the player to get the most enjoyment out of it. Immersion cannot occur when the game is surrounded by a brightly coloured web-browser interface. Some Web browsers have recently implemented a full screen feature to cater for an immersive experience, with such a simple addition it will make a huge difference. The browsers without full screen availability will not provide a fully immersive experience and thus will lack a necessity for 3D gaming. WebGL faces many problems in relation to entering the mobile device market, problems that range from support to hardware capabilities. WebGL requires processing to be done in the GPU but mobile devices are too small to contain powerful GPU's, as mobiles are the most commonly used casual gaming system this could be extremely problematic. Intel's relatively new processors nicknamed 'Sandybridge' is revolutionary in its design, presently they are found in desktop pc's for reasonable prices. Their GPU layer within the processor offers performance that can surprisingly match that of low end graphics cards sold by Nvidia and ATI. They are also the first integrated GPU's to support direct x and OpenGL giving instant support for 3D graphics. It has been rumoured that Sandybridge hybrids will be integrated into new Android phones and tablets; this would seriously increase the graphical potential for mobile gaming enabling developers to provide much more 3D content. As mentioned earlier creating 3D games with WebGL will be a difficult task, some developers have created 3D Graphical JavaScript libraries to simplify this process.

Figure 1: Google WebGL Quake 2 [8]

SpiderGL is one of these libraries which states that "*we have shown how the programming facilities exposed by the library help speeding up the creation of 3D web applications without forcing the developer to adopt predefined programming paradigms like scene graphs."* [9]. Libraries such as SpiderGL will require frequent updates if they are to be useful, due to the dynamic changes being made to WebGL as HTML5 is being standardised. WebGL has been frequently criticised concerning certain security issues, predominantly due to the related hardware exposure to the web. Microsoft have decided due to these security risks they will refuse to support the WebGL specification on their web operating system until they feel it is safe [10]. Although Microsoft's Internet explorer is not the leading web browser it still holds a considerable demographic of approximately 22% of web users [11]. This could lead to WebGL becoming less popular, or it could potentially pose as the final nail in the coffin of Internet Explorer which has gradually become the underdog of web browser operating systems. The Mozilla Company (creators of the web browser OS FireFox) have agreed that WebGL may pose security risks, but they have criticised Microsoft's refusal to support WebGL. Mozilla stated that Microsoft's product Silverlight posed the same security risks due to its use of Direct X. Before the Internet can see the rise of rich 3D content several security problems will need to be addressed, once these problems are addressed then WebGL may become supported effectively by all browsers. With hardware acceleration and WebGL being the technologies that are most likely to shape how canvas is used in the future [12] it's important for game developers to understand and observe them as they evolve. WebGL has the potential for 3D games to take off rapidly in the casual gaming market, with new and exciting opportunities unfolding with every new version release. Innovative WebGL games could flood the World Wide Web from social networking sites to web browsers; many have already been created as was mentioned earlier. It is important to note that WebGL is still in its infancy and it is not expected to populate the web until 2022.

2.4 Adobe Air and Stage3D

Recently Adobe released Air (Adobe Integrated Runtime) which allows mobile devices to run flash programs; this also means that developers using Adobe Flash could quickly and easily release content they had already created onto the phone and tablet markets. With the ability for flash to target mobile platforms so easily and neatly it remains a giant in the battle for dominance of games design. This still however does not solve the plug-in dependence problem; Adobe Air is simply another plug-in that is installed the first time a flash game is run on a mobile device. Although Adobe has added some very innovative features how will they keep up with the 3D graphics race? Stage3D is the answer, Stage3d or Molehill is the name given to a new 2D and 3D rendering API offered by the Adobe Company. The newest releases of the Flash platforms enable designers and developers to create immersive 3D cross-platform games, this is possible due to the same reasons as WebGL. Stage3D can leverage GPU hardware acceleration enabling it to offer huge performance increases. That said though suggests it will also encounter similar problems to WebGL due to the limited hardware available in phones and tablets. One key feature it offers is the ability to create hardware accelerated games using ActionScript. This is obtainable due to an abundance of Frameworks and Libraries that are offered for Stage3D. Bringing an element of simplicity to 3D design will enable Adobe to cater for their niche market which was discussed earlier. For the more technical developers who want to create their own frameworks, Flash allows development in the low-level AGAL language. Adobe have created an impressive API and have recognisably attempted to target a large audience using diverse ways in which to create applications. The IDE offered in Adobe is a very distinguished feature offering an attractive way to create 3D assets and move them around the game world. The 3D Asset-Centric interface offered is a huge selling point to games designers but it is not the only product offering this feature. 3D games engines, most notably Unity 3 offer a very impressive cross-platform multi-language IDE that could rival Adobe and turn users away from WebGL.

2.5 Unity 3D

Unity [12] offers console-like graphics on mobile platforms as shown in figure 2, as well as offering capabilities to release games directly to consoles. It is an Asset-Centric games engine focusing more on the manipulation of games object around an environment, with the ability to implement features using JavaScript, Boo or C# it also offers the developer multiple languages for script implementation. Not only does Unity allow for the creation of state of the art games but due to the features provided by the IDE itself it allows for games to be created quite quickly. There are hundreds of API's that cater for particular genres within unity, increasing the efficiency of the games engine itself. Unity have recently announced that they will be supporting the ability to import Flash 3D content directly into Unity. These two technologies combined offer an outstanding opportunity for the development of web and mobile games. Not much information has yet been released on the matter but Unity have been doing a tremendous job of promoting this exciting feature with events dedicated to teaching Flash developers how to build games with Unity. This also prevents users having to download the quite large file that is required to run Unity programs, as they could be run using Adobe Shockwave player. The main question is how WebGL can match an IDE that is designed to create games efficiently and effectively. HTML5 has some additional features that when used alongside WebGL offer a revolutionary gaming experience, these features are WebSockets and Web Storage. Although both these features are JavaScript API's they are being standardised along with HTML5 and are frequently referred to within the 'hot topic' HTML5.

Figure 2: Unity Cross Platform Game ShadowGun [13]

2.6 WebSockets

Normal web communication is achieved using HTTP, the problem with this is that it only allows transfer to occur in one direction at a time. HTTP communications also requires constant web page requests each time new data is required. This results in slow communication between client and server. Current technologies such as Ajax and Comet attempt to speed up this transfer. These technologies attempt to simulate a full-duplex connection, but they are merely a hack and still truly only provide one way communication. Comet uses methods such as long-polling or streaming, these methods however still involve using HTTP requests which causes latency issues. This is where WebSockets come in; HTTP was not designed to support real time communication but WebSockets upgrade this HTTP protocol to a WebSocket protocol. This enables true full-duplex, bidirectional connections, what this signifies is the ability to create real time multiplayer games on web applications. WebSockets portray a massive advancement for real time applications on the web. Although WebSockets do not render Ajax completely obsolete they do however supersede its solution for real time functionality. Originally WebSockets were implemented into all web browsers providing developers with the means to begin creating interactive real time games, however due to security vulnerabilities WebSockets were disabled in a number of browsers including Mozilla Firefox 4 [14]. This security issue has been corrected in the latest version of the WebSocket protocol, which means we can still expect to see multiplayer web based games coming becoming available in the latest web browser OS versions. The potential for WebSockets when used alongside other HTML5 technologies such as Canvas and WebGL is incredible, a number of games have already been created displaying different creative features built with WebSockets. Figure 3 shows an online multiplayer game "Rawkets" which looks similar to asteroids only offering multiplayer free for all game play. The game is in fact quite enjoyable and shows the potential of games, signifying that real time multiplayer Games could begin to expand all over social networking sites in the near future. Another game which adds to this theory is the online social chatting game (Figure 4) "Rumpetroll", which transforms players into a tadpole floating around space. The real time chat feature from WebSockets can clearly be seen here as the messages sent between players reach each other almost instantaneously. Using WebSockets in these games shows the availability of real time collision detection and chat features, with almost no latency being displayed WebSockets really is the "*Quantum Leap"* [15] that will bring real-time multiplayer games to the web and mobile devices. The Pusher API [16] enables developers to easily add WebSocket functionality to web games and mobile games, they also offer a number of libraries to be used within your game that explain how to effectively use Pushers client libraries. With all the new elements of HTML5 and the new JavaScript features there comes an abundance of third party software to optimise features and lessen the learning curve required to delve into these innovative elements.

Figure 3: Rawkets [17]

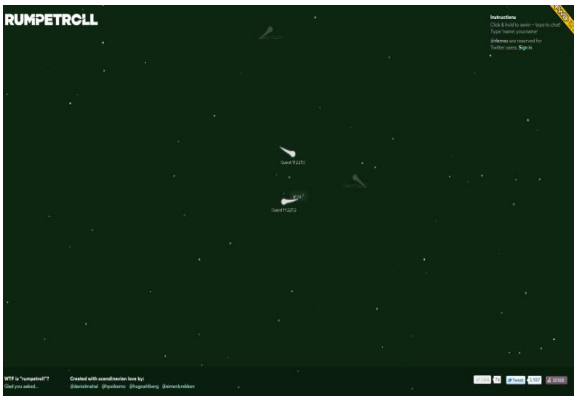

Figure 4: Rumpetroll [18]

The AVES engine provided isometric games using the real time functionalities provided by WebSockets. It showed a very impressive full screen world editor for browsers and mobile platforms. Some features were extremely impressive such as the ability to play videos and games on your protagonists in game television. The games engine provided an abundance of features that would help bring web games into a new era. The most notable of these was the ability to support a large number of users on the same server and the ability to communicate with the other users in real time. The engine was originally targeted at serious browser game publishers with a licensing model. Regretfully however this engine was bought out by Zynga before its potential was able to be fully shown. However Zynga may have bought out this product to revolutionise games such as Farmville or to simply remove potential competitors from the market. If Zynga do use the AVES engine to release content onto FaceBook, it will undoubtedly surpass the level that current FaceBook games are at. The AVES engine provides real multiplayer which is something that is not often seen on Facebook or any other social networking sites.

2.7 Web Storage

The Web Storage [19] JavaScript API provides developers with a tool for storing data on the client. This was previously done using cookies, but there are several problems with this. Firstly cookies cannot store a lot of data; usually they're used for minimal amounts of data storage such as website login details. Another disadvantage of cookies is that every request leads to the cookies being passed to the server; this can affect the website performance. Web Storage has been described as "*Cookies on steroids*" [20], it supports the removal of data when the browser is closed down but also supports the storage of data for scenarios in which the user may open the browser at a later date requiring older data. The two types of storage offered by Web Storage are Session Storage and Local Storage. Session storage will only store data on one particular 'session. If a new browser window is opened the new browser will not be able to retrieve the data. If the original window is closed the data is deleted. Local Storage however works differently; data that is stored in local storage can be retrieved on any browser window of the same domain. If the original window is closed the data will still be available for the user next time the browser is opened. Web Storage provides functionality that allows game data to be saved and loaded up when the user returns to play that game (whenever that may be). Although cookies are unreliable and if the user has cleared their internet storage it would include the removal of the local storage data. Although the size limit of local storage which is 5mb has become much larger than regular cookies which were 4kb it is still quite limited. The main drawback of the Web Storage API is that it only supports the storage of string values, there is a way around this however using JSON (JavaScript Object Notation) the string values can be parsed when recalled allowing them to return to their original format. The use of Web Storage has begun to show some interesting ideas for game development, one such game "Twitch" [21] is a series of puzzle games each on its own browser. When a puzzle is completed a new browser opens with the protagonist then appearing in the new window. Another example of an exceptionally unique game is Browser Pong [22], this game takes the traditional pong and replaces the paddles and ball with browser windows. The background for browser pong becomes whatever your background happens to be at the time and the player can control the web browser paddles using the arrow keys. Games like this are what show us the true potential of HTML5 by breaking the boundary, innovation is a marketing point and with all these new opportunities offered by HTML5 there are hundreds of revolutionary games to be made.

2.8 HTML5 Audio

The HTML audio element brings sound effects and music to games, when a certain event is triggered for example a player being shot we can then use JavaScript to play the correlating shot sound. Currently there is one main problem facing HTML audio, there is no consistent support. Web browsers are currently supporting different codec's. Originally when HTML 5 audio was introduced web browsers were to conform to use the open source Ogg Theora codec, however Apple refused to support this codec due to complications with Ogg and QuickTime. These complications led to the WHATWG dropping the specific requirements for HTML5 audio, now in order for the HTML5 audio to work cross-browser multiple Codec's must be inputted like so:

```
<audio id= "SelectButton">
<source src="http://audioexamplewebsite.com/sound.ogg">
<source src="http:// audioexamplewebsite.com/sound.mp3">
<source src="http:// audioexamplewebsite.com/sound.wav">
</audio>
```

This way if one codec is not supported, the program will fall back to the next codec and check if that is supported. Using 3 codec's should cover most web browsers as there only are 5 different audio codec's used across the 3 web main browsers. Hopefully soon web browsers will be able to agree on a default codec to simplify the implementation of HTML5 audio.

Offline applications [23] in HTML allow developers to store data in a cache manifest; the cache manifest is a list of files that can be accessed without an internet connection. We can then store all the game content files in this cache manifest, thus allowing the players to play games offline on both desktops and mobile platforms. Although this feature is not quite as exciting as other features in HTML5 it does have its beneficial qualities, offering players continuous entertainment even when an internet connection is not available. The Impact GameDev XDK [24] is a game development and deployment environment which uses HTML5. It is a mixture of two outstanding pieces of software; one aspect provides the Impact HTML5 Game engine and the other aspect provides the AppMobi XDK development tool and cloud services. The Impact JavaScript game engine provides the ability to build attractive games using canvas in HTML5. AppMobi offers several cloud services such as statMobi which allows you the developer to have access to an assortment of interesting metrics informing you on how well your game is doing. Other cloud services included with the AppMobi XDK include Live Update and 1Touch. Live Update offers an efficient way to update your game, due to bug fixes or new releases without the need for a re-release to the market. 1Touch offers the developer the opportunity to offer features in game for monetization with ease. These features all assorted into one package display an impressive development tool which allows developers to rapidly build innovative mobile games. The Impact XDK also offers a new adapted version of canvas called DirectCanvas; DirectCanvas has been optimised for games and boasts a 500% improvement in FPS (Frames Per Second) for mobile HTML5 games. Software that pushes the boundaries of HTML5 technology is hastening the process of getting new content into the World Wide Web. Sadly though AppMobi XDK is closed source and costs quite a lot, this is understandable due to the cloud services offered.

3. CONCLUSION

HTML 5 offers the chance to invigorate the web with its list of new amazing features. Each of the features that have been mentioned in this paper bring with them innovation and the possibility of great things. The features though reach their peak when incorporated with each other, in particular WebSockets and WebGL. We must also recognise the potential of other software like Adobe Flash's Direct 3D brings with it the chance to artistically create 3D environments and convert them directly into HTML5. It is an exciting time to be a game developer with such a rapidly growing market and with so many new revolutionary technologies to use each developer can have a part to play in the recreation of the 3D web. With Google at the fore front of demonstrating the capabilities of WebGL it truly displays its potential. Companies such as Zynga and Adobe have displayed a faith in the growth of HTML5 features and are preparing to enter the market instantaneously. Zynga bought Aves while adobe created their Stage 3D. 3D is coming to the web and research would suggest if companies do not embrace it they could be left behind.

It is obvious that the games industry has dramatically changed in the last few years; a huge amount of monetization now comes from mobile games and advertising on mobile games. There is also now potential for Indy games companies and lone developers to release their own content without the need for a publisher,

thanks to the app stores provided by android, iPhone and soon the amazon marketplace for the kindle fire. The only problem in this case is that the iPhones marketplace is extremely filtered, whereas androids market is filled with some good games and some awful games. When selling games on iPhone it is suggested to use a one-time fee, were-as on android due to the piracy of games it is suggested that monetization be done with either adds or by selling additional content. Mozilla, Google and other web browser creators are aware that with HTML5 changes need to be made to offer gamers an immersive enjoyable experience, changes such as the implementation of full screen modes. With HTML5 games take up the full size of the screen thus providing gamers with the immersion they desire. Gone are the windows that restrict your game to a small space in the middle of the browser. With statistics displaying how many users play games like Farmville alone it gives some insight into why it's so important to lead web browsers on a path that will hopefully help them in becoming valid gaming platforms. HTML5 provides a world wide web without the need for plugins, when you open a page it will work without the infuriating message "Plugin required".
Canvas does not deprecate Flash or SVG, at the moment all these technologies uses are viable for different reasons and perform most effectively in individual cases. Canvas is far from perfect at the moment and like other aspects of HTML5 will become more effective with time, for now we must accept that there will be problems such as the security issues with WebSockets. These problems should not prevent the creation of HTML5 games and applications, getting used to these technologies now will be beneficial. It is not uncommon to see HTML5 and flash being compared in depth to judge which will be the reigning game development toolset. With Adobes recent release Adobe Edge there is no need to compare the giants when they can be used in harmony to deliver rich 3D content. Browser based 2d multiplayer games have yet to become more popular, these types of games should appear all over the web relatively soon. Research suggests that 3D games won't become popular until a much later date so it is not advisable to wait until the expansion of 3D games. Deciding on which tool to use for the creation of 3D web games is merely a matter of personal preference.

Looking into all the technologies that have been mentioned that enable the creation of 3D games would be best advised, they are each unique and offer extremely different toolkits. They should be investigated in depth to make the correct choice. With phones becoming more powerful and soon implementing quad core processors with the possibility of Sandybirdge style processors we can expect to see a lot more games at the level of ShadowGun and perhaps beyond within the next few years. Unity and Flash 3D still require plugins to work and WebGL is not quite yet up to the same standard as Unity or Flash 3D. With the use of graphics API's like SpiderGL hopefully WebGL will move along in leaps and bounds. With Unity supporting importing to and from flash this offers a huge amount of flexibility for development. An animation could be created in flash and then directly imported into Unity to be used within a game scene. There is huge competition between third party software companies attempting to quickly get into such a new and immerging market. Most of this software offers extremely useful features that can ease the development of games. AppMobi has dominated the market with its solution for cross platform mobile game development using HTML5, Javascript and CSS3. It's no surprise though as it does offer an abundance of very interesting and amazing features. With companies like Aves being bought by huge companies like Zynga you can understand exactly why there is such a desire to create something that will provide revolutionary features. It can be concluded that huge advancements have been made in recent years with the implementation of hardware acceleration and real full-duplex communications via WebSockets. These advancements enable the consideration for 3D content to be added to web browser games along with access to multiplayer functionality. Local Storage offers a way for the player to save their game, though this will require a fall back in case the cookies are removed. Several dispute need to be corrected before certain features can reach their full potential i.e. Support for WebGL and A generic Codec. WebGL was not very well supported by most of the browsers used, Google's quake example only worked on a heavily modified version of Firefox. It will be very interesting to see how these technologies perform once they have been fully supported and optimised. But it is very clear to see why the development community is so excited about these new features.

REFERENCES

[1] http://gamasutra.com/view/news/37194/Angry_Birds_Surpasses_350M_Total_Downloads.php accessed on 26/10/2011
[2] I. Fette "The Web Socket protocol," IETF Standard September 30th, 2011
[3] Bijin Chen; Zhiqi Xu; , "A framework for browser-based Multiplayer Online Games using WebGL and WebSocket," *Multimedia Technology (ICMT), 2011 International Conference on* , vol., no., pp.471-474, 26-28 July 2011 IEEE
[4] http://www.youtube.com/html5
[5]http://www.techradar.com/news/gaming/farmville-now-has-80-million-facebook-users-671731 27/10/2011

[6] Makzan "HTML5 Games Development by Example: Beginners Guide" August 2011 p. 16.
[7] S. Ortiz Jr, "Is 3D finaly ready for the web?," IEEE Computer, Volume 43 Issue 1, pp. 14-16, January 2010
[8] "Quake 2" http://playwebgl.com/games/quake-2-webgl/
[9] M. Di Benedetto, F. Ponchio, F. Ganovelli, R. Scopigno "SpiderGL: A JavaScript 3D Graphics Library for Next-Generation WWW" Visual Computing Lab, ISTI-CN, Web3D 2010. 15th Conference on 3D Web technology – 2010
[10] http://www.computerweekly.com/Articles/2011/06/17/247024/Internet-Explorer-will-not-support-insecure-WebGL-3D-graphics-spec-says.htm accessed on 28/10/2011
[11] http://www.w3schools.com/browsers/browsers_stats.asp
[12] R. Hawkes "Foundation HTML5 Canvas For Games and Entertainment" April 2011 p. 285
[13] http://www.madfingergames.com/g_shadowgun.html
[14] http://hacks.mozilla.org/2010/12/websockets-disabled-in-firefox-4/
[15] http://websocket.org/quantum.html accessed 02/11/2011
[16] http://pusher.com
[17] "Rawkets" http://rawkets.com/
[18] "Rumpertroll" http://rumpetroll.com/
[19] I. Hickson "Web Storage" Specification, Editors Draft, W3C, October 4th 2011 http://dev.w3.org/html5/webstorage/
[20] B. Lawson, R. Sharp "Introducing HTML5" 18 October 2011 p. 143
[21] "Twitch" http://www.html5gameonline.com/game/enigma/twitch
[22] "Browser Pong" http://stewd.io/pong/
[23] I. Hickson, A. Kesteren "Offline Web Applications" http://www.w3.org/TR/offline-webapps/
[24] "Impact XDK" http://www.appmobi.com/?q=node/204

REFERENCES

The main references are international journals and proceedings. All references should be to the most pertinent and up-to-date sources. References are written in Vancouver style. Please use a consistent format for references – see examples below (9 pt):

[1] X. S. Li, *et al.*, "Analysis and Simplification of Three-Dimensional Space Vector PWM for Three-Phase Four-Leg Inverters," *IEEE Transactions on Industrial Electronics*, vol. 58, pp. 450-464, Feb 2011.
[2] R. Arulmozhiyal and K. Baskaran, "Implementation of a Fuzzy PI Controller for Speed Control of Induction Motors Using FPGA," *Journal of Power Electronics*, vol. 10, pp. 65-71, 2010.
[3] D. Zhang, *et al.*, "Common Mode Circulating Current Control of Interleaved Three-Phase Two-Level Voltage-Source Converters with Discontinuous Space-Vector Modulation," *2009 IEEE Energy Conversion Congress and Exposition, Vols 1-6*, pp. 3906-3912, 2009.
[4] Z. Yinhai, *et al.*, "A Novel SVPWM Modulation Scheme," in *Applied Power Electronics Conference and Exposition, 2009. APEC 2009. Twenty-Fourth Annual IEEE*, 2009, pp. 128-131.

BIBLIOGRAPHY OF AUTHORS

Kevin Curran BSc (Hons), PhD, SMIEEE, FBCS CITP, SMACM, FHEA is a Reader in Computer Science at the University of Ulster. His achievements include winning and managing UK & European Framework projects and Technology Transfer Schemes. He has published over 700 published works to date. He is the Editor in Chief of the International Journal of Ambient Computing and Intelligence (IJACI). Dr Curran is a Fellow of the Higher Education Academy, a Fellow of the British Computer Society and is listed by Marquis in their prestigious Who's Who in Science and Engineering. He is also listed in the Dictionary of International Biography and by Who's Who in the World.