Investigation into the use of HTML 5 game engines to create a responsive social educational game for children

Pankaj Kashyap

A Thesis submitted in part fulfillment of the requirements for the degree of

Master of Information Technology (Software Engineering)

School of Engineering and IT

Faculty of EHSE

Charles Darwin University

Darwin

May 2015

Abstract

HTML5 is a growing markup language with advanced and sophisticated features. Game development in HTML5 has caught the attention of the web development community, as HTML5 provides a collection of elements that can be used in conjunction with associated technologies such as CSS and JavaScript to develop interactive games. It has been adopted broadly by developers to create “responsive” web content which is in demand in today’s increasingly mobile society. Apart from the ability to create aresponsive design, HTML5 has tremendous capabilities to create enriched and interactive web contents and games.

To create an interactive game, it would be advantageous to use HTML5 along with game development frameworks known as a game engine. Game Engines are reusable software components that act as frameworks and are popularly used to create video games (Cowan 2014). The game functions such as graphical user interface, sounds, collisions, display, artificial intelligence and more can easily be implemented by game engines (Ward 2008). These frameworks are set of tools, which can be implemented directly into the game development process reducing the time it takes to develop a game. This project will initially investigate about HTML5 and its usability and then analyse different JavaScript game engines (HTML5 compatible), that can be used to develop a social game which will work on modern handheld device and range of web browsers.

To develop a social game, firstly the technologies associated with HTML5 were mastered. A variety of HTML5 game engines were reviewed and analyzed in order to understand and evaluate their different features. A variety of mobile social games were also investigated to gain the knowledge of the features of the HTML5 game development.

Finally, based on the above knowledge, a social mobile game was developed that would help school children to learn elementary Mathematics. The game is called MathMe which will help kids learn basic mathematics operation such as a+b, a-b, a/d, a*d. This game can be referred as foundation to create other more complex and interactive games.

Acknowledgement

I would like to express special gratitude towards my supervisors Mrs. Rebecca England, Mr. Sami Azam & Mrs. Barbara White for providing guidance and necessary information related to the project. All of the supervisors enlightened my path with knowledge and information.

Also, I would like to thank Ms. Mirjam Jonkman for her efforts for providing all the necessary information regarding this unit.

Table of Contents

List of Tables

List of Figures

Glossary

I. 2D & 3D Rendering- This is a process of displaying animations and visual effects on a screen (Sanders 2014).

II. Sound Engines- These are part of game engines and are important for providing sound effects within a game (Sivak 2009).

III. Physics Engines- These are algorithms in game engines which are responsible for simulating physical phenomena, for example the trajectory of the ball when pushed with certain angle and force (ComputerHope 2014).

IV. Collision detection and response- This refers to the computations and calculations involved to determine the intersection of objects in the game. These computations can be used to measure time of impact and how objects must behave before and after the impact (Ericson 2005).

V. WebGL- WebGL is based on OpenGL, a code base that provides for rendering of 2D and 3D graphics. It can be used to render hardware generated 3D graphics directly to the web browser. To bring the graphics to browsers no additional software is required (Cantor & Jones 2012). In games OpenGL can be implemented to control hardware devices such as mobile, gaming console or desktop computers (OpenGL 2014).

VI. APIs- OpenGL and WebGL are examples of API. API stands for Application Programming Interface. They provide a standardized interface allowing the programmers and developer to concentrate on creating a quality product, rather than thinking of specifications for different platforms (Sellers, Wright & Haemel 2013).

VII. Frameworks- These are software products that give generic functionalities. During the software development, these functionalities can be used accordingly depending on the product requirements. These frameworks may include libraries, smart tools, compiler, APIs and other components (Riehle 2000).

1. Introduction

In the past two decades, markup languages have evolved as a form of web communication and have attained recognition because of their ability to create, capture and share information over the World Wide Web (WWW). The commercial use of Markup Languages has led to exponential competitive benefits. In the current competitive business world, the web standard which has gained most attention among the web development community is HTML 5, the latest in the evolution of the web mark-up languages.

HTML 5 is a development platform upon which very successful web based games can be developed, especially when making use of what are called ‘game engines’. Game engines are frameworks, which can be used to develop games. A game engine facilitates the developers to easily implement core functionalities such as video, animations, sound, physics, collision and many more (Smiley 2012). Each game engine is unique and designed to achieve a specific set of functionalities within a game (html5gameengine 2014). A major benefit of using HTML5 game engines is that they are very lightweight and modular and support JavaScript-friendly syntax (Rettig 2014). Moreover, HTML 5 game engines can perform complex tasks which are required when building social games. Before the popularity of HTML 5, social games were developed with emphasis on scripting and animation platforms. However, HTML 5 caught the attention of developers due to its rich features.

To develop a social game using HTML 5it is important to understand its evolution and its associated markup languages. Moreover, the thesis will attempt to find answers to the following questions:

1. What are the most suitable game engines for development of HTML5 social games?
2. What are the HTML5 solutions for creating single code base game that will run on all web-enabled devices?

2. Literature Review

The aim of this section is to investigate established and cutting edge web technologies and discover the significance of HTML5 in a modern gaming scenario. The literature review details the timeline of different web technologies and their association with each other. Moreover, it discusses the significance of using game engine in HTML5 game development.

2.1.Origin of Internet

Early in the 1960s, the use of computer systems increased rapidly, and communication between computers had become a priority. The idea of setting up a communication interface between computing systems was first proposed in 1962 by computer scientist, J.C.R Licklinder of Bolt, Beranek and Newman (BBN) (Leiner et al. 2009) who became the chief of Defense Department's Advanced Research Projects Agency (DARPA). Licklinder initiated the concept of different methods of computer communication (Leiner et al. 2009).

In 1964, Kleinrock (Kleinrock 1972) developed a new communication system called Packet Networking/Switching where information could be transmitted using packets rather than circuits. The feasibility of the packet-based transmission was tested by researchers Roberts and Marill (Marill & Roberts 1966) who set up a connection between two computers located in Massachusetts and California using a Low-speed Dial-up connection. This experiment proved that the circuit switched telephone communication systems were not efficient for computer communication. From 1965 to 1988, significant developments occurred in the development of computer-to-computer communication processes. These include: the introduction of ARPNET, using the Network Control Protocol (NCP), of the Transmission Control Protocol/ Internet Protocol (TCP/IP) and the Domain Name Server (DNS) (Marill & Roberts 1966).

A few years later, the research group in European Organization for Nuclear Research (CERN), Switzerland, led by Tim Berners-Lee, proposed a network upon which they could easily transmit information by connecting a Hypertext Transfer Protocol (HTTP) to the TCP and DNS. This invention led to the World Wide Web (WWW) (Leiner et al. 2009). The idea behind the WWW was to provide a Graphical User Interface (GUI) for the display of transmitted data. Berners-Lee supported his idea of WWW system with the use of Hyper Text for representing the data on network (Carey 2012).

2.2.Web Coding Languages

Hyper-Text defined a new relationship the between the users and the developers. Unlike conventional information distribution systems like newspapers, books and journals, it allowed a means of communication which was not restricted to only one knowledge base, and users could directly jump from one web page to other using hyperlinks (Johnson-Eilola & Hea 2003). Figure-1 shows how the use of Hyper Text allowed a communication method in which there was no need to follow a sequential approach to access the information. The information from any source was easily accessible by any user sitting on a networked computer.

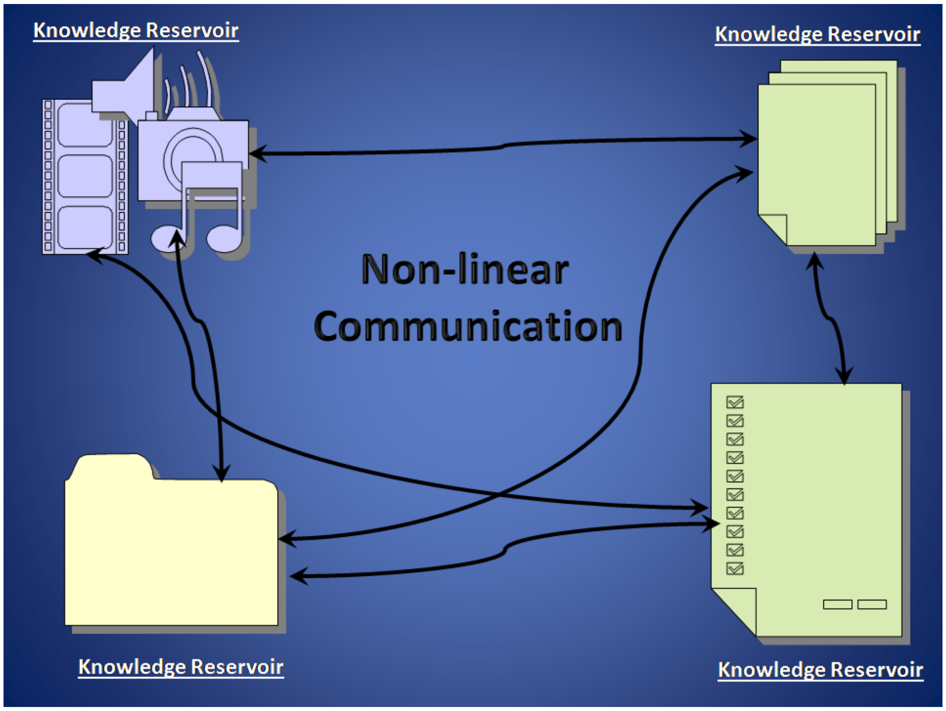


Figure 1: Example Diagram showing basic idea behind non-linear communication (author's own diagram)

earlier in the 1970's on Markup Languages. A range of other web coding languages were also developed to support the original markup languages. An outline of the evolution of the web coding languages will be discussed following.

2.2.1. Beginning of HTML

Hypertext Markup Language was based on the Standard Generalized Markup Language (SGML), defined in the document ISO8879: 1986 as an international standard. SGML was introduced with a philosophy of separating the data processing system from the information representation structure (W3C 2013b). The Electronic Manuscript Project of the Association of America Publisher (AAP) was one of the first applications of the SGML (sglmsource 1990). SGML was introduced as a descriptive markup system, enabling developers to declare the characteristics and structure of the document. Markup languages like SGML are static in nature, and they do not have procedures to process the raw data into information (Toshniwal & Agrawal 2004). SGML become a standard system for defining next generation markup languages such as HTML and XML (W3C 2013b; W3CRecommendation 1999b).

2.2.2. HTML versions 1 to 4

HyperText Markup Language (HTML) was the first simplified versions of the SGML. There were also new add-on features in HTML, which were missing in SGML, such as, “p” (paragraph), “li” (list items), “ol” (ordered list), “ul” (unordered list), “h1” to “h6” (heading levels from 1to 6) for example. The hyperlink feature was also introduced in HTML, which was missing in SGML, and this feature was introduced with an “a” (anchor) tag. The simplicity inherited from SGML, and the new additions to the language made it a first choice for a web development (Raggett, D. et al. 1998). Berners-Lee introduced the first version of the HTML with twenty elements (Shannon 2012).

After the release of the first version of HTML, browser vendors added their own versions of the language. Many of these different tags were then collected together, and the first draft of HTML 2.0 was released (Raggett, D. et al. 1998). HTML 2.0 was the first official release of HTML, and it was released in 1994; meaning that HTML 2.0 was the first standard markup language for web browsers. During that time, Mosaic Communications released the Netscape browser with a Graphic User Interface (GUI)(Bartels 2011).

The next version of HTML was a further extension of HTML 2 with new tags such as “fig” and “style”. A formal standard of HTML 3.0wasdeveloped by the International Engineering Task Force (IETF) built on the previous version and designed in such a way that it was backward compatible to HTML 2.0(Raggett, Dave 1995). One of the significant additions to the new version was the introduction of the “Table” element, allowing web pages to display tabulated data. It was the very first occurrence of page styling in web pages (Raggett, Dave 1995).

Released in December 1997, HTML 4.0 included the implementation of some new and very sophisticated mechanisms such as embedding objects, style sheets and scripting. It was designed to meet international standards, and provided the capability for websites to be coded in any language (Yergeau et al. 1997). HTML 4.0 supported the ISO/IEC: 10646standards as a HTML character set. In this version, there were additional features to enable people with disabilities to use websites (W3CRecommendation 1999a). There were other advancements in this version of HTML with the introduction of new tags such as “q” (quote), “acronym” (indicates an acronym in the text), “colgroup” (for more control on table formatting), “button” (can use to add push button) and many more (Charlesworth 2014).

Because of these advancements in HTML 4.0 the web industry adopted it very quickly. However, still there was further need of a markup language, which could achieve design goals such as:

- Design should be formal and concise (W3C 2008).
- The markup language must support a wide variety of applications (W3C 2008).
- The optional features must be kept to a minimum (W3C 2008).
- The design should be created quickly (W3C 2008).

All of these initial goals were achieved with the introduction of the new form of markup called Extensible Markup Language (XML), which is further discussed below.

2.2.3. XML and It’s need

As the Internet expanded to a bigger information canvas with involvement from a growing number of technical and not technical developers from different professions and interests building web pages, a need was identified for a more structured markup form. eXtensible Markup Language (XML) was introduced to meet this need. It was comprised of a number of technologies including XQuery, XSLT, XPath, which increases the human-readability of the XML written code(Kramis & Giannakaras 2008).XML does not require any other data model, and it can provide structured data directly by not even including any references, however like HTML, it also requires styling and linking standards. XML allows web page authors to compose their ideas into custom tags, and it provides a facility for the web developers can create their own tags in any phase of the development (Boeri, RJ & Hensel 1998). Further, a wide variety of information was available on the internet, and people used different search queries on the search engines. Therefore, XML provided a formatted standard structure of data representation, exchange and storage. It allowed the search engines to easily carry out the focused searching (Li 2009).

2.2.4. XHTML Markup Approach

In response to the popularity of XML, the W3C developed a new markup language which concentrated on both the HTML and an XML vocabulary called XHTML. As with XML, in XHTML, the document must be properly constructed, and each tag or element must be terminated explicitly (Beatty, Dick & Miller 2008).The W3C (2000) identified the benefits of using XHTML (W3C 2000):

- Documents were formatted with XML standards: they were readily edited, viewed and validated using XML tools.
- XHTML could utilize scripts and applets which were dependent on XML Document Object Model or HTML Document Object Model.

However, the biggest criticism received by the later version XHTML 2.0 was that it was not backward compatible with its previous versions of HTML (Carey 2012).

2.2.5. Next Generation Markup: HTML5

During the last decade, technology has evolved very rapidly, with the introduction of smart devices such as smart phones, tablet PCs, and televisions. These technological developments have created a demand for a markup language that can render information on both traditional and the latest in up-to-date devices. HTML 5 was developed with the intention to provide developers with the power to write a single code for multiple browsers on diverse devices (VisionMobile 2011). HTML supports the “responsive” web page development. Here “responsiveness” is referred to the ability of the HTML5 pages to adjust themselves according to different screen sizes on various devices (Firdaus 2013).

HTML 5 is created as a standard with features that are adopted from other associated technologies such as flash, CSS and JavaScript and so on. The “canvas” element is one of the features introduced in HTML 5. This element provides the ability to draw complex animation, graphics and diagrams. It is an easy-to-use tool that can be used to draw almost any type of 2-D digital graphics. JavaScript can be used with the “Canvas” element to draw graphics and images

(Martinet 2005). It is a low level bitmap tag, and graphics are generated within a specific height and width. JavaScript can be used to get the reference of the specified “Canvas” and the canvas can be customized easily (Hickson, I 2011).

Other useful advantages of HTML 5 are as following:

- HTML5 provides the capability of long-term browser data storage that was missing in the previous versions of HTML. The “Local Storage” feature presents a solution for web pages to use key/value pair to store larger amounts of data within the browser than the long used cookies (Hickson, L 2013).The key/value system is easily to store and retrieve, and key handlers can be register to analyze for changes to Local storage values (McArdle 2014).
- HTML 5 takes the advantage of Web SQL Database, which brings the capabilities of Relational SQL to the client side. WebSQL wraps around the database to interact with it, and JavaScript is used to access it efficiently (Hickson, I 2010). Moreover, all modern browsers are equipped with SQLite database for proper and well-defined data storage. The advantage of having database in the browsers is that there is no need of installing the database separately, and it is easy and convenient way of storing web content (Leblon 2010).
- HTML 5 is enabled with “video” and “audio” tags which can be used to play videos and audios respectively without any intervention of other third party languages (Daoust et al. 2013). But, still there are some conflicts among different browser vendors to select a standard codec for these two tags (Jakus & Sodnik 2010).
- HTML 5 carries elements that support location-based services (LSB) and newer formats like Scalable Vector Graphics. It provides Geolocation API to add location-based functionalities to the web pages. This API checks the position of the users by finding their latitude and longitudes by using mobile phone network or GPS (Popescu 2014).
- HTML5 has been designed to work across devices, whether desktop or mobile, and Apple Inc. shares a very big chunk of smart devices industry. Moreover, Apple devices do not support the use of Flash Technology by Adobe (Boeri, R 2011). But, the advantages of HTML 5 are well appreciated by the Adobe, and the company is regularly contributing to it. Adobe has changed its focus from Flash to HTML 5, and it is working with WC3 and regularly putting their efforts on HTML5’s webkit (Taft 2011).

Above are a few new additions to HTML 5. However, there is ongoing research occurring with the technology. Adobe is working on the “Cascading Style Sheet” module to give the web designers more power towards the streaming of HTML content, so that text can be wrapped around graphics. One of the related HTML 5 application development applications is PhoneGap: this is enriched with tools that accelerate the HTML5 development process (Taft 2011). PhoneGap is an open source and free framework, and that uses general web APIs for any platform the developer want to work on (Adobe 2014).

HTML 5 is the next generation gaming platform which can be successfully augmented with CSS and JavaScript to produce high quality and memory efficient games. HTML5 can be used with JavaScript libraries such as Quintus, which is lightweight, but at the same time a very powerful game engine created by Pascal Rettig (2014). Quintus allows the game developers to develop all sorts of 2-D games with HTML 5. Quintus and HTML 5 can also be used with a free program called “Tiled” that allows creating levels in a visual way (Navarro 2014).

2.2.6. Supporting Technologies: CSS, JavaScript and Flash

Like SGML, HTML 5 also requires other supporting languages to make web pages more interactive and dynamic for a high-quality user experience. Use of technologies like JavaScript, .net, AJAX and so on, play a critical role in new age web pages. These technologies equip the web browsers with the abilities to manipulate and compute web content, and allow them to transmit the changes (Mac Conaonaigh 2007). Important web technologies: Cascading Style Sheet (CSS), JavaScript and Flash will be discussed in the following section:

I. Cascading Style Sheet (CSS)

The styling language Cascading Style Sheets (CSS) evolved from discussions in the W3C during 1994 and 1995 (Lie, Hakon Wium 1994). Bos (1995) developed the main foundations for CSS1 and together with Lie developed the first W3C CSS recommendation (Lie, Hakon Wium & Bos 1996b).

CSS was developed mainly to provide the facility to separate the “Hypertext content” from “presentation document” (W3C 2013a).The CSS styling approach of separate style documents provides more accessibility, content flexibility and organization of presentation characteristics. CSS reduces the code rewriting and repetition of code structure as multiple pages and page elements can use same style sheets for multiple purposes (Shea & Holzschlag 2005). Essentially CSS replaced the use the of some irrelevant HTML elements by implementing formatting rule. For example, the style sheets might specify all the “h1” (main heading) to be bold. Therefore, no formatting elements like "<b></b>" (bold) is required within the content (HTML document) (Meyer 2000).

As new versions of HTML evolved, CSS has also become more sophisticated with advanced stylistic capabilities to fulfill the needs of modern web developers. During the initial stages, there were variations in implementation between browsers from different vendors that made it very difficult to produce a web page appearance (Cailliau 1997). But, constant efforts were made by all the vendors to achieve compatibility with the CSS standards. The Macintosh edition of Internet Explorer 5.0 which was deployed in year 2000 was one of the examples of how Microsoft (one of the leading browser vendors) implemented full support to CSS1 (W3C 1994).To date CSS have released different versions of CSS that include:

- **CSS 1-**This version supported basic styling of Fonts, coloring of backgrounds, texts and other elements, text styling such as spacing, and text aligning (Lie, H.W. & Bos 2005). It also supported image and table aligning. Concepts like "padding," "border," "margin" and "positioning" were also introduced in this version (Lie, Hakon Wium & Bos 1996a).
- **CSS 2-** In 1998, the W3C developed the CSS 2 specification as a recommendation. CSS 2 was a subset of CSS 1 and included new capabilities such as z-indexing, fixed, relative and absolute arrangements of HTML elements and new styling properties like "shadows" (Bos et al. 1998).
- **CSS 2.1-** Cascading Style Sheets level 2 revision 1 fixed all the errors in CSS 2. The poorly supported interoperability feature was removed, and new browser extensions were implemented. After many drafts, in 2011, CSS 2.1 was published as W3C recommendation (Bos et al. 2011).
- **CSS 3-** CSS3, introduce in 1999, added a modular approach to developing style documents. Each module adds new functionality, and they help maintaining backward compatibility. There are more than 50 modules published, with four of these modules published as W3C recommendations as listed below.

a. **Media Queries-** These allow the web content to be rendered on different devices with various screen sizes (Rivoal 2012).

b. **Namespaces-** It is a rule that restricts different types of selectors such as universal, type and attribute to choose elements under itself (Etemad 2011).

c. **Selectors Level 3-** These are pattern that can be used to bind the HTML or XML elements to bind with the style (Çelik, Etemad, et al. 2011).

d. **Color-** CSS provides different color related properties that can be used in the styling of the web content (Çelik, Lilley, et al. 2011).

II. JavaScript

JavaScript was developed by Netscape Communication Corporation computer scientist, Brendan Erich. Netscape developed JavaScript as a client-sided language that could run on all operating systems. It was a lightweight language which complemented Sun Microsystem's Java (Charles 2012). Initially, it was named *Mocha*. However it was officially called *LiveScript* when its first Beta version was released (Netscape 1995). The release of the Netscape browser version 2.0B3 saw this language renamed to *JavaScript.* The name change was considered as Netscape included the features to support the Java language in their new web browser (Andreessen 1998).

JavaScript is a dynamic scripting language used within web browsers to implement client-side interactions (Flanagan 2011). It is also a useful tool for game development, server side network programming and desktop and mobile application development (Flanagan 2011). JavaScript is specified as a "first-class functions" language, i.e. in JavaScript the functions can be passed in other functions as an argument, can be returned as a value from a different function and functions can be held in different data structures (Abelson, Sussman & Sussman 1996). It is a mix of features which defines it as a multi-paradigm programming language including object-oriented, functional and imperative approaches (ECMAInternational 2011).

A popular use of JavaScript is to add dynamic client-side features to static HTML pages. The scripts can be added or embedded to the HTML pages and can work along with Document Object Model (DOM)(W3C 2005). JavaScript can load new information to a page or add content to the server without reloading the page with the help of "Asynchronous JavaScript + XML" also known as AJAX. JavaScript can also be used to add page animations such as fading in and out, moving the content according to the screen size and resizing content, etc. JavaScript can be used to build interactive content like games, play video and audio and many more with the help its virtual machines also known as JavaScript engines (Shankland 2009).

Between 1996 and 1997, European Computer Manufacturers Association (ECMA) took the JavaScript to develop a standard specification. The motivation behind the standardizing JavaScript was that the web browser other than Netscape could implement it (Standard_ECMA-262 1999). The work done in this period resulted in the release of ECMAScript; with JavaScript was the basis of the implementation. The standardization process was further carry on, and ECMAScript 2 was released in 1998 and ECMAScript in 1999, which set the baseline for current JavaScript (ISO/IEC 2011).

III. **Flash**

When the web was first developed, Flash was the core multimedia platform for web based game development. It was originally acquired by Macromedia and is currently owned and developed by Adobe Systems. Flash was introduced to the market in 1996, and since then it has been the most-popular platform for adding animations and interactive features to online pages. It is popular for carrying out operations like creating advertisements, interactive animations, embedding videos and developing web based games (Sutopo 2010).

The interactive content developed in Flash Technology can be played, executed and viewed by using Adobe Flash Player. In 2009, flash player for Smartphone device was made available by the Adobe (Offerman & Hunter 2009). However, after the release of version 11.1, Adobe halted working on the Flash Player for smart phones and continued with Adobe AIR for Smartphone applications (Winokur 2011). Furthermore, at the end of 2011 Adobe ended its services on “Mobile Platform Flash”, and started shifting its focus to HTML5 and Adobe AIR (Vaughan-Nichols 2011).

Currently, there is no software available that provides all the features that Adobe Flash delivered for animation and interactive content. However, when HTML 5 was introduced, it was considered as an alternative to Flash implementation on web pages. In an extension to this, Google has introduced a tool called as “Google Swiffy” to transform Flash content into HTML 5 content (Google 2013).The transition from Adobe Flash to HTML 5 is discussed in the next section.

2.2.7. Will HTML5 completely replace FLASH?

HTML 5 and Flash both support the embedding of video and audio multimedia content in web pages. Both technologies also support the integration of vector graphics. Therefore, HTML 5 could be a good alternative for replacing Flash from web pages (Bilton 2010). Web browsers are not capable of rendering the flash content by themselves; rather they need to install Flash player as a plug-in to execute the Flash media. HTML 5 is a solution to this problem as it is a markup language that is supported by all the web browsers without the need for extra plug-ins. So, the animations and multimedia content developed in the HTML 5 can be rendered on different browsers and diverse devices (Scott 2011).

Though development of animations and flash websites are easier to develop in flash by using Adobe tools, however, these tools are proprietary software and expensive to buy (Wayner 2010). On the other hand, HTML 5 stack (CSS and JavaScript) are recommendations by W3C. These technologies do not require any special integrated development environment (IDE). Hence, the development in HTML5 is becoming popular as dynamic web development is cheaper in HTML5 compared to Flash (Boeri, R 2011).

Flash has an advantage over HTML 5 in providing cleaner and crisper websites because it can define the measurements in the form of sub-pixel increments. Whereas, the websites built on HTML5 and CSS are measured on sub-pixel scale, and browsers round up or down the measurements to the nearest whole number. This results in inconsistent measurements and hence unreliable page display(Wayner 2010).

Alongside these advantages, there is a big disadvantage as well that makes Flash the last option for web development. Since the beginning of Flash, it gained a bad reputation in Search Engine Optimization (SEO) (Fishkin 2008). The primary reason behind poor performance of Flash websites when in comes to SEO is that usually Flash content is loaded with interactive features and images and less with plain text. On the other hand, leading search engines mostly emphasises on text to interpret the meaning of the web pages (Hochman 2009). This makes Flash a poor choice for a web development for the commercial and business websites.

Both the technologies have their good and disadvantagious qualities. But, HTML 5 has shown that it can perform a lot of operations that Adobe Flash can do to develop a web content and pages.HTML5 can completely replace Flash if continuous work is carried out to improve the markup language, and every browser starts adopting its features. Though Flash may disappear from web pages, it willnot disappear entirely as it has many other functions other than the web designing such as game designing and animation, and it is predicted that the industry will continue to use it in one or another way (SooHoo 2012).

2.3.Social Gaming

Digital gaming has become very popular, and it is a multi-billion-dollar business. Social gaming is an emerging sector within the gaming industry (ESA 2014). Social Gaming is considered to be simple with game play and game control because most of the social games are played by casual gamers who want less commitment towards these games (Wallace & Robbins 2006).With the introduction of technologies like HTML5, it has become very easy to spread social games to various platforms like personal computers, mobiles, consoles and other digital devices. The following discussion will define social gaming, provide examples of some of the different types of social games and discuss some of the important design considerations important for this type of gaming.

2.3.1. Defining Social Gaming

In the multimedia age, Social Gaming refers to playing digital games on a social network, and sharing the gaming space with people on the network as a method of social interaction (Cutler 2012). It is different from the solitude gaming where the computer is the only competitor.Social gaming came into attention approximately a decade ago with the introduction of gaming applications on popular social media sites like Facebook (Cutler 2012). To develop social games, various trends were observed, and social game development has increased exponentially with the introduction of additional network platforms, high-performance mobile devices and an increasing shift from gambling to social gaming (CrowedPark 2012).

2.3.2. Types and Examples of Social Games

In the year 2012, 250 million people played social games on Facebook per month, and approximately 100 developers raised around $1 million in revenue(Hockenson 2013). These statistics make it clear that social gaming will maintain its position for a long time until people will find other ways to entertain themselves. Below are afew examples of different types of social games.

- **Social Role Play Games (RPGs):** Social RPGs take concepts from regular video game RPGs. All of these games works on the same rule that players perform “Jobs” or “Quests” to earn virtual money, and then use that money to unlock new levels and features. Some examples of Social RPGs include Hillfolk, Burning Wheel, Legend, Mouse Guard & Torchbearer and many other (Neslon 2013).

- **Sports RPGs:** These games are very similar to social RPGs, but the main focus is concentrated on competition-oriented scenarios. Examples of popular sports RPGs are Football Manager, Tennis Mania, Penalty Shootout 2012 and Billiards Master Pro (Kaiser 2013).
- **Casual Games:** In casual gaming the game is developed such that it will be used by a huge audience on a casual basis. These games require less commitment as they have small missions and game play. Examples of casual games are Missing Magician, Picture Shifter, Barry Smart, Wild Wild Cow and many other (Boyes 2008).
- **Word Games:** Word games are a special part of casual games because they try to attract those players who play less frequently in their free time. These games provide mathematical and logical scenarios to solve. Examples include Word War, Lolly Balls, Don't Cross the line (Nichol 2012).
- **Virtual Villages:**This type of game is based on the idea of developing structures in a defined area or performing some labor tasks. The player'objective is to construct some infrastructure in a designated area. Games of this type include Pet Society, YoVille, myFarm (Humphrey & Humphrey 2010).
- **Casino:**These games are based on the real world Casino Games. There are lots gambling and non-gambling Casino games. For example, Texas Hold 'Em, Playdom, Poker Palace a (Kavouras 2013).

2.3.3. Understanding the purpose of games and Audience's needs

There are lots of players who pay money to play social games, and very few games provide an opportunity of material gain. So, why do people payto play these games? Social Games are entertainment products, and gamers pay real money to buy virtual currency and goods to entertain their brains subconsciously. These purchases unlocks new levels and features in the game, and this plays a crucial role in intensifying the game play (Gatto, Esplin & O'Reardon 2014). In return, gamers expect quick game progression, clear progress and new challenges (TheEconomist 2011).

Boswell (2013) writes that the reason behind the exponential growth of social games is that they are strongly irrational and are powered by social rewards that are highly engaging (Boswell 2013). People are spending their time, money and efforts to achieve highest scores because they always look for a friendly and competitive environment where they can share their accomplishments to gain social status (Wang 2011). It is very important for the game developers and producers to understand that it is a human need to socialize with each other. In the current scenario, there is a rise in rationality, and hence, people are diverting their lives towards gaming.

Social games developers have made a real connection with audiences. Considering human psychology in game development is definitely paying off.

2.4.Game Development with Engines

Game Engines are reusable components that act as software frameworks, and are used to create games (Cowan 2014). Game functions such as graphical user interface, sounds, collisions, visual, artificial intelligence and many more can easily be implemented by game engines (Ward 2008). These frameworks are set of tools that can be implemented directly in the game development process reducing the efforts to develop the games. These engines provide an easy and reusable platform with important game development tools, hence reducing the cost of development (Cowan 2014).

2.4.1. A glance at some useful Game Engines

There are lots of game engines written in JavaScript and other languages that can be used with HTML5 to build platform independent games. A number of the game engines are discussed following.

I. Impact- It is a JavaScript framework that facilitates the developers to construct HTML 5 games for Smartphone and desktop browsers (ImpactJS 2014a). The game engine provides different layers to work on, and each can be stacked on top of other. Each layer can be optimized with unique behavior and patterns, and one of the most useful layers is "collision" which adds collision behavior to the game (ImpactJS 2014c). Code organization is a big issue in JavaScript game development; however Impact provides a module system which takes all the effort of code keeping and organizing itself and let the programmers focus on game development.

   A Module system boosts the performance of the games and increases efficiency of the development lifecycle (Szablewski 2011). Impact provides a very sophisticated debugging feature that allows the developers to look into the game play while simultaneously analyzing the debug module containing Background Maps, Entities, and Performance panels. These defaults panels can be accompanied by other useful panel as per the game type and needs (ImpactJS 2014b).

II. PlayCanvas- PlayCanvas is an open source game engine having effective tools to develop 3-D games (Kalpias 2014). The game engine is a web hosted and can be used on the cloud and also installed on a personal workstation. As, it runs entirely on JavaScript and is 100 percent dependent on HTML 5, games developed on PlayCanvas can be run on cross platforms. Developers can build 3D models and export these to FBX file format, and then this file can be used in PlayCanvas with the help of drag and drop features (PlayCanvas 2014a). PlayCanvas provides a cloud environment, which can be used from any remote device using the user credentials. Any asset developed in any 3D content creator can be uploaded to the PlayCanvas cloud asset library (PlayCanvas 2014b).

III. Turbulenz- Turbulenz is a game engine specially designed to develop social HTML 5 games. Turbulenz provides a Software Development toolkit (SDK) which includes local gaming server, documentation, asset processing tools and samples (Turbulenz 2012b). It has a JavaScript engine, which contains all-important services such as graphics, physics, audio, animation and networking. Turbulenz is also provided with APIs to provide functionalities such as a leader board, save games, badges, friend list, chats, payments and challenges (Turbulenz 2012a).

Turbulenz provides SDK and game engine so that the developers can develop the game locally on their workstations. However, “developer hub” feature helps to test the scalability and performance of the games. In the hub, the developer can integrate their games and distribute the results with other team members, tester and guests (TurbulenzLabs 2011). Turbulenz is a growing technology and funded by EA sports, Play Station, Apple, Google and many more (Turbulenz 2011).

IV. CoCos2d-JS- It is facilitates the game developers to develop games for HTML5 compatible browsers and JavaScript Bindings (JSB). The APIs used in Cocos2d-JS are derived from Cocos2d-x. Cocos2d-JS consists of Coco2d-html5 and Cocos2d-x JSB (COCOS2DX 2014b). Cocos2d-x is an MIT licensed (Massachusetts Institution of Technology) and open source game engine. Developers can develop interactive cross browsers games, apps and other interactive programs (COCOS2DX 2014a).

Cocos2d-JS is supported all modern browsers such as Firefox 3.5+, Safari 5.0+, and Chrome 14+. It is compatible with iOS 5.0+, Android 2.3+ and Windows Phone 8+. It is also compatible with desktop computer operating systems including OS X v10.6+ and Windows 7+ (COCOS2DX 2014b).

V. Game Closure- Game Closure is developed entirely on JavaScript. It is specially designed to work with HTML5. Game in Game Closure can be developed in any text editor and can be tested on almost all web browsers. This gives developers the opportunity to use advanced debugging tools present in browsers like Google Chrome and Mozilla. The developers can write a single code for iOS and android and can use a single command to create an installation package for respective operating systems. Game Closure provides the development kit known as DevKit, which have a web simulation for instant testing of the games for iOS or android environment (GameClosure 2014).

VI. Gamvas- It is an open source game engine licensed under MIT. It is a 2D game engine specially designed to work with HTML5. Gamvas is written in JavaScript, and it is best suited for commercial use. It has Box2D physics simulation, which allows the game characters and objects to having attributes such as friction, mass and electricity. Gamvas supports resource management that tells the developers about number of resources they have used. On the basis of this information, the developers can design a loading screen in the game. This loading screen may show a “progress bar” to depict the time for loading the game. It also supports camera feature that allows the use multiple cameras in different directions and rotational axis (93-interactive 2014).

VII. KIWI.Js- It is an open source game engine specially designed to develop HTML5 mobile and desktop games. KIWI.JS supports WebGL for 2D and 3D rendering. It uses cocoonJS to create and publish the HTML5 games (KiwiJS 2013). CocoonJS is a platform that facility the developers to manage quality assurance and deployment process. It especially helps the developers to design the games, and also provides easy-to-use testing and debugging option (ludei 2013).

VIII. CraftyJS- CraftyJS is developed in JavaScript and supports HTML5. The developers have the choice to choose “Canvas” or Document Object Model (DOM) which allows to developers to create the games with best-suited technologies (canvas or DOM).The advantage of developing the game in CraftyJS is that the file size of the source is small enough that the developer does not need to worry about managing the network bottleneck. It supports Entity component system as the software architecture pattern (SOP). SOP is an implementation of composition based programming using a database structure rather than implementing inheritance (Nystrom 2014). CraftyJS also supports “sprite map” that is very useful for creating game entities (Craftyjs 2013).

IX. LycheeJS- LycheeJS is also a JavaScript Engine for developing 2D games. It supports canvas, Simple DirectionMedia Layer (SDL), OpenGL and WebGL for animation rendering. LycheeJS allows the development for almost all modern browser and hardware platforms (mobiles, gaming consoles and Desktops) (LazerUnicorns 2014b). It contains the bitmap font engine which facilitates the developers to create text font easily from the existing pixels present in the memory (LazyFoo' 2014). However, the drawback of using LycheeJS is that it do not provide 3D animation support, moreover, the graphics quality are on a very basic level (LazerUnicorns 2014a).

X. melonJS- melonJS is also an open-source JavaScript engine. It supports multi-channel HTML audio for embedding the sounds. It is provided with a “state manager” easily to manage game states, menu, loading and state switching. melonJS also have a feature for transition effects that is very important in the game visuals and animations. It is also provided with very sophisticated math API for vector and matrix elements. However, the disadvantage of melonJS is that it does not support WebGL, and there is no direct support for 3D animation (melonJS 2014).

From the analysis of different game engines, a comparison matrix is drawn.

Table 1 Comparison Matrix

Note: In matrix below some of the cells are highlighted with a comment "Not Mentioned". The comment does not specify that the functionalities are not present in the engines. Rather this comment specifies that the respective functionalities are not documented in the official specifications of the game engines. The online documentations and websites of the different game engines give broader highlights of important functionalities.

S. No.	Game Engine	2D Rendering	3D Rendering	Sound	Physics	Collision	WebGL	Platforms	Advantages	Disadvantages	Cost	Reference website
I.	Impact	✓	With the help of the third party frameworks 3D games can be developed	✓	✓	✓	✓	HTML5 games for mobiles and desktops	Uses CocoonJS, Accelerated HTML5 Execution, Testing and debugging, very good documentation for the beginners to develop professional games	No direct implementation of 3D	USD 99	(ImpactJS 2014a)
II.	PlayCanvas	✓	✓	✓	✓	×	✓	HTML	Support	Does not have separate engine for collision	Free	(PlayCanvas 2014b)
III.	Turbulenz	✓	✓	✓	✓	✓	✓	HTML5, iOS, Android	Support debugging and networking	No proper beginners guide	Free	(Turbulenz 2012a).
IV.	Cocos2d-JS	✓	3D Character support	✓	Not Mentioned	Not Mentioned	✓	Windows, Linux, OS X, iOS, Android, BlackBerry, Tizen	Multichannel sound, In-app purchase, multiplayer, push notification	There is no proper documentation for the 3D implementation	Free	(COCOS2DX 2014b)
V.	Game Closure	✓	Not Mentioned	✓	Not Mentioned	Not Mentioned	Not Mentioned	iOS, Android, HTML 5	Supported on all platforms	Lack in major functionalities	Free	(GameClosure 2014)
VI.	Gamvas	✓	×	Not Mentioned	✓	✓	Not Mentioned	HTML 5	Path finding 2D grid, Box2D, Camera, Resource Management	Does not have specifications for sound, No support for 3D	Free	(93-interactive 2014)
VII.	KIWI.JS	✓	Difficult to add 3D graphics	✓	✓	Not Mentioned	✓	iOS, Android and Chrome	Physics and plug-in infrastructure, Uses cocoonJS, Multi-touch,	3D rendering is complex; Collision is not mentioned in the documentation	Free	(KiwiJS 2013)
VIII.	CraftyJS	✓	×	✓	Not Mentioned	✓	Not Mentioned	Cross Browser Compatible	Compatible with bower, supports both Canvas and DOM, Sprit Map Support,	No support for 3D, No proper documentation for physics engine	Free	(Craftyjs 2013)
IX.	LycheeJS	✓	×	✓	✓	✓	✓	Cross Browser Compatible, Mobile, console and desktop development	Web socket engine, Practice engine, Remote Debugging	No support for 3D, poor graphics	Free	(LazerUnicorns 2014b)
X.	melonJS	✓	No direct support for 3D	✓	✓	✓	×	Compatible with major browsers and mobiles	Sprit-based engine, Standalone libraries	No built-in support for WebGL	Free	(melonJS 2014)

The columns in the matrix are set according to the need of this project and the basic features that required being develop the expected game. It is clear from the matrix that the Impact Game Engine is having all the necessary features and stands out of rest of engines. The game engine is not a freeware as it cost USD 99, however it uses CocoonJS, accelerate HTML5 execution, testing and debugging. It supports 2D, support for sound, physics and collision engines.

2.4.2. Other Game Development Platforms

There are many applications that also use game engines to provide development environment to build the games (Lee 2010). Few of them are discussed below.

I. Construct 2- Construct is a game development tool specially designed to work on "Windows". It is a powerful tool for creating 2D games (Scirra 2014a). Developers do not need any specialized experience, education or programming knowledge. It is designed to be visual, intuitive and instantly understandable. Making a 2D game is simple in Construct 2, as the developers can start with an idea, can use drag and drop feature, introduce behaviors and events (Scirra 2014b).

II. CREATEJS Suits- CreateJS is an open source suite which is sponsored by Adobe, Microsoft, AOL, Mozilla and gskinner. It provides JavaScript tools and Libraries for creating interactive games that can be published on the web via HTML5. It has four major libraries which include EASELJS, TWEENJS, SOUNDJS and PRELOADJS. (CreateJSSuits 2014). These JavaScript libraries can be used to develop enriched and interactive HTML5 games.

    The EASELJS module has a feature to offer retained graphic mode that can be used with HTML5's "Canvas" element. It provides a complete hierarchical display, helper classes for 2D graphics and a core interaction framework which is very easy to implement with "Canvas" (EaselJS 2014). The TWEENJS module offers simple and powerful "tweening" interface (TweenJS 2014). "tweening" is a short form for in-betweening. "tweening" is a key animation process of producing the sandwiched frames present between two images which helps one image to evolve smoothly into next (Beal 2014).

    SOUNDJS is another important JavaScript library present in the CreateJS suit. SOUNDJS adds audio and playback effects in games that can be used with the help of plug-ins. These plug-ins can fetch original audio implementation. It is easy to implement sounds on any platform without having any knowledge about the playback mechanism for that platform (SoundJS 2014). Another very important module on EASELJS is PRELOADJS. It is very important for maintaining the performance by quickly loading the content and scripts. PRELOADJS gives an easy means to preload the game content on the browsers by using XMLHttpRequest (XHR) and HTML5 tags. It is fully functional with all the modern web browsers (PreloadJS 2014). Hence, CREATEJS can be used as a framework to build high end multi-platform video games.

III. pixi.js- Pixi.js is famous for its ability to produce speedy 2D rendering (Pixi.js 2013b). Pixi.js is a rendering engine which supports cross-platform game development and WebGL (Pixi.js 2013d). Pixi.js also supports multi-touch interactivity for the touch-screen smart devices. Pixi.js supports different visual effects such as tinting and blends modes. These features are well supported by canvas and WebGL renderers with no compatibility issues (Pixi.js 2013c). Above features and other useful attributes such as “sprite sheet” support, “hierarchical tree organization”, easy API support, “asset loader”, “renderer auto detect” and text support makes Pixel.js a popular choice among the game developer (Pixi.js 2013a).

IV. GameMaker Studio: GameMaker Studio is best-suited for novice and entry level game developers. It is a fast-to-develop framework which provides cross-platform game creation. Before creating the games, the developers can create the prototypes quickly and can draw a basic idea about the yet-to-develop game (YOYOGAMES 2014e). GameMaker is equipped with an integrated development environment (IDE) which can help developers to develop professional games without any understanding of programming. The IDE also provides Game Maker Language (GML) and in-built Scripting language. It provides a hierarchical structure for game assets such as images, animation, audio and fonts. The IDE supports testing of the game on multiple devices either through wired or wireless connections (YOYOGAMES 2014d).

GameMaker Studio provides export feature to deploy games for consoles, mobiles and desktops in the form of ready-to-run executable apps. It can be published online by exporting the game in HTML5 (YOYOGAMES 2014a). GameMaker is equipped with YoYo Compiler (YYC) which harnesses CPU resources efficiently and eventually increases the performance of the games (YOYOGAMES 2014f). GameMaker is best-suited to develop commercial games, and it helps monetize the games. Developers can easily connect their games with big advertisers on mobiles and across the internet (YOYOGAMES 2014b). GameMaker is an efficient development tool to develop high-end commercial games as it provides feature to analyze the audience interaction and keep players engaged with the games (YOYOGAMES 2014c).

2.5.HTML5 Games Efficiency and Performance

Throughout the history of the internet, third-party browser plug-ins such as “Flash Player”, “Skype Click to Call” and many others have played an important role in enabling a web-browser to present the online multimedia content to the users. However, plug-ins have disadvantages as they run with the help browser support and use extra system resources. Moreover, they make the system more venerable to external security attacks. The other disadvantage of plug-ins is that they are not designed to support touch response that is very crucial in modern day touch devices. Also, plug-ins are not open source, but proprietary technologies that make it difficult for browser vendors to create their browsers to supports all kinds and types (Microsoft 2014).

The solution to modern day touch-screen and browser compatibility issues was initially added in the HTML5 specifications. This feature in HTML5 specification is known as “HTML5 Storage”, “Local Storage” or “Web Storage”. However, due to internal decisions within W3C, “Web Storage” was then defined in its independent specification (Pilgrim 2011). “HTML5 Storage” is a process in which web pages can locally store data key/value pairs on the client browser. The key value pairs persist even after the user moves to another tab/window or even after closing the web browser (W3C 2014).

HTML5 game engines harness the “Web Storage” property that HTML5 provides. “Web Storage” effectively partitions local persistent storage and session storage and it use larger client-side storage as compared to cookies (Vanghan-Nichols 2013). “Web Storage” provides a capacity of 5 Megabytes per domain for local storage. The advantage of using “Web Storage” is that it offers effective partitioning of persistent and session storage and allows the browsers to store only those data items those are important for local use. Data items that are transient and not relevant after their access are not stored (GWT 2014).

Offline applications and games consist of only one HTML5 pages. As soon as the page is loaded the data items that are necessary for the offline games are stored in “Web Storage”. This process reduces the overhead created by multiple page request-response cycles (Panigrahy 2012). This allows the games to run smoothly without worrying about loading the data from network and resulting lag. Hence, this feature allows the HTML5 games to run smoothly with high performance and efficiency (GWT 2014).

2.6.Literature Review Findings

The research focused on the importance of HTML5 in modern web development. It is a leading technology in game development and has cross-browser support. From the study, it was found that HTML5 is among the top technologies today used to develop platform-independent web content. It also provides features to embed animations and multimedia content directly into browser content without the need of third party plug-ins. Along with the above attributes and features like “web storage”; HTML5 is also a good option for development of online games.

Study found that social gaming is becoming popular activity among mobile and desktop users. There are many game engines available to develop these games. During the latter part of the study, ten game engines were analyzed, and comparison matrix was drawn. It was developed to find the best game engine among all. The assessment criteria were set as per the development need.

As the part of this project, it was decided that the game will be developed in the next half of the thesis. The audience of the game will be children of age 3 to 8 years. The movie behind choosing this age group is that, the interactivity level for the game of this group should be very high to keep the audience paly it for log time. This will help to test the abilities of all the used technology and their efficiency.

The game will be an educational game that will teach them about some basic mathematics in a fun and interactive way. To develop this game, “Impact” game engine was considered as it has necessary features that are required to develop this game.

Impact provides detailed documentation that explains all the methods, classes, APIs and internal code. It also provides a number of tutorials for the developers to adapt to all the functionalities. Two complete textbooks named “HTML5 Game development with ImpactJS” and “Introducing HTML5 Game Development” are also available as a complete reference and user guide.

The columns in the matrix (Table 1) are set according to the need of this project and the basic features that are required to develop the expected game. It is clear from the matrix that the Impact Game Engine has all the necessary features and stands out of rest of engines. The game engine is not a freeware as it costs approximately USD 99.However it uses CocoonJS, accelerate HTML5 execution, testing and debugging. It supports 2D, support for sound, physics and collision engines. All of these features will play a vital role in accelerating the development process in later half of the project.

3. Methodology

System Development research methodology (Nunamaker et al 1991) is used as the methodology for this project. This approach is an appropriate framework around which the project could be structured. It includes the process of gaining conceptual knowledge, building the system architecture, analyzing and designing of the system. Further focuses on building of the prototypes and required system. It also encompasses the process of observation evaluation and testing (Nunamaker et al 1991).

The following diagram shows the representation of the process involved in this methodology –

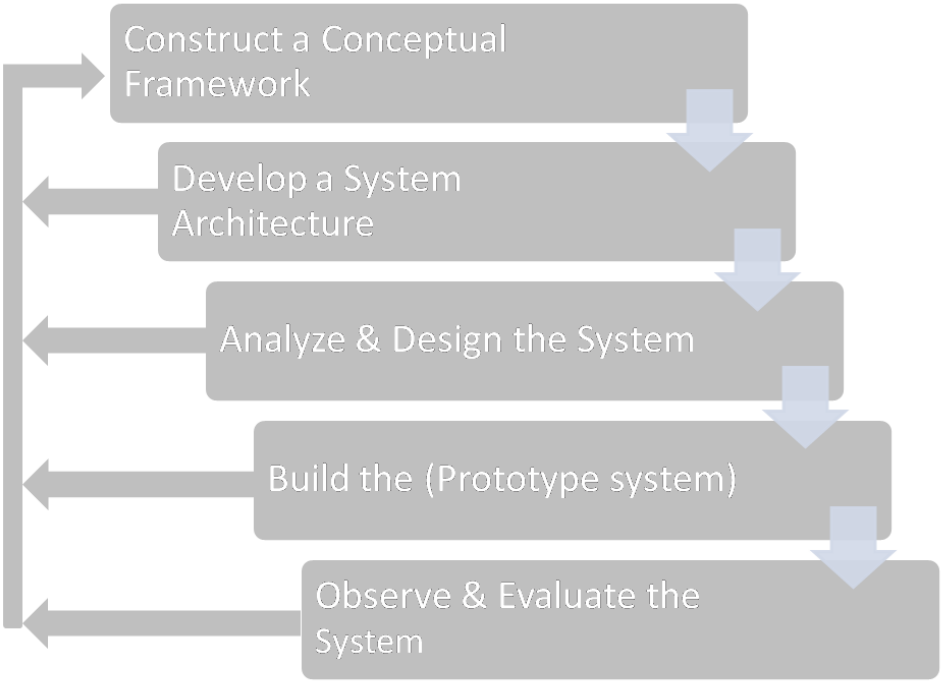

Construct a Conceptual Framework
Develop a System Architecture
Analyze & Design the System
Build the (Prototype system)
Observe & Evaluate the System

Figure 2: The Process for Systems Development Research (Nunamaker et al 1991)

3.1.Construct a Conceptual framework

Computer games are a popular means of entertainment for both adults and children. Their use in the education sector has become more widespread in recent years, and educating through games is now increasingly used to enhance the learning experience of students (Prensky, 2005; Kiili, 2005).

From a software engineer’s perspective it is crucial to use the most current and sophisticated tools and techniques to develop any product.

Moreover, the process of development of a conceptual framework for a social educational game requires multiple aspects to be considered such as technologies, frameworks, tools, domains etc. This framework helps the developer to learn and know about the essential elements of game development (Fisch 2005).

The conceptual framework used in this project is taken from the work of Yusoff and team (2009). The figure below illustrates the features of the framework that were an important part in this game development process.

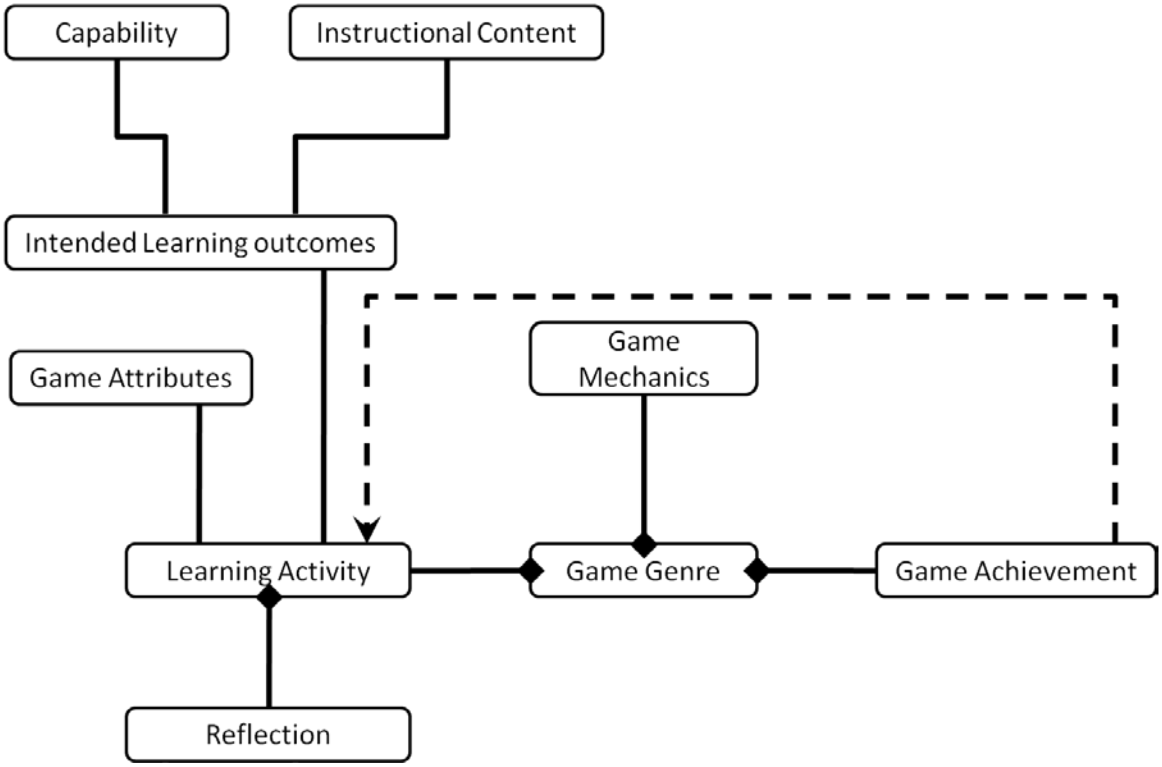


Figure 3: Conceptual Framework for Educational Game (Yusoff et al. 2009)

3.1.1. Capability

The intention of the game (MathMe) is to teach children the necessary skills to perform elementary mathematical operations such as addition (+), subtraction (-), multiplication (X), division (/).

These skills include:

- Analysis- The ability for children to analyze what a particular operator is supposed to do. For instance, if 2 is subtracted from 4 then what will be the result.
- Recall- It is necessary to recall what they have learned in the previous experience. This is achieved in the game by repetitively asking them addition, subtraction, multiplications and division questions as part of their progress through the game.
- Evaluation- Each time the child gives an answer; it is evaluated by the algorithms written specially for that purpose.

```
createQuestion: function(){

        **
**
        **
        **

if(correctAnswers>0){

correctAnswers--;

randomOwl.bonus=Math.floor(Math.random()*4);

randomOwl.updateNumber(this.answer,true);

}

else{randomOwl.updatNumber(this.getARandomNumber(1,20,this.answer),
false);}
this.owlsInGameCopy.splice(randomOwlIndex,1);

           i--;},
```

Table 2: The Logic behind the calculating the correct answer

The above algorithm shows that how MathMe Game evaluate the correct answer and rewards the children for it.

4. Effective execution of the Tasks- in the game the main objective is to hit the Owls with correct answer. Whenever, the player performs this operation they are awarded with points and frequent random special rewards.
5. Proper Attitude- The main aim of develop this game is to take off the fair of mathematics from the minds of young children. Therefore, the game is designed and developed in such a way that it is it interactive, colorful, animated and fun to play.
6. Logical thinking- The aim of the project is to teach the children the logic behind the mathematics and, thus building strong skillful foundation. Rather, then repetitive questions, the game always generates unique set of problems. This eliminates the opportunity for the children to memories the answers for the problems.   The below lines of code shows that how randomness of the problems are achieved:

```
var op1 =1+Math.round(Math.random()*9);

var op2 =1+Math.round(Math.random()*9);

this.answer= op1 + op2;

**

**

**
```

Table 3: Randomness in the game

3.1.2. Instructional Content

An important aim of this game is that it will help young children to learn as they play it. It was necessary to ensure that the content helped the learning process to take place. This includes the type of activities, events and challenges the children will face while playing the game.

There are three areas in particular that have been addressed and improved as development progressed:

- Round time- the time is takes to complete a single round of the game is very important instructional information. Initially, each round in the game was clocked to 100 seconds.

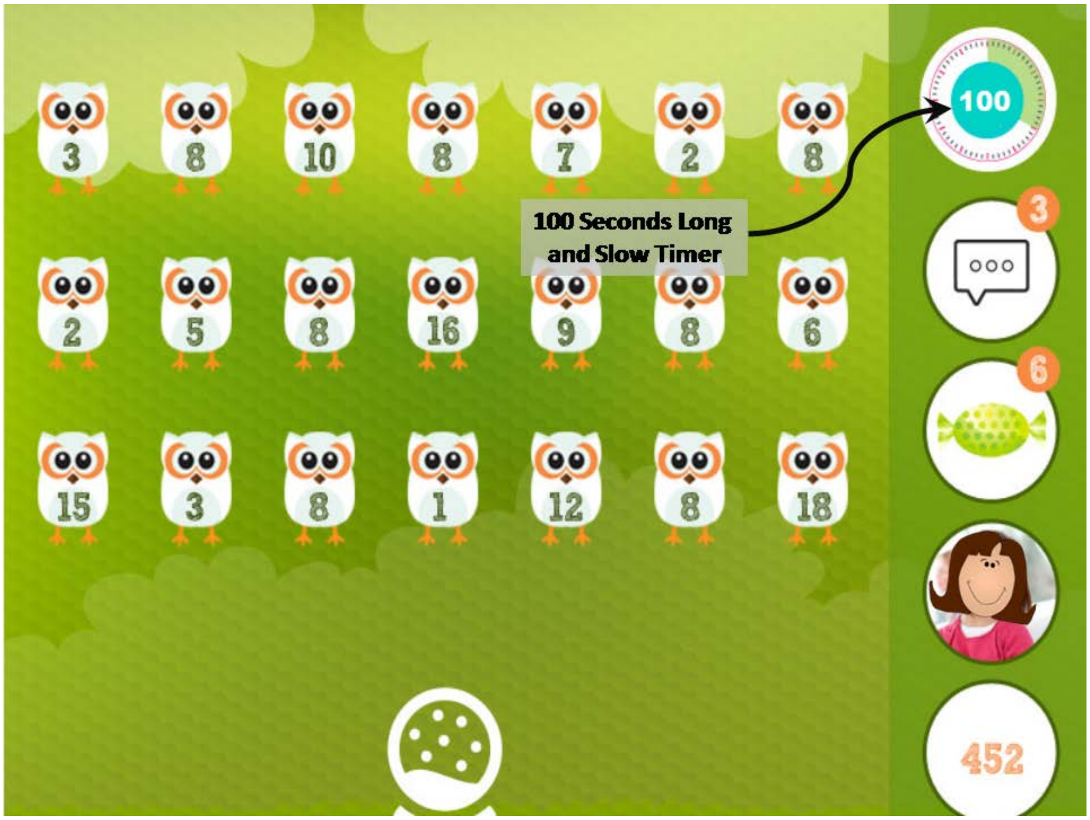
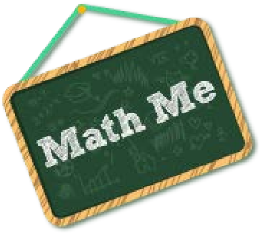
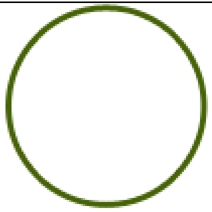


Figure 4: The game screen showing long and slow Round Timer

The above screen shows the maximum timer size, i.e. 100 seconds. This may be too long for young children. As the methodology allows multiple iterations, this timeframe was later shortened in a following stage of the development lifecycle.

- Shapes size and visual representations- young children are not proficient readers and lack the ability to understand complex written instructions. Therefore, big buttons and simple navigation is provided to help them understand how to use the game.

- The table below describes the visual components included in the game:

Component Name	Component Design	Author
Ball		Designed by Freepik.com http://www.freepik.com/free-photos-vectors/background
Help Answered Button		Custom Icon Design http://www.customicondesign.com/
Help Asked Button		Created in Photoshop
Math Me Logo		Created in Photoshop
High Score Background		Created in Photoshop
Game Owl Tiles		Created in Photoshop
Other Buttons	Top 10! Top 10! Top 10!	Created in Photoshop

Background Image		Created in Photoshop

Table 4: List of Component Designs used in MathMe Game

- Textual Information- Continuing from the above point, it is already explained that young children are less able to read text. Therefore, the approach used is to provide a simple graphical interface that contains minimal text. However, the parents are provided with all the necessary textual information that is required during and after the registration for the game.

3.1.3. Intended Learning Outcome

The intended learning outcome is that the game will help young children to learn basic Maths and also those they find using it both interesting and rewarding. As the figure 3 shows (Yusoff et al. 2009), learning outcomes are tightly coupled with content of the game and determined capabilities. It is not necessary that each child will gain the same level of skill in same amount of time. It will be quite irrational to measure the success of the game on the bases of studying individual need. Rather, it was realized, in the very early stages of the game development that interactive and educational content is necessary for the children of a very young age.

The foundation to mathematics is addition, subtraction, multiplication and division operations. The intended learning outcome of this game is to help children become more confident in these operations. Other main objective of the game is to teach children the importance of socializing within the society. There are two special features to the game that motivates the children to socialize with other kids by taking and giving help in solving the problems. Whenever the players are stuck with some problem then they can press the **Help** button that will send a help request to rest of the players. This help request will be saved in the separate column for each player in the database except the flagged player who has sent the request. When rest of the players opens the game, they will get a notification in the **Notification** icon where they can give the correct answer. So, the question arises that why the children will use this feature without any benefit? The answer to this question is that both the players, helper and the helped, will be rewarded with bonus points. This will encourage the children to involve in healthy social activity and will teach them the importance of team work and helping others.

3.2. Develop System Architecture

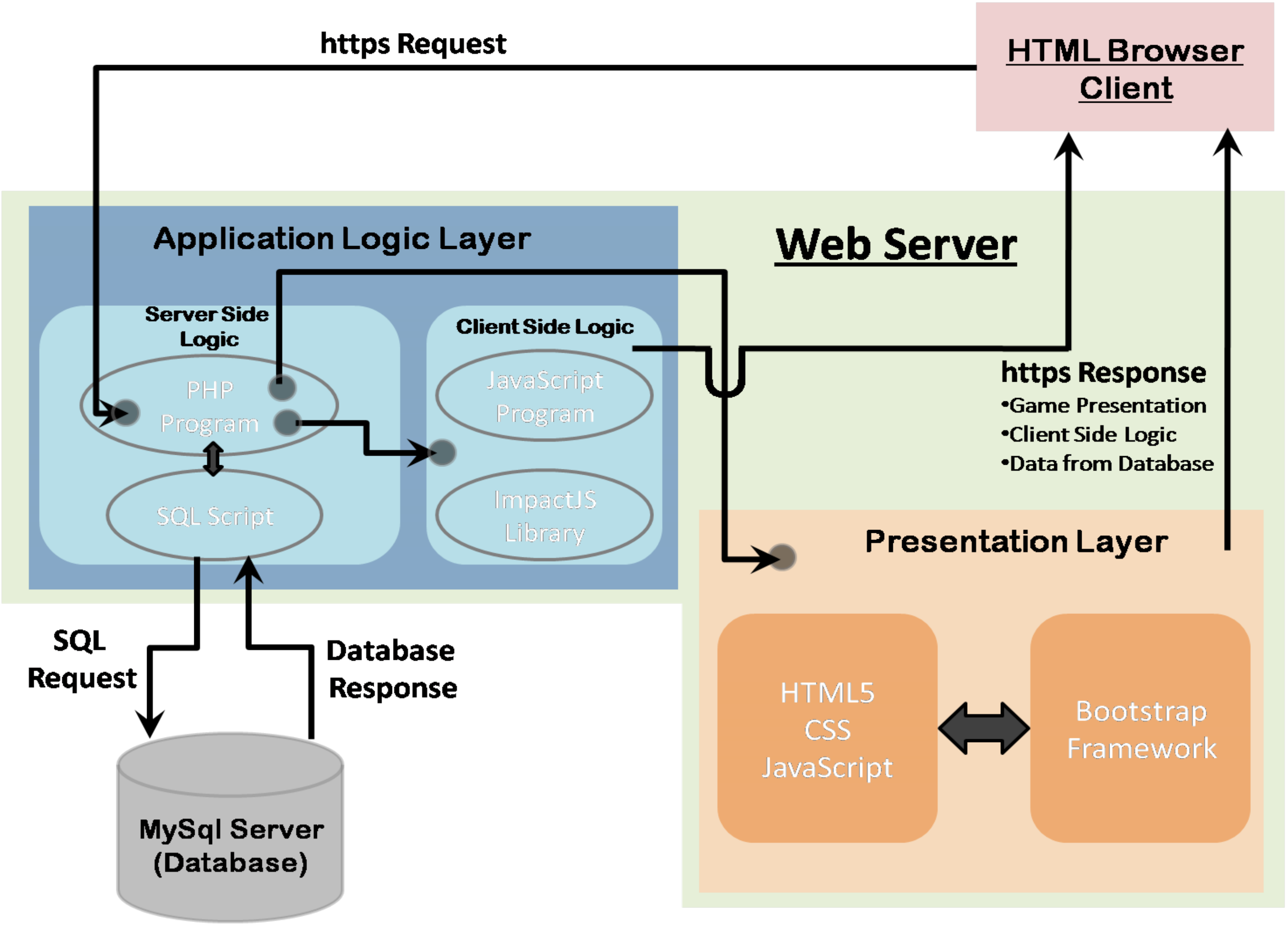


Figure 5: System Architecture for MathMe Game (author's own diagram)

The MathMe game is developed keeping in mind that rather than writing independent code for different platforms and installed on the devices. The game is developed in HTML5, CSS, JavaScript and ImpactJS library. The advantage of using these technologies is that the game can be kept remotely on the server and can be invoked by any browser on diverse devices.

Figure 5 shows the complete architecture of the application.

The above figure shows that Web Model View Controller (Web MVC) framework was used as software architecture for this game development (Bucanek 2009). There are three main components associated with Web MVC.

3.2.1. MathMe the Web Application and implementation of Web MVC

3.2.1.1.View

A view manages the display of the information (Buschmann et al 2007). In this project, one of the questions sought to be answered is:

How can HTML5 based technology be used to create single code base for multiple platforms and investigation into multiple alternatives to render the same information as per the device type?

HTML5, CSS and JavaScript are technologies that are still in a process of evolution. The use of these technologies is increasing day-by-day in order to meet the needs of rendering information on modern hybrid computers and mobile based environment.

There are three main classifications of display types of computing devices:

- Desktop and laptop devices- with relative screen size between 12 inches to 27 inches (diagonal) (Apple 2015).
- Hand-held Tablets- These are a hybrid device, having some characteristics of both desktop and mobile. These devices are generally touch screen devices and are designed in such a way that the battery and all circuitry are fitted in a single unit. This gives users the ability to use it as hand-held device with the added functionality of computers (Mathews 2004). As per Statista (2015), the top five tablet computer vendors include Apple, Samsung, Asus, Lenovo and LG Electronics. These five vendors cover 58.2 percent of the market share as of first quarter of year 2015 (Statista 2015). According to their official websites, these vendors provide tablet computers with relative sizes between 9.7 and 7.9 inch (Apple 2015), 8.0 and 12.2 inches (Samsung 2015), 7.0 and 11.6 inches (Asus 2015), 8 to 13.3 inches (Lenovo 2015) and 7.0 and 10.1 inches (LG 2015). So this classification covers almost all screen sizes between 7.0 and 13.3 inches.
- Mobile phones- Most of the modern smart phone screen sizes vary between 3.75 and 7 inches (Chris 2015).

  As, we can see from the above discussion that there are wide variety of devices available in the market, moreover, these devices support various operating systems and browsers. So, the challenge for software developers is to find an efficient way to develop content for all these devices.

Development approaches for these devices:

There are three choices available for development of software on these devices.

- Develop separate code and applications for different devices with different screen sizes, different operating systems (iOS, Android, Windows, so on), and browser compatibility. Depending on the application's requirements this may be the best option, but there are a number of disadvantages in using this approach:
    - Developers and designers need to develop separate applications for different target devices causing extra development cost and time.
    - Developers need to learn multiple skills and programming languages to develop applications for different devices.
    - The developed apps are generally installed within the devices, causing unnecessary memory and resource consumption.
- Depending on the requirements of the application a more efficient approach may be to develop a web application which can be stored on remote server and can be accessed by any device, desktop, laptop, tablet or mobile via the device's web browser. The advantages associated with this approach are:
    - Single code multiple devices.
    - Less effort
    - Minimal consumption of time and money
    - Potentially faster development
    - Potentially quicker implementation
    - Minimal usage of device resources.
- Hybrid application- That has both native and significant web application sections that combine to form a single native application.

The second approach appears more efficient, however there are challenges associated with using a web based approach:

- There must be a set of technologies which are supported by all the platforms.
- These technologies must have advanced features that can function on all devices.
- These technologies must have the ability to reliably render the content of the application across different screen sizes.

- Support for memory management.
- Ability to use various software architecture patterns which are supported by all the platforms.
- Have a strong history in the software industry and ability to evolve and adapt itself quickly as the technologies changes.

One solution for many of these problems is HTML5 and associated technologies (CSS and JavaScript):

- As already covered in the literature review, HTML5 is a markup language which will run on any device if it is equipped with a modern browser. HTML5, CSS, JavaScript and the Bootstrap framework (HTML5 based rendering framework) provide features such as support for web SWL database, video and audio tags, support for animation, and support for Geolocation API, and support for scalable vector graphics and so on.
- Newer HTML5 tags provide support for responsive design which helps a web page to automatically adjust it display according to a device's screen size.
- Because the entire code base of HTML is kept in remote server, HTML5 based web applications provide high-end memory management ability. A feature such as local storage, where small amounts of data are held in the browser's storages also very efficient.
- HTML5 bases applications can be developed using various software architectures which can be implemented across devices or operating systems such as web MVC.
- HTML has been used from the very start of internet history and it has evolved significantly. An exciting aspect of using HTML5 is that various profit and not-for-profit organizations and consortiums are working together to keep the language up to date.

Implementation and user scenario

We have discussed why HTML5 and its associated technologies were used in this project. Now, we will see how these technologies are used here. We can understand this by considering an example. For instance, there are three children. The first one is using desktop computer, the second is using a mobile phone and the third is using hand-held tablet computer. All of them will open their HTML5 based browsers and request the Web Server for the game. When the game will be sent to three different devices, the server will send **same** HTML5, CSS, JavaScript code, game and Logic code to all of the devices. As soon as, the information from the server reaches to different browsers then JavaScript and CSS will try to recognize the dimensions of the screens.

After recognizing the size of the screen, the HTML5 will render the design as per decided by designer for that particular screen size. In this project, below is one of the examples from the source code where CSS is determining the size of the screen and then rendering the design according to that.

```
@media(max-width:1200px) {

#canvas

          {

width:95%;

height:95%;

background-color:white;

        }
```

Table 5: CSS code for achieving responsiveness

The above code is written in CSS that specified that the width and high of the canvas used in the game must be 95 percent when screen width is less 1200 pixels.

3.2.1.2.Controller

The controller takes the input from the users. This information can be raw data informing the **model** or request to modify the view (Zhang 2012). This project is using PHP as the server scripting language. The use of PHP here is whenever the MathMe players enters the playmathme.com links in the address bar of the web browser then the request to render the page is sent to the PHP program. The PHP program on the server checks the validity of the request and sends the response. If the request meets the set criteria then the server side program sends the requested items such as HTML5, CSS, JavaScript design, client side game logic and data entries from the database. Other operations associated with server side PHP program is authentication and authorization check of the users. It also processes the data requests from the users. Whenever user asks for any data entries then the request goes to PHP program in serve. Then the program processes the request and forwards it to the **Model**. Model, then fetch the data from the database and forward it to the PHP program. The data then sent to the browser/**View** which is then rendered to the user.

For example, in MathMe game when user presses the **"Top 10!"** button present on the canvas. The request to see the top 10 scores goes to PHP program. This request then processed and sent to database and fetched from there. The PHP program then sends the list of top 10 players to the canvas.

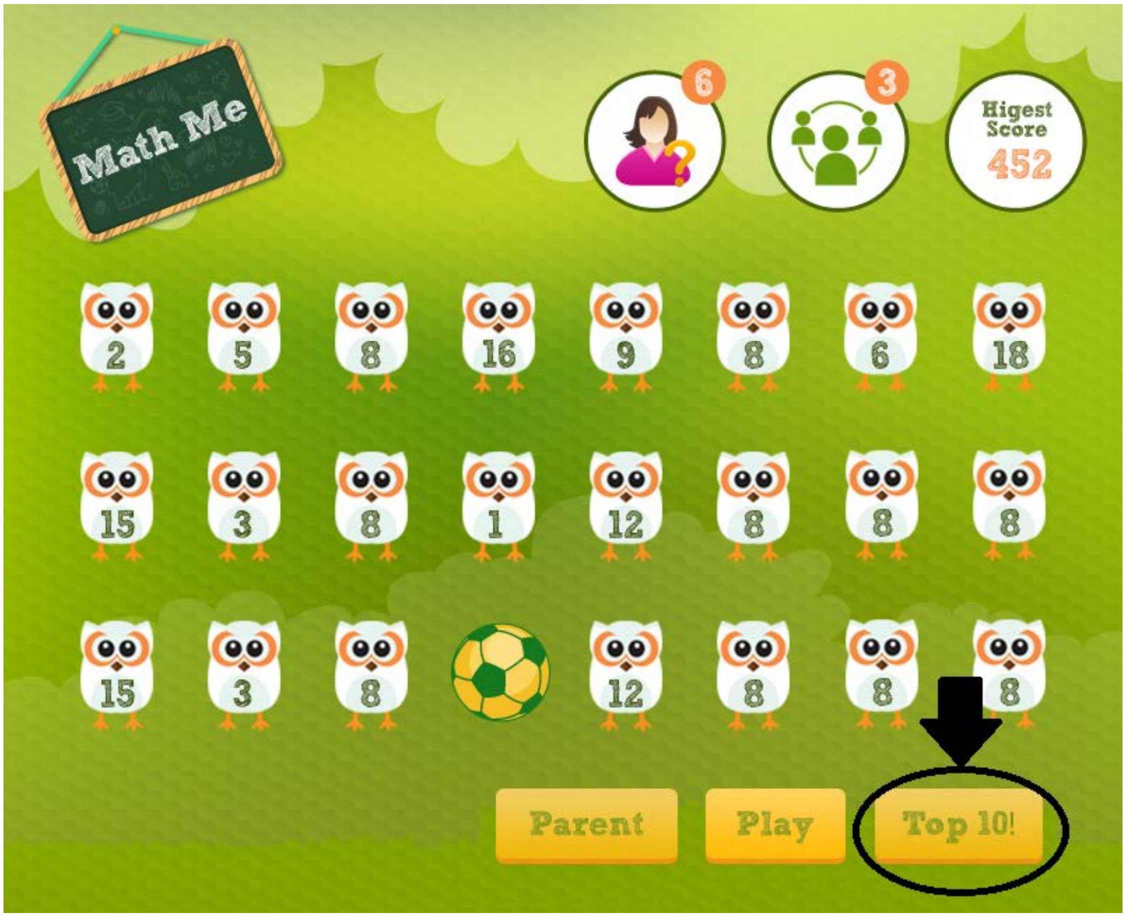


Figure 6: Request for top 10! Scores

When the player will press the “Top 10!” button on the screen, then client side JavaScript code will render the ‘Top 10!’ screen:

```
top10ButtonHandler:function(){

thisRef=this;

this.clearMenu();

this.topten=ig.game.spawnEntity(Topten);

    thisRef.Top10BackButton =ig.game.spawnEntity(
Button,ig.system.width/2-62.5,ig.system.height-130,{ size:{ x:123,
y:54},animSheet:this.Top10homeButtonAnimSheet,
pressed:function(){thisRef.home();

}});


},
```

Table 6: Table 6: Client side JavaScript code for rendering 'Top 10!' screen

When the Top 10 screen will appear then the below code will executed and a request to the server is sent:

```
ig.module('game.entities.topten')
.requires(
'impact.entity'
).defines(function(){
Topten=ig.Entity.extend({
    size:{x:52,y:80},
topten_name:[],
topten_score:[],
font:null,
init:function(x,y,settings){
varthisRef=this;
this.parent(x,y,settings);
this.font=settings.font;
var score = $.get( document.URL +'getscore',function( data ){
       $.each($.parseJSON(data),function(key,value){
thisRef.topten_score.push(value.score);
thisRef.topten_name.push(value.name);
});
});},
```

Table 7: The JavaScript code will send a request to the server

When the request is received by the server then the PHP program will process the request and send back to the browser for rendering:

```
public function getScore()
{        $scores = DB::table('users')
->join('scores','users.id','=','scores.user_id')
->select('users.name','scores.score')
->orderBy('scores.score','desc')
->take(10)
->get();
        echo json_encode($scores);

}
```

Table 8: The PHP Code will process the request

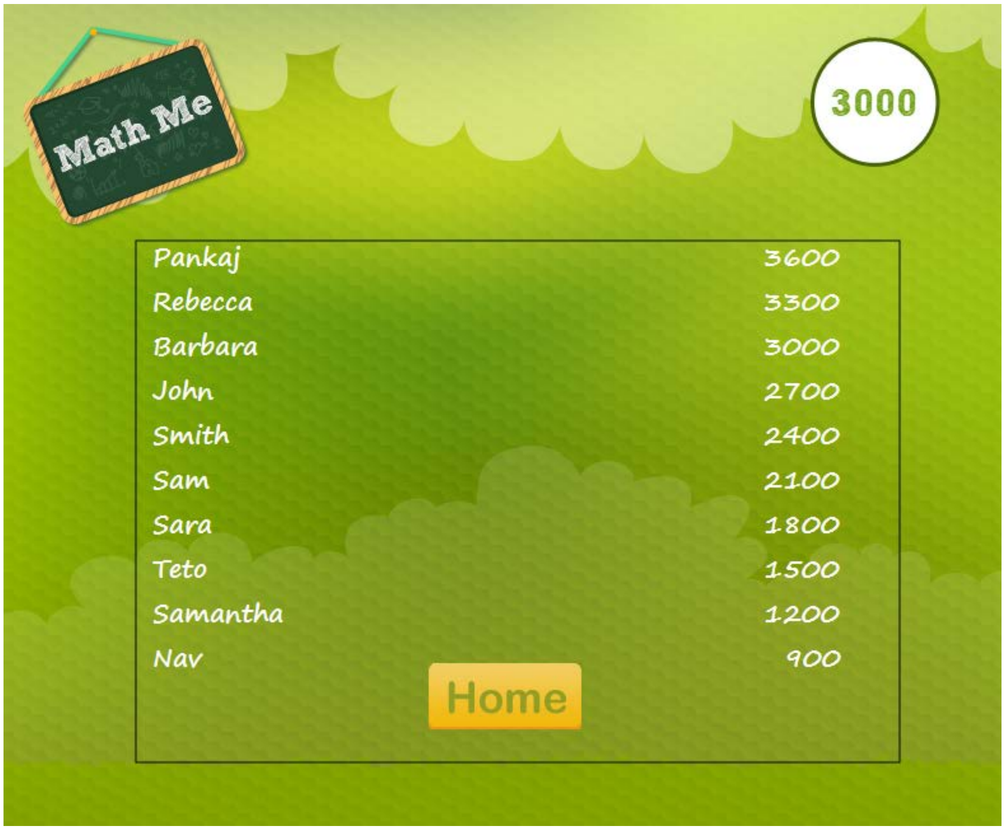
The above function is written in PHP and is present on the server. The request is then processed by this function and collected information is then sent back to the client browser. On the client browser the following JavaScript code will be executed and the top 10 score is rendered on the screen:

```
if(this.topten_score){
varctx=document.getElementById('canvas').getContext('2d');
ctx.font="bold 24px Segoe Print";
ctx.fillStyle="White";
varposY=0;
ctx.strokeRect(110,190,600,410);
for(var i =0; i <this.topten_score.length; i++){
ctx.fillText(this.topten_name[i],200,220+posY);
ctx.fillText(this.topten_score[i],560,220+posY);
posY+=30;

};
```

Table 9: JavaScript code for rendering top 10 players

The above code (in tables 6, 7, 8 and 9) will produce following results:



Pankaj	3600
Rebecca	3300
Barbara	3000
John	2700
Smith	2400
Sam	2100
Sara	1800
Teto	1500
Samantha	1200
Nav	900

Figure 7: 'Top 10!' Screen (Design From one of the inbetween iterations)

Above example is one of the operations of the **Controller**. There are lots of other operations that are involved with the PHP program in the server-side.

3.2.1.3.Model

The Model represents, controls, stores, manipulates and maintains the knowledge (Crawford & Kaplan 2003). In the MathMe project, all the Model related tasks are performed using MySql Server version 10. All the commands for creating, reading, updating and deleting operation are performed using SQL queries.

For, MathMe project, playm694_mathmedatabase was created with five relations:

- users
- password_resets
- scores
- asked_questions
- migrations

SQL JOIN operations are performed to extract data from "asked_questions" and "scores"tables with "users" table.

Data Dictionary for 'playm694_mathme' database:

asked_questions

Column	Type	Null	Default	Comments
id	int(10)	No		
user_id	int(10)	No		
question	varchar(255)	No		
is_answered	tinyint(3)	No	0	

created_at	date	No		
updated_at	date	No		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	7	A	No	
user_id	BTREE	No	No	user_id	7	A	No	

migrations

Column	Type	Null	Default	Comments
migration	varchar(255)	No		
batch	int(11)	No		

password_resets

Column	Type	Null	Default	Comments
email	varchar(255)	No		
token	varchar(255)	No		
created_at	timestamp	No	0000-00-00 00:00:00	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
password_reset_email_index	BTREE	No	No	email	0	A	No	
password_resets_token_index	BTREE	No	No	token	0	A	No	

scores

Column	Type	Null	Default	Comments
id	int(11)	No		

user_id	int(10)	No		
score	bigint(20)	No		
created_at	date	No		
updated_at	date	No		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	23	A	No	
user_id	BTREE	No	No	user_id	4	A	No	

users

Column	Type	Null	Default	Comments
id	int(10)	No		
name	varchar(255)	No		
email	varchar(255)	No		

password	varchar(60)	No		
remember_token	varchar(100)	Yes	Null	
created_at	timestamp	No	0000-00-00 00:00:00	
updated_at	timestamp	No	0000-00-00 00:00:00	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	0	A	No	
users_email_unique	BTREE	Yes	No	email	0	A	No	

ER Diagram for ‘playm694_mathme’ database:

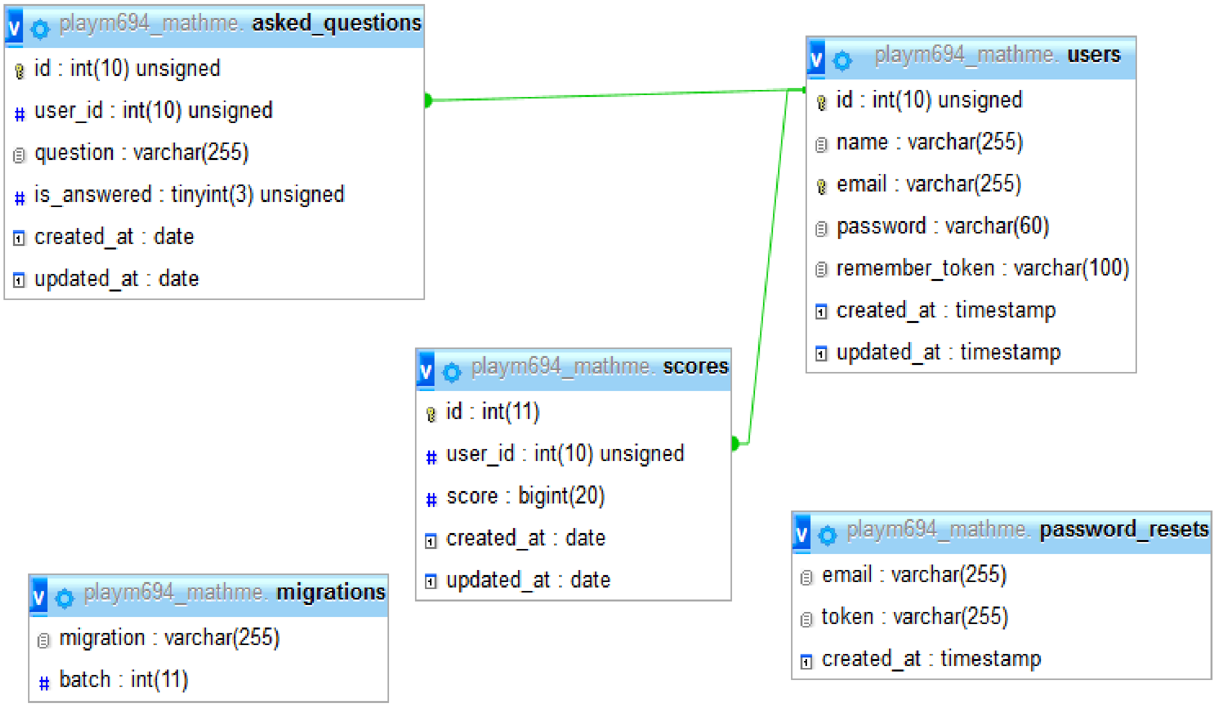


Figure 8: ER Diagram- 'playm694_mathme' Database

3.3. Analysis and Design the system

3.3.1. Analysis

Requirement analysis and software research is critical for the success of a software or game development. Below are the analysis outcomes for this project.

3.3.1.1.Background

The aim of this project is to develop a digital educational game for children aged from 5 to 8 years old. The game's objective is to engage children in play while at the same time helping them to learn the basics of Mathematics. The game can be played on any electronic device that has a browser, e.g. a desktop computer, a laptop or a Smartphone; any mobile device that a family would normally own.

The goal of the game is to provide a positive learning experience for children in the early years of their education. When playing the game, the child would advance through a series of levels and as advancement is made the child could then learn some basic mathematics concepts. The game could be played alone or within a classroom setting. A socialization aspect has been added to the game so that children can anonymously help other children in the class to explore right solutions within the game.

3.3.1.2.Benefits of the study

Children naturally love to play and to learn something educational at the same time could only benefit them. Also associating education with enjoyment at an early age would be advantageous to any child. The social aspect of the game would help children to both give and receive help from other children, which would encourage co-operation and help a child to overcome a potential learning obstacle in a non-threatening manner.

3.3.1.3.What would be expected from the children

All the child will be required to do is to play the game on an iPad, desktop, mobile or any other hand-held smart device for 5 to 10 minutes. It is not child's ability that will be tested here but rather the child will encouraged in interactive manner to learn basic mathematics.

If parents do decide that their child can participate in this project then they will be assured that their child's safety is foremost priority. The parents accompany their child throughout the play time, and the game can be played in a place of parent's choice. If the child decides not, wish to participate at any time or wishes to discontinue participation during the game play that is fine. He or she has the freedom to withdraw from participation at any time.

3.3.1.4.Risks

There are no identified risks associated with participating in this project. However, it is required that the parents be present with their child at all times during they play the game.

3.3.1.5.Confidentiality

The data collected will be confidential, and no identifying information will be sought.

3.3.1.6.Participation

Participation of the child in this project is entirely voluntary, and parents and child are free to decline to participate. If parents want their child to play this game, then parents will still have the opportunity to withdraw from the project at any stage during. Any associated data will not be used in the project.

3.3.1.7.Scope of the project

Project scope is the part of the project analysis where it is determined that what the project will deliver. The project analyze, design, develop and deliver a HTML 5 based educational games for children using advance ImpactJS game engine (JavaScript Library and HTML5 compatible).

A. In Scope

1. A new educational game that will allow the children of age 5 to 8 to learn mathematics.
2. The Game will support all modern devices including desktop computers and Smart devices.
3. The Game will support all modern mobile and desktop browsers.
4. Support for social networking in the child’s pair group and class.
5. Database record for score of each child.
6. Support for personal profile for each player/child.
7. Support for asking help.
8. Support for giving help.

B. Out of Scope of this project

1. Support for Facebook connectivity.
2. Support for mobile calling or SMS.
3. Support for Auto save feature.

4. Training tutorials for the children.
5. Monetary rewards or other incentives.
6. System generated help.
7. Support for email notification to the parents.
8. Offline game play
9. Account verification
10. Support for sounds

3.3.2. Design

It is a process of implementing software engineering solutions to one or more problems to achieve the intended outcome (in this case- to develop the MathMe game). Below are some of the designing approaches used in the project.

3.3.2.1.Working of the game

The working of the game is properly defined in the functional specification document (FRS). As per the project's methodology the FRS underwent multiple changes in number iterations. The information from the FRM is available in the Appendix A.

3.4. Build the system

The game is developed using HTML5, CSS, JavaScript, ImpactJS and PHP. This section will discuss all the necessary development components used during the development.

3.4.1. Directory Structure

The ImpactJS provides the following directory hierarchy:

```
lib/

lib/weltmeister/

lib/game/

lib/impact/

lib/game/entities/

lib/game/levels/

lib/game/entities/
```

During the development, all the necessary game-design resources such as animation, images, logo sounds, etc. were kept in media/directory.

All the game logics were written in JavaScript using ImpactJS libraries and were hosted from lib/directory. The ImpactJS game engine and its library files were hosted form lib/impact/. The ImpactJS also provides one of its unique level editors that was hosted from lib/weltmeister/.

Entities in the game were custom made which were hosted from lib/game/entities. The entities directly supported the source file for their functioning. The source file were hosted from lib/game/.

3.4.2. HTML5 Files

Game written using ImpactJS game engine is a client side code that can be directly run on user's browser. As it is markup based technology, therefore there is no need of any third party plug-in that is needed to be installed on the browser. ImpactJS game engine is JavaScript based, and it do not have the capability to render the interactive component on the browsers. For that purpose, ImpactJS uses HTML5 pages with <canvas> tags to render the information.

```
< div class = " canvas_wrapper " >

< canvas id = " canvas " > < /canvas >

< /div >
```

Table 10: <canvas> tag used in the project

3.4.3. Graphics

Images- ImpactJS provides a wrapper feature that can be wrapped around the images from the media directory. This wrapper is helpful in scaling and loading the images.

Animation- ImpactJS provides animation sheet that can be used to create animation entities. In MathMe game, the bouncing ball, all the buttons, timer, character images, scoreboard and so on are entities.

Animation Sheet- We can load a source image as animation sheet and defined its length and height for the individual frame of the animation.

Font- we can load the font in bitmap image and draw it on the canvas.

3.4.4. Modules

In ImpactJS, modules are the basic unit to organize the source code. Below is the example of the module from the project:

```
ig.module(
    'game.main'
)
.requires(
    'impact.game', 'impact.font', 'plugins.button', 'game.entities.owl',
    'plugins.tween', 'game.entities.Background', 'game.entities.Ball',
    'game.entities.Question', 'game.entities.Score', 'game.entities.Timer',
    'game.entities.topten'
)
.defines(function(){
MyGame = ig.Game.extend({
    currentState:"menu",
//rest of the code for this module
*
*
}
```

Table 11: Example of module definition from the source code

'game.main' is the module name that directly relates to the file name. 'game.main' is hosted from lib/game/main.js. Moreover, there are multiple modules within **.**requires**()** method. These modules ('impact.game'**,** 'impact.font'**,** 'plugins.button' and so on) will be hosted from their files respectively. All the modules presented in the **.**requires**()** method will be preloaded before defines***()*** function is executed.

3.4.5. How ImpactJS Works?

ImpactJS appears as an ordinary library. However, it is a complete framework. It means that we not even used some functionality from the library, but used ImpactJS as a fully functional container. Inside that container, we added our source code. ImpactJS do not require any separate IDE or third party interface. It can run by its own upon HTML5 <canvas> tag on any HTML5 based browser. All the source code for the game is managed by ImpactJS itself.

ImpactJS provides primary base class, and all the user-defined classes in the source code acts as child classes. The most important class that ImpactJS provides is the `ig.Entity` base class. Each and every entity defined in the game inherits `ig.Entity`.

When the browser sends the request to open the game, ImpactJS framework defines an interval that invokes the predefined method `ig.system.run()`at the rate of sixty times per second. The `ig.system.run()`performs some regular housekeeping activities and then invokes another method `.run()`. This method then automatically invokes `.draw()` and `.update()` methods. `.draw()`performs a simple but important task of clearing the screen and is called to render all the entities and background layers. Then the `.update()` method is called each time when an entity is positioned on the background. The `.update()`set up each entity according to the physics property set on them. For example, the bouncing ball entity in MathMe game is positioned as per its defined bounciness property.

After the canvas layout is set up on the screen, the `checkEntities()` method is invoked by ImpactJS. This method manages all the collision operations between different entities. In our case, the collisions between the ball entity and answer owls are maintained by this method. These predefined methods are overwritten in few places to achieve custom functionality and logics.

3.4.6. Working with Assets

Only assets type used in MathMe game are the images. The paths of the all the image files are not relative to the .js files, rather the image files in this project were relative to the **app.blade.php** file, where the actual MathMe game is running.

3.4.7. Collision

There are two types of collision available in the ImpactJS framework:

Static Collision- Between Entities and the Game

Dynamic Collision- Between different Entities

In MathMe game, both dynamic and static collision events occur many a times during the game play. The static collision is executed automatically by the ball entity itself when it moves around in the game environment. Before the ball moves to the next destination in the game environment, it performs a **trace** check from source to the final position. The **CollisionMap** checks the entire track the ball and tells if there is any collision in between.

Dynamic collision operation is performed when the ball is moved. The dynamic collision traces if the **Ball** is currently overlapping with the answer **Owl** and in case of collision between Ball and Owl, the dynamic collision moves the ball apart.

3.4.8. Animation

AnimationSheet is a large packet in ImpactJS that holds individual animation frames. In MathMe project, there were a number of AnimationSheet created by defining an image file from the media directory and then defining the length and width for the specified frame. These AnimationSheets are used as the properties of the class that is loaded beforehand for smooth rendering.

```
this.helpAskedButton = ig.game.spawnEntity( Button, ig.system.width-120,280, {
      size: { x: 102, y: 103 },
      animSheet: thisRef.helpAskedButtonSheet,
      pressed: function() {
        thisRef.helpAskedButtonHandler();
      }
    });
```

Table 12: Example of creating animation for the 'Ask help' button

3.4.9. Moving the ball to particular direction

For moving the ball, simple logic is used. When the tile (Owl) is clicked, it games checks if the answer is correct or wrong. We calculate the angle between the ball and the target tile using following formula:

dy = target.y – ball.y

dx = target.x – ball.x

angle = Math.atan2(dy,dx)

The above formula gives the angle, and this is used to hit the tile. Once we have the angle we can move the ball to that particular position:

ball.pos.x+=Math.cos(ang)*speed

ball.pos.y+=Math.sing(ang)*speed

This moves the ball towards the owl. Nothing will happen when the player will choose an empty area in the game environment because click events are only set for tiles (Owls).

3.5. Observe and evaluate the system

The following guided approached were used for testing of the application during the development of the game.

- Unit testing of MathMe game functions to verify the functioning individual written program module functioning as per specified requirements.
- System and integration testing to verify and validate the functionality of the game against the specified requirements and business rules as per Functional Requirement Specification (FRS). Test to insure that all interfaces between various functions and features and other systems including database and server side system function correctly.
- Testing the game from a point of view of having satisfactory usability, scalability, acceptability, performance, capability and reliability and other non-functional requirements specified in the FRS.
- Additionally as the system's functionality were delivered in multiple iterations. It was imperative that for each change, regression testing was conducted on all impacted modules delivered previous iterations.
- All the testing to ensure that the data was processed accurately as specified by FRS. It was also insured that all the data was recorded in the database for further analysis purpose for other researchers.

4. Conclusion and Recommendations

The first half of the thesis was aimed to find out the limitations of the previously popular technology **Flash**. The objective was to investigate into the future technology i.e. HTML5. Moreover, the part A of the thesis also aimed towards investigating an efficient and suitable game engine for development of HTML5 social game.

It was found from the study that the web browsers need to install the Flash Player as a plug-in to execute the Flash media which is one of the major disadvantages. HTML 5 is a solution to this problem as it is a markup language that is supported by all the web browsers without the need for extra plug-ins. So, the animations and multimedia content developed in the HTML 5 can be rendered on different browsers and diverse devices. Though development of animations and flash websites are easier to develop in a flash by using Adobe tools, however, these tools are proprietary software and expensive to buy. On the other hand, HTML 5 stack (CSS and JavaScript) are recommendations by W3C. These technologies do not require any special integrated development environment (IDE). Hence, the development in HTML5 is becoming popular as dynamic web development is cheaper in HTML5 compared to Flash. Since the beginning of Flash, it gained a bad reputation in Search Engine Optimization (SEO). The primary reason behind the poor performance of Flash websites when it comes to SEO is that usually Flash content is loaded with interactive features and images and less with plain text. On the other hand, leading search engines mostly emphasizes on text to interpret the meaning of the web pages.

HTML5 provides the capability of long-term browser data storage that was missing in the previous versions of HTML. It takes the advantage of Web SQL Database, which brings the capabilities of Relational SQL to the client side. Moreover, all modern browsers are equipped with SQLite database for proper and well-defined data storage. HTML 5 is enabled with “video” and “audio” tags which can be used to play videos and audios respectively without any intervention of other third party languages. HTML 5 carries elements that support location-based services (LSB) and newer formats like Scalable Vector Graphics. It provides Geolocation API to add location-based functionalities to the web pages.

HTML5 is a markup language which will run on any device if it is equipped with a modern browser. HTML5, CSS, JavaScript and the Bootstrap framework (HTML5 based rendering framework) provide features such as support for web SWL database, video and audio tags, support for animation and support for scalable vector graphics and so on. Newer HTML5 tags provide support for responsive design which helps a web page to automatically adjust it display according to a device's screen size. Because the entire code base of HTML is kept in remote server, HTML5 based web applications provide high-end memory management ability. HTML5 bases applications can be developed using various software architectures which can be implemented across devices or operating systems such as web MVC.

HTML has been used from the very start of internet history and it has evolved significantly. An exciting aspect of using HTML5 is that various profit and not-for-profit organizations and consortiums are working together to keep the language up to date.

In the first part of the thesis a brief investigation was done on the various game engines for HTML5 based games. It was found that ImpactJS is one of the most suitable game engines. ImpactJS is not only a library but is a fully functional framework that can be used to develop effective and responsive HTML5 social games. ImpactJS provides 2D rendering of visual contents. It supports engines like sound, physics and collision. ImpactJS also support WebGL for rendering 3D and 2D animation on the compatible browser without any plug-in. ImpactJS is fully compatible with HTML5 and we can use <canvas> tag to include its code into it. Support for HTML5 gives the capability to render the content on all modern smart devices and computers. ImpactJS have the advantage that it preloads all the game assets such as images, sounds, entities and their positioning before stating the game. ImpactJS have other advanced features such as CocoonJS, accelerated HTML5 execution, testing and debugging and a very good documentation for the beginners to develop professional games. There are two main disadvantages of ImpactJS. First, it does not provide a direct implementation for 3D rendering. The other disadvantage of ImpactJS is that the rest of the game engine libraries are freeware, where are, to use ImpactJS, developers, need to spend USD 100/.

The second half of the study focused on the finding out HTML5 solutions to create single code base games that will run on all web-enabled devices. The goal is achieved by taking ImpactJS as the game engine and then developed a social educational game for children. The development process found number of challenges while implementing HTML5 technology. These challenges were overcome by using unique solutions that can be generalized and standardized for other HTML5 bases game development.

There are number of solution that are recommended for other HTML5 games and web development:

Recommendation 1: Content Choreography- The best way to avoid overflow of the one element into another when the screen size varies is to choreograph the contents of the responsive design beforehand. Rather than first developing the design, it is recommended that the designers must initially make themselves aware of the target devices and their screen sizes. Then after that they can decide that how, when and where they want to show the elements as per the screen size.

Recommendation 2: Resolution Independence- Still, in the modern day web development environment, designers creates multiple copies of graphics for different screen sizes. However, the recommendation is to create high pixel-density graphics. One of the formats that was discovered as best suited format for responsive design is SVG (Scalable Vector Graphics).

Recommendation 3: Compressive Images- Modern day smart devices provide very high pixel density in their displays. These displays are capable enough to render standard compressed images with great precision. The recommended to use highly compressed and very large in dimensions images than the size of the screen.

Recommendation 4: Conditional Loading- When we use the CSS media queries independently then the browser loads everything progressively. This can be annoying for the mobile users who want to access some important information at the very beginning of the page load. However that information is still needed to be loaded. In such cases, it is better to implement recommendation 1. However, it is also recommended that the CSS media query must be used with JavaScript conditions to priorities the rendering process according to the needs and significance of the content.

Recommendation 5: Lazy Loading- When the browser requests the server for some web content, it starts receiving all the HTML, CSS, and JavaScript code. JavaScript is time-consuming and if the browser decides to download everything in the single transaction then it consume too much time resulting it bad user experience. The recommendation is to know the correct and necessary demand of the JavaScript code. Using this solution, the browser will download those features/JavaScript, which are needed first. The rest of the scripts can be downloaded simultaneously while the important information is loaded first.

Recommendation 6: Protecting Images/Video aspect ratios- To protect image aspect ratio in responsive design, it is recommended that when the `max-width: 100%;` is set for the width of the image then the height must be set to `height: auto;`. To protect the video aspect ratio in the responsive design, it is recommended that it is effective to create a box with needed ratio and then resize the video inside that box to fit the dimensions of the box.

Recommendation 7: Debugging media Queries- The standard way of testing the media query is to and condition to the media query such that the design component change color when the query is executed. However, it is recommended to add `::before` pseudo selector. With this pseudo selector we can attach the descriptive debugging checks in a clear and clean manner with media queries.

Appendices

Appendix A: Functional Requirement Specification

Game Development| MathMe

February 2015

Version 1.0

Status: Draft

DOCUMENT CONTROL SHEET

For information regarding this document

Name: Pankaj Kashyap

Contact details: pnkjkshp80@gmail.com

Document Version History

Version	Version Date	Revised By	Detail of amendment
0.1	16/03/2015	Pankaj	Initial Draft
0.2	18/03/2015	Pankaj	General modifications
0.3	25/03/2015	Pankaj	New additions to the game rules
1.0	25/03/2015	Pankaj	Proofreading

1.Introduction

1.1. Document Purpose

The purpose of this document is define the functional requirements involved in developing the social, educational game for kids (named MathMe) using HTML5 and supporting technologies. The game will help childrenbetween the ages of 3 and 8 years old to learn basic mathematics.

Note: some of the detailed level requirements pertaining to Data, Integration, and Interfaces will be addressed in the Technical Design stage of this project as mentioned in the later sections of this document.

1.2. Audience

It is anticipated the audience for this document will be:

- CDU panel who will judge the project.
- Software developer involved in the development and integration of the game, to understand the requirements.

1.3. Background

The project will allow the children between the ages of 3 and 8 to interact with the educational content presented in a friendly and interactive manner. The game will provide following will provide the following benefits:

- Present a new method of learning in line with existing educational structure
- Present an opportunityfor the kids to learn with a virtual and paperless environment
    - New applications deployed will continue to support the concept of a consistent addition of interactive educational medium integration to the current educational system.

1.4. Proposed Key Features

At the end of its completion, the game will adopt the following features:

- Design and delivery of game play that will allow the kids to learn basic elementary mathematical concepts in an interactive manner.
- Design and delivery of a prototype game which help the children to be socialized in a positive mannerand interact with each other that may help them overcome any gaming barriers. This will enhance the social skills of the children.

- The ability for children to interact remotely with the educational content outside a classroom.
- Delivery of a technical solution that is affordable, scalable, maintainable and performs consistently across a broad range of platforms.

1.5. Exclusions

The following items are out of scope of this project but may be addressed in future projects:

- Design and delivery of the complete game. The initial prototype of the game will cover the intended functionality as mentioned in this document. Further high-end and enriched functionality can be added to the game as per any additional requirements.

2.Workflows

This section is a high-level overview of the individual workflows identified that the development will need to allow for. Each workflow will encapsulate multiple requirements as detailed in the Requirements Section.

Workflow numbering is a product of existing processes determined in the discovery process for the project requirements for MathMe.

2.1. WF-1 Initial screen showing animated logo.

This will be the welcome screen of the game. It will display the logo of the application and the characters or components used in the games.

2.2. WF-2 Register the kids for the game.

- We cannot take/use personal details of kids to keep their identity anonymous.
- We need to create their virtual profiles.
- The initial registration can be a bit complex process so a parent/guardian of the child should set the profile for their child.
- The sign-in process must be straightforward and simple so that the children between the age group 3-8 can easilylogin without any assistance once a profile is created.

2.3. WF-3 Leaderboard

Leaderboard will show the Score of top 10 players. It will also display the current position of the user. The columns in the lead board will be:

- Rank
- Player Character Name
- Top Score.

2.4. WF-4Welcome Screen

The welcome screen will display the logo/animated images of the gameplay. On the top side, the screen will display three Labels showing. These labels are populated from the database.

At the top of the screen the following will display:

- Number of helps taken
- Number of helps given
- Highest Score of the player

At the bottom side the screen will display three buttons:

- Parent button- will direct the user to the parent disclaimer
- Play button- direct the player to main game screen
- Leaderboard- This Screen will take the user to the leaderboard.

2.5. WF-5 Main Game Screen

This screen will be the maingamingarea. The screen will be divided into two sections.

- The game field/area
- States area.

The Game field/area will be the section where the game will run.

The States area will contain a timer which will display the following:

- Dynamic clock time for the round.
- Notification button-
    - This will show the notification when some other user asks for help.

    - The notifications will be updated after each round and not during the game play.
    - By selecting this button, the player can see all notifications and can select any of the helprequests.
    - When the player selects a request, a small popup window will appear on thescreen, and the game will be paused. The user can help the other player by answering the question in the popup message.
- Help button- This button will allow the play to submit his request to the database to ask for the help.
- Lives- this will show a number- this number will represent the number of lives the player has.
- Player name label- This text label will display the character name of the player.
- Scoreboard- This scoreboard will display the score of the player.
- The main game screen will also contain the exit button. This button will direct the player to the lead board.

2.6. WF-6Disclaimer screen for parents

This screen will contain the text that will explain the feature of the games, its effects on the kids and its benefits, and safety measures that have been taken to protect any child who plays the game.

3.Game Rules

“Must” indicates mandatory functionality or data validated to be required.

“May” shows optional functionality or data, which can be provided but is allowed to be omitted.

3.1. BR-1 Welcome Screen

The introductory welcome screen must contain components used in the game such as

- Tiles
- Ball
- Other shapes.

The purpose of this screen is to give the game a unique and attractive look. The welcome screen must also include a play button. This will direct the player to sign Up/Login screen.

3.2. BR-2 Sign-up and Login rule

- This also will ensure the privacy of the child is kept.
- The registration process consists of:
    - Selecting a cartoon character name from the list of 100 predefined characters.
    - After selecting the character the parent will be prompted to the shapes list. The shapes list will contain images of different shapes. The parent can select one shape for the list as a password.
    - After selected the shape the parent will asked to enter their email ID.
    - The parent can tell the description of cartoon character and the shape to their child.
- The sign-in process must be straightforward and simple so that the children between the age group 3-8 can easilylogin without any assistance once a profile is created.
    - During the sign-in the child can select the cartoon character that was already told to him/her by the parent and similarly select the shape.
    - This feature will provide a sufficient number of combinations and keep the sign-in process simple and straight forward.
- There must be a “Parent” button in the sign-up/Login screen. This button can direct the parents or user of the game to the parent disclaimer screen.

3.3. BR-3 LeadBoard Rule

When any of the columnsis clicked, then that column will expand and will show

- Rank
- Player name
- How many people they have helped
- How many helps they took
- Highest score

The lead board screen will also contain two buttons:

- Welcome screen
- Main Game Screen

3.4. BR-4 Welcome Screen Rule

The label on the top side will be displayed in the order (From Left to right):

1. Help given

2.Help took

3. Highest score.

The button on the bottom side will be displayed in the following order (From left to right)

1. Parent button

2. Play button

3. Lead board button

3.5. BR-5 Main Game Screen

The game will consist of a bouncing ball and a number of tiles kept one next to each other. Each tile will show a random number. An equation will appear on the screen, and the user will calculate the answer to the equation and select the appropriate tile to earn the point. If the answer is correct the points will be added to the user's existing score and the screen will be roll on. The ball will then move to the next selected tile, and a new equation will appear. If the answer is wrong, then the user will be given a choice of the following options:

1. The player will quit
2. The player will use part of his life's score. If the player chose the 2nd option, then the game will be continued and the timer will be his current time plus 100 seconds, and his life's score will be subtracted by one.

3.5.1.BR-5.1. Score Board Rules

For each correct answer, the user will be given 300 points

3.5.2.BR-5.2. Timer Rules

The player will be given 100 seconds in each round. They have to solve as many as questions to earn the points.

3.5.3.BR-5.3. Notification Rules

- The notification icon will show a number (for ex 1,2,3,…) to specify number of pending help notification.
- The player will get 1000 points for giving correct answer for the each notification.

3.5.4.BR-5.4. Help Rules

- If the players find it difficult to answer any question then they can press the help button. This will allow them to submit their help request to other player. When the player will press this button, a pop-up will appear for confirmation of the action. As soon as the player will confirm, other player will instantly get the notification on their screen.
- For asking help the players A will get 500 points. This point will add up to their scoreboard. The player A will be given 30 minutes time to get the answer from the other players B. The player A who has asked the answer can quit the game and come before 30 minutes time and if somebody has given the correct answer then player can continue from the same stage.
- If no one gives the answer in 30 minutes or gives the wrong answer then the instance of the game for player A will be terminated. If he has achieved top 10 score then it will appear in the LeadBoard.

3.5.5.BR-5.5. Life rule

The “life” is a feature which will give players the ability to continue the game after they die. When the life is used the player’s previous timer will be added by 100 seconds. When a life is used, then its total value is subtracted by one. The life button will show a number (for example, 1,2,3….) which will represent that how many lives are left. The life is not like the scoreboard which changes for each game. The number of lives must be stored in the database in a life column for each player. This will allow the players to choose Life in any instance of the game.

3.5.6.BR-5.6. Score Board Rules

For each correct answer, the user will be given 300 points

3.5.7.BR-5.7. Exit button Rules

If the player chooses the exit button, then their score at the time of pressing the button will be compared with their previous highest score in the usertable in the database.If the current score is greater than the highest score then update the highest score with the current score, otherwise do nothing. That means that the player never goes backwards in score and only advances.

3.5.8.BR-5.8. Bonus Tiles Rules

There will be three types of bonus tiles.

1. Blue tiles/Timer Bonus- These tiles will appear like normal tiles, and these will contain the correct answer. However, when the user will select this tile then they will change the colour to blue and the timer will again reset to 100 seconds an d given 300 points.
2. Red Tile/Points bonus- These tiles will appear like normal tiles, and these will contain the correct answer. However, when the user will select this tile then they will change the colour to red and the player will be given 1000 points.
3. Green tile/Life Bonus-- These tiles will appear like normal tiles, and these will contain the correct answer. However, when the user will select this tile then they will change the colour to green and the number of lives will be increased by one.

3.5.9.BR-5.9. Notification Rule

Whenever the players provide help to other players by selecting the notification and giving the right answer to that question, the helping player will get 3000 points as bonus. The bonus 3000 points will be added to their current score.

3.5.10. BR-5.10. End of the game

There are three instances in which the game will end:

1. The player gives the wrong answer- if the player gives the wrong answer then follow rule BR-9.6. [Exit Button Rules]
2. The player presses the exit button-follow rule BR-9.6.[Exit button rules]
3. The time indicates zero time- Follow rule BR-9.6.[Exit button rules]

3.6. BR-6 Disclaimer Screen Rule

The information will be displayed in a horizontal manner in landscape view.

4.Other Requirements

4.1. Interfaces

The interface should be suitable for the user group (ages 3 – 8 years).Because of this, the interface will predominately be comprised of minimal use of text and maximum use of animations and images. The interface should contain large display items such as buttons, labels, tiles, bouncing balls, the notificationicon,etc. It should also be colorful and appealing in nature to the target audience. Bright colors should be used.

4.2. Screens

The game is supportedby all types of screensizes. However, the target audience is the iPad users

4.3. Availability

The game should be available 24x7 online and should provide the latest information about the user states.

4.4. Browser Compatibility

The game will be accessible in all modern browsers.

Appendix B: Game Source Code and Database

Table 13: Source code

Table 14: MySql Database

If the above attachments are not accessible then the source code and database can be downloaded/fork from Git.

The link to the Git is: https://github.com/Pankajkashyap/MathMe-Game

Appendix C: Initial designs and prototypes

Table 15: Initial Design and Prototypes

If the above attachments are not accessible then the source code and database can be downloaded/fork from Git.

The link to the Git is: https://github.com/Pankajkashyap/MathMe-Game

References

93-interactive 2014, *Gamvas Web*, viewed 10-10-2014, <https://www.93i.de/products/software/gamvas>.

Abelson, H, Sussman, GJ & Sussman, J 1996, *Structure and Interpretation of computer programs*, The MIT Press London, England

Adobe 2014, *Easily create apps using the web technologies you know and love: HTML, CSS, and JavaScript*, viewed 11-09-2014, <http://phonegap.com/>.

Andreessen, M 1998, *INNOVATORS OF THE NET: BRENDAN EICH AND JAVASCRIPT*, <https://web.archive.org/web/20080208124612/http://wp.netscape.com/comprod/columns/techvision/innovators_be.html>.

Apple 2015, Compare Models, Apple Inc., viewed 10-05-2015, <http://store.apple.com/au/ipad/compare>.

Apple 2015, Find the Mac that's right for you., Apple Inc., viewed 10-05-2015, <http://store.apple.com/au/mac/compare>.

Asus 2015, Tablets ASUSTeK Computer Inc., viewed 10-05-2015, <http://www.asus.com/Tablets/>.

Bartels, A 2011, *Internet History: HTML Code Evolution 1.0 to 5.0 [INFOGRAPHIC]*, viewed 06-09-2014 2014, <http://www.rackspace.com/blog/internet-history-html-evolution/>.

Beal, V 2014, *tweening*, webopedia, viewed 29-09-2014, <http://www.webopedia.com/TERM/T/tweening.html>.

Beatty, P, Dick, S & Miller, J 2008, 'Is HTML in a Race to the Bottom? A Large-Scale Survey and Analysis of Conformance to W3C Standards', *IEEE Internet Computing*, vol. 12, no. 2, pp. 76-80.

Bilton, N 2010, *Amazon to Introduce Web-Based Book Previews*, viewed 15-09-2014, <http://bits.blogs.nytimes.com/2010/06/30/amazon-to-launch-web-based-book-previews/?_php=true&_type=blogs&_r=0>.

Boeri, R 2011, 'Will HTML5 Edge Out Flash?', EventDV, vol. 24, no. 10, pp. 39-.

Boeri, RJ & Hensel, M 1998, 'XML: The new document standard', *EMedia Professional*, vol. 11, no. 6, p. 33.

Bos, B, Celik, T, Hickson, I & Lie, HW 2011, *Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification*, W3C, viewed 14-09-2014, <http://www.w3.org/TR/2011/PR-CSS2-20110412/>.

Bos, B, Lie, H. W, Lilley, C & Jacobs, I 1998, *Cascading Style Sheets, level 2 CSS2 Specification*, W3C, <http://www.w3.org/TR/2008/REC-CSS2-20080411/>.

Boswell, W 2013, *Developers and the psychology of social gaming*, Intel Developer Zone viewed 14-09-2014, <https://software.intel.com/en-us/blogs/2013/11/08/developers-and-the-psychology-of-social-gaming>.

Boyes, E 2008, *GDC '08: Are casual games the future?*, viewed 14-09-2014, <http://www.gamespot.com/articles/gdc-08-are-casual-games-the-future/1100-6186207/>.

Bucanek, J 2009, Learn Objective-C for Java Developers, Apress.

Buschmann, F, Henney, K & Schmidt, DC 2007, Pattern-Oriented Software Architecture, On Patterns and Pattern Languages, Wiley.

Cailliau, R 1997, *WWW Proposal: "How It really Happened"*, Institute Of Electrical and Electronic Engineers

Cantor, D & Jones, B 2012, *WebGL Beginner's Guide*, Packt Publishing.

Carey, P 2012, *New Perspectives of HTML and CSS*, Nicole Jones Pinard, United States of America

Çelik, T, Etemad, EJ, Glazman, D, Hickson, I, Linss, P & Williams, J 2011, *Selectors Level 3*, viewed 14-09-2014, <http://www.w3.org/TR/css3-selectors/>.

Çelik, T, Lilley, C, Baron, LD, Pemberton, S & Pettit, B 2011, *CSS Color Module Level 3*, W3C, viewed 14-09-2014, <http://www.w3.org/TR/css3-color/>.

Charles, S 2012, 'JavaScript: Designing a Language in 10 Days', vol. 45, no. 2, pp. 7-8, <http://doi.ieeecomputersociety.org/10.1109/MC.2012.57>.

Charlesworth, S 2014, New Tags in HTML 4.0, viewed 11-09-2014, <http://www.htmlgoodies.com/tutorials/html_401/html4-ref/article.php/3460281>.

Chris, P 2015, The super-sized chart: meet the largest smartphones on the planet, phoneArena.com, viewed 10-05-2015, <http://www.phonearena.com/news/The-super-sized-chart-meet-the-largest-smartphones-on-the-planet_id51327>.

COCOS2DX 2014a, *Cocos2d-x*, viewed 10-10-2014, <http://www.cocos2d-x.org/wiki/Cocos2d-x>.

—— 2014b, *Getting Started with Cocos2d-JS*, viewed 10-10-2014, <http://www.cocos2d-x.org/wiki/Getting_Started_Cocos2d-js>.

ComputerHope 2014, *Physics engine*, viewed 09-10-2014, <http://www.computerhope.com/jargon/p/physics-engine.htm>.

Cowan, D 2014, *Joystiq*, viewed 29-09-2014, <http://www.joystiq.com/editor/danny-cowan/page/92/#posts>.

Craftyjs 2013, *Crafty- Code Like A Fox*, viewed 04-10-2014, <http://craftyjs.com/>.

Crawford, W & Kaplan, J 2003, J2EE Design Patterns, O'Reilly Media.

CreateJSSuits 2014, *CreateJS*, viewed 29-09-2014, <http://createjs.com/#!/CreateJS>.

CrowedPark 2012, 'The State of the Socail Gaming Insustry: Gambling to Socail Gaming, Social Gaming to Gambling '.

Cutler, K-M 2012, *There is a $100M discrepancy in Facebook's payments revenues*, Inside Social Gaming, viewed 17-09-2014, <http://www.insidesocialgames.com/2012/02/10/there-is-a-100m-discrepancy-in-facebooks-payments-revenues/>.

Daoust, F, Hoschka, P, Patrikakis, CZ, Cruz, RS, Nunes, MS & Osborne, DS 2013, 'Towards Video on the Web with HTML5 '.

EaselJS 2014, *EaselJS Module*, viewed 29-09-2014, <http://createjs.com/Docs/EaselJS/modules/EaselJS.html>.

ECMAInternational 2011, *ECMAScript Language Specification*, ECMA International Rue du Rhone.

Ericson, C 2005, *Real-Time Collision Detection*, Taylor & Francis.

ESA 2014, *Entertainment Software Association: Top 10 Industry Facts.*, viewed 17-09-2014, <http://delivery.acm.org/10.1145/1330000/1328221/p105-kuittinen.pdf?ip=138.80.102.54&id=1328221&acc=ACTIVE%20SERVICE&key=65D80644F295BC0D.1DEB099AEB0D7EC3.4D4702B0C3E38B35.4D4702B0C3E38B35&CFID=566429413&CFTOKEN=85646329&__acm__=1410927454_a15431d8727cbe5820eca47546a605af>.

Etemad, EJ 2011, *CSS Namespaces Module Level 3*, W3C, <http://www.w3.org/TR/css3-namespace/>.

Fisch, SM 2005, 'Making educational computer games "educational"', paper presented to Proceedings of the 2005 conference on Interaction design and children, Boulder, Colorado, DOI 10.1145/1109540.1109548.

Firdaus, T 2013, *Responsive Web Design by Example*, Packt Publishing, Limited.

Fishkin, R 2008, *Flash and SEO - Compelling Reasons Why Search Engines & Flash Still Don't Mix*, viewed 18-09-2014, <http://moz.com/blog/flash-and-seo-compelling-reasons-why-search-engines-flash-still-dont-mix>.

Flanagan, D 2011, *JavaScript: The Definitive Guide* O'Reilly Media.

GameClosure 2014, *Features* viewed 09-10-2014, <http://www.gameclosure.com/>.

Gatto, JG, *Esplin, DB & O'Reardon, ME 2014, Blurry Lines: Legal and Business Uncertainty in Social Media - Virtual Currency, viewed 14-09-2014, <http://www.socialgameslaw.com/virtual-goods/>.*

Google 2013, Google Swiffy Beta, viewed 15-09-2014, <https://www.google.com/doubleclick/studio/swiffy/>.

GWT 2014, What is HTML5 Storage?, viewed 10-10-2014, <http://www.gwtproject.org/doc/latest/DevGuideHtml5Storage.html>.

Hickson, I 2010, Web SQL Database, viewed 07-09-2014, <http://www.w3.org/TR/webdatabase/>.

—— 2011, 'HTML5 A Vocabulary and Associalted APIs for HTML and XHTML', *W3C*.

Hickson, L 2013, *Web Storage*, viewed 07-09-2014, <http://www.w3.org/TR/webstorage/>.

Hochman, J 2009, *How to SEO Flash*, viewed 18-09-2014, <http://pasaportes.gov.do/wp-content/uploads/2012/04/seo-flash.pdf>.

Hockenson, L 2013, *Zynga may be coming back, but social gaming is not what it was*, viewed 14-09-2014, <https://gigaom.com/2013/10/25/zynga-may-be-coming-back-but-social-gaming-is-not-what-it-was/>.

html5gameengine 2014, *HTML5 Gaming Engines* viewed 14-09-2014, <http://html5gameengine.com/>.

Humphrey, A & Humphrey, C 2010, *Virtual Villagers 4 - The Tree of Life*, viewed 14-09-2014, <http://www.ldw.com/virtual_villagers_4_the_tree_of_life.php>.

ImpactJS 2014a, *The Awesomest Way to Create Even More Awesome HTML5 Games!* , viewed 29-09-2014, <http://impactjs.com/>.

—— 2014b, *Documentation >> Debug*, viewed 29-09-2014, <http://impactjs.com/documentation/debug>.

—— 2014c, *Documentation >> Weltmeister*, viewed 29-09-2014, <http://impactjs.com/documentation/weltmeister>.

—— 2011, *Information technology — Programminglanguages, their environments andsystem software interfaces— ECMAScript language specification* <http://webstore.iec.ch/preview/info_isoiec16262%7Bed3.0%7Den.pdf>.

Jakus, G & Sodnik, J 2010, 'New technologies for web development'.

Johnson-Eilola, J & Hea, ACK 2003, 'After hypertext: Other ideas ', *Computers and Composition*, vol. 20, no. 4, pp. 415-25.

Kaiser, R 2013, *When sports and role-playing games collide*, Joystiq, viewed 14-09-2014, <http://www.joystiq.com/2013/02/01/when-sports-and-role-playing-games-collide/>.

Kalpias, V 2014, *playcanvas/engin*, Github, viewed 29-09-2014, <https://github.com/playcanvas/engine>.

Kavouras, L 2013, *Roulette House Edge: No Big Deal* Roulette 30, viewed 14-09-2014, <http://www.roulette30.com/2013/10/roulette-house-edge-strategy.html>.

KiwiJS 2013, *Documentation- What is Kiwi.js?*, viewed 09-10-2014, <http://www.kiwijs.org/documentation/>.

Kleinrock, L 1972, *Communication nets; stochastic message flow and delay*, Dover Publications, Incorporated.

Kiili, K 2005, 'Digital game-based learning: Towards an experiential gaming model', The Internet and Higher Education, vol. 8, no. 1, pp. 13-24.

Kramis, M & Giannakaras, G 2008, 'TEMPORAL REST - HOW TO REALLY EXPLOIT XML', *Proceedings of the IADIS International Conference on WWW/Internet*, pp. 93-102.

LazerUnicorns 2014a, *HTML5 Examples* viewed 10-10-2014, <http://lycheejs.org/examples.html>.

—— 2014b, *LycheeJS*, viewed 10-10-2014, <http://lycheejs.org/features.html>.

LazyFoo' 2014, *Bitmap Fonts*, viewed 10-10-2014, <http://lazyfoo.net/SDL_tutorials/lesson30/>.

Leblon, R 2010, 'Building Advanced offline web application with HTML 5'.

Lee, J 2010, *Games and Gaming* makeuseof, viewed 16-10-2014, <http://www.makeuseof.com/tag/five-free-game-development-tools-make-your-own-games/>.

Leiner, BM, Cerf, VG, Clark, DD, Kahn, RE, Kleinrock, L, Lynch, DC, Postel, J, Roberts, LG & Wolff, S 2009, 'A brief history of the internet', *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 5, pp. 22-31.

Li, R 2009, *The State-of-the-arts in Focused Search*, 09-28, University of Twente, Centre for Telematics and Information Technology, Enschede, the Netherlands, ["eprint_fieldopt_monograph_type_Technical Report" not defined], <http://doc.utwente.nl/67516/>.

Lie, HW 1994, *Cascading HTML style sheets -- a proposal*, viewed 14-09-2014, <http://www.w3.org/People/howcome/p/cascade.html>.

Lie, HW & Bos, B 1996a, *Cascading Style Sheet, level 1*, W3C viewed 14-09-2014, <http://www.w3.org/TR/2008/REC-CSS1-20080411/>.

—— 1996b, *Cascading Style Sheets, level 1*, viewed 14-09-2014, <http://www.w3.org/TR/1999/REC-CSS1-19990111>.

Lie, HW & Bos, B 2005, *Cascading Style Sheets: Designing for the Web*, Pearson Education.

Lenovo 2015, Tablets, Lenovo, viewed 10-05-2015, <http://shopap.lenovo.com/au/en/tablets/?menu-id=tablets>.

LG 2015, Tablets, LG Electronics viewed 10-05-2015,<http://www.lg.com/us/tablets>.

ludei 2013, *CocoonJS*, viewed 08-10-2014, <https://www.ludei.com/cocoonjs/>.

Mac Conaonaigh, S 2007, 'The Best of Both Worlds', *Scientific Computing*, vol. 24, no. 10, pp. 10-6.

Marill, T & Roberts, LG 1966, 'Toward a cooperative network of time-shared computers', paper presented to Proceedings of the November 7-10, 1966, fall joint computer conference, San Francisco, California, DOI 10.1145/1464291.1464336.

Martinet, K 2005, *Canvas tutorial*, viewed 07-09-2014, <https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Canvas_tutorial>.

Mathews, CF 2004, Absolute Beginner's Guide to Tablet PCs, Que.

McArdle, R 2014, 'HTML5 OVERVIEW: A LOOK AT HTML5 ATTACK SCENARIOS'.

melonJS 2014, *melonJS- A lightweight HTML5 game engine*, viewed 10-10-2014, <http://melonjs.org/>.

Meyer, EA 2000, *Cascading Style Sheets: The Definitive Guide*, O'Reilly \&amp; Associates, Inc.

Microsoft 2014, *Get ready for plug-in free browsing*, viewed 30-09-2014, <http://msdn.microsoft.com/en-au/library/ie/hh968248%28v=vs.85%29.aspx>.

Navarro, PF 2014, *Intro to HTML5 Game Development*, Zenva Academy, viewed 12-08-2014 2014, <https://academy.zenva.com/product/intro-to-html5-game-development/>.

Neslon, G 2013, *Looking for Role playing games with interesting social mechanics*, Role Play Games viewed 14-09-2014, <http://rpg.stackexchange.com/questions/21487/looking-for-role-playing-games-with-interesting-social-mechanics>.

Netscape 1995, *NETSCAPE AND SUN ANNOUNCE JAVASCRIPT, THE OPEN, CROSS-PLATFORM OBJECT SCRIPTING LANGUAGE FOR ENTERPRISE NETWORKS AND THE INTERNET*, Netscape Communications Corporation, California, <https://web.archive.org/web/20070916144913/http://wp.netscape.com/newsref/pr/newsrelease67.html>.

Nichol, M 2012, 15 Great Word Games DailyWritingTips, viewed 14-09-2014, <http://www.dailywritingtips.com/15-great-word-games/>.

Nunamaker, J. F. Jr., Chen, M., & Purdin, R. D. M. 1991, 'Systems Development in Information Systems Research', Journal of Management Information Systems, vol. 7, no. 3, pp. 89-106.

Nystrom, B 2014, Component, Game Programming Pattern, viewed 09-10-2014, <http://gameprogrammingpatterns.com/component.html>.

Offerman, S & Hunter, A 2009, *Palm Latest Mobile Industry Leader to Join Open Screen Project*, viewed 15-09-2014, <https://www.adobe.com/aboutadobe/pressroom/pressreleases/200902/021609AdobePalmOSP.html>.

OpenGL 2014, *OpenGL API Documentation Overview*, viewed 10-10-2014, <https://www.opengl.org/documentation/>.

Panigrahy, N 2012, 'Developing Offline Web Applications using HTML5', viewed 10-09-2014, <http://www.tcs.com/SiteCollectionDocuments/White%20Papers/TEG_Whitepaper_Developing_Offline_Web_Application_Using_HTML5_0212-1.pdf>.

Pilgrim, M 2011, *The Past, Present & Future of Local Storage for Web Applications*, viewed 30-09-2014, <http://diveintohtml5.info/storage.html>.

Pixi.js 2013a, *2D webGL renderer with canvas fallback* viewed 29-09-2014, <http://www.pixijs.com/>.

—— 2013b, *Fastest kid in town* viewed 29-09-2014, <http://www.pixijs.com/fastest-kid-in-town/>.

—— 2013c, *Tinting and Blend Modes* viewed 29-09-2014, <http://www.pixijs.com/Tinting-and-Blend-Modes/

>.

—— 2013d, *True multi-paltform* viewed 29-09-2014, <http://www.pixijs.com/True-multi-platform/>.

PlayCanvas 2014a, *Developer Resources* viewed 29-09-2014, <http://developer.playcanvas.com/user-manual/introduction/>.

—— 2014b, *Developer Resources* viewed 29-09-2014, <http://developer.playcanvas.com/user-manual/designer/assets/>.

Popescu, A 2014, *Geolocation API Specification*, viewed 07-09-2014, <http://dev.w3.org/geo/api/spec-source.html>.

PreloadJS 2014, *PreloadJS Module*, viewed 29-09-2014, <http://createjs.com/Docs/PreloadJS/modules/PreloadJS.html>.

Prensky, M 2005, 'Computer games and learning: Digital game-based learning', Handbook of computer game studies, vol. 18, pp. 97-122.

Raggett, D 1995, *Introduction to HTML 3.0*, viewed 30-08-2014, <http://www.w3.org/MarkUp/html3/intro.html

>.

Raggett, D, Lam, J, Alexander, I & Kmiec, M 1998, *Raggett on HTML 4*, Addison-Wesley.

Rettig, P 2014, *Free Introduction to HTML5 Game Development Course*, HTML 5 GAME DEVELOPMENT, viewed 14-09-2014, <http://www.html5gamedevelopment.com/html5-game-tutorials/2014-07-free-introduction-to-html5-game-development-course>.

Riehle, D 2000, 'Framework Design: A Role Modeling Approach ', Ph. D thesis, ETH Zürich.

Rivoal, F 2012, *Media Queries*, W3C, viewed 14-09-2014, <http://www.w3.org/TR/css3-mediaqueries/>.

Samsung 2015, Tablets, SAMSUNG, viewed 10-05-2015, <http://www.samsung.com/au/consumer/mobile-phone/tablet/>.

Sanders, A-L 2014, about technology, viewed 11-10-2014, <http://animation.about.com/od/faqs/f/In-Computer-Animation-What-Is-Rendering.htm>.

Scirra 2014a, *Creat games. Effortlessly.*, viewed 29-09-2014, <https://www.scirra.com/>.

—— 2014b, *Interface*, viewed 29-09-2014, <https://www.scirra.com/manual/36/interface>.

Scott, H 2011, *HTML 5: A closer look at the technology that will replace Flash*, viewed 15-09-2014, <http://www.phonearena.com/news/HTML-5-A-closer-look-at-the-technology-that-will-replace-Flash_id23722>.

Sellers, G, Wright, RS & Haemel, N 2013, *OpenGL SuperBible: Comprehensive Tutorial and Reference*, Pearson Education.

sglmsource 1990, *A Brief History of the Development of SGML*, viewed 02-09-2014, <http://www.sgmlsource.com/history/sgmlhist.htm>.

Shankland, S 2009, *Need for speed spurs Opera JavaScript overhaul*, viewed 15-09-2014, <http://www.cnet.com/news/need-for-speed-spurs-opera-javascript-overhaul/>.

Shannon, R 2012, *The History of HTML*, viewed 28-08-2014, <www.yourhtmlsource.com/starthere/historyofhtml.html>.

Shea, D & Holzschlag, ME 2005, *The Zen of CSS Design: Visual Enlightenment for the Web (Voices That Matter)*, Peachpit Press.

Sivak, S 2009, *What is Game Engin?*, slideshare, viewed 09-10-2014, <http://www.slideshare.net/sjsivak/what-is-a-game-engine>.

Smiley, B 2012, *What is a game engine?*, Deluge.co, viewed 17-10-2014, <http://www.deluge.co/?q=what-is-a-game-engine>.

SooHoo, K 2012, 'The role of HTML5 and Flash in Web Design'.

SoundJS 2014, *SoundJS Module*, viewed 29-09-2014, <http://createjs.com/Docs/SoundJS/modules/SoundJS.html>.

Standard_ECMA-262 1999, 'ECMAScript Language Specification', *ECMA Standardizing Information and Communication Systems*.

Statista 2015, Global market share held by tablet vendors from 2nd quarter 2011 to 1st quarter 2015 http://www.statista.com, viewed 10-05-2015, <http://www.statista.com/statistics/276635/market-share-held-by-tablet-vendors/>.

Sutopo, H 2010, 'FLASH GAME WITH MYSQL DATABASE DEVELOPMENT', *Journal of Theoretical & Applied Information Technology*, vol. 20, no. 1, pp. 45-51.

Szablewski, D 2011, 'Impact Game Engin ', *Game Developer Magazine* viewed 29-09-2014, <http://impactjs.com/files/GDM_May_2011.pdf#page=33>.

Taft, DK 2011, 'Adobe Commits to Both Flash and HTML5', *eWeek*, vol. 28, no. 17, pp. 16-8.

TheEconomist 2011, *The Play's the thing* viewed 14-09-2014, <http://www.economist.com/node/21541159>.

Toshniwal, R & Agrawal, DP 2004, 'Tracking the Roots of Markup Languages ', *Commuincations of the ACM*, vol. 47, no. 5, pp. 95-8.

Turbulenz 2011, *turbulenz THE TURBULENZ ENGINE*, viewed 29-09-2014, <http://biz.turbulenz.com/static/presentations/Turbulenz-GDC-NVIDIA-Technology-2011.pdf>.

—— 2012a, *Create, test and monetise your game with Turbulenz free of charge*, viewed 29-09-2014, <http://biz.turbulenz.com/developers>.

—— 2012b, *PLATFORM ARCHITECTURE OVERVIEW*, viewed 29-09-2014, <http://biz.turbulenz.com/static/presentations/Turbulenz-Platform-Architecture-2014-04.pdf>.

TurbulenzLabs 2011, *Turbulenz Games Platform: Technology Introduction*, viewed 29-09-2014, <http://biz.turbulenz.com/static/presentations/Turbulenz-Whitepaper-2011.pdf>.

TweenJS 2014, viewed 29-09-2014, <http://createjs.com/Docs/TweenJS/modules/TweenJS.html>.

Vanghan-Nichols, SJ 2013, *W3C defines local Web storage API*, ZDNet30-09-2014, <http://www.zdnet.com/w3c-defines-local-web-storage-api-7000019055/>.

Vaughan-Nichols, SJ 2011, *Flash is dead. Long live HTML5.*, viewed 15-09-2014, <http://www.zdnet.com/blog/networking/flash-is-dead-long-live-html5/1633>.

VisionMobile 2011, 'HYML 5 and what it means for the mobile industry '.

W3C 1994, *Partial CSS software list*, viewed 14-09-2014, <http://www.w3.org/Style/CSS/software.en.html#w26>.

—— 2000, *XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition)*, viewed 06-09-2014, <http://www.w3.org/TR/xhtml1/>.

—— 2005, *Document Object Model (DOM)*, viewed 15-09-2014, <http://www.w3.org/DOM/#why>.

—— 2008, *Extensible Markup Language (XML) 1.0 (Fifth Edition)*, viewed 06-09-2014, <http://www.w3.org/TR/2008/REC-xml-20081126/>.

—— 2013a, *HTML & CSS*, viewed 14-09-2014, <http://www.w3.org/standards/webdesign/htmlcss#whatcss>.

—— 2013b, *ISGMLUG*, viewed 02-09-2014, <http://www.isgmlug.org/>.

—— 2014, *Web Storage*, viewed 30-09-2014, <http://dev.w3.org/html5/webstorage/>.

W3CRecommendation 1999a, *Introduction to HTML 4*, viewed 27-08-2014, <http://www.w3.org/TR/REC-html40/intro/intro.html#h-2.3>.

—— 1999b, *On SGML and HTML*, W3C, viewed 02-09-2014, <http://www.w3.org/TR/REC-html40/intro/sgmltut.html>.

Wallace, M & Robbins, B 2006, '2006 Casual Games White Paper'.

Wang, R 2011, *Research Summary: Demystifying Enterprise Gamification For Business*, Forbes, viewed 14-09-2014, <http://www.forbes.com/sites/ciocentral/2011/12/07/research-summary-demystifying-enterprise-gamification-for-business/>.

Ward, J 2008, *What is a Game Engine?*, viewed 28-09-2014, <http://www.gamecareerguide.com/features/529/what_is_a_game_.php>.

Wayner, P 2010, *HTML5 vs Flash: The case for Flash* viewed 15-09-2014, <http://www.infoworld.com/d/application-development/html5-vs-flash-the-case-flash-721>.

Winokur, D 2011, *Flash to Focus on PC Browsing and Mobile Apps; Adobe to More Aggressively Contribute to HTML5*, viewed 15-09-2014, <http://blogs.adobe.com/conversations/2011/11/flash-focus.html>.

Yergeau, F, Nicol, G, Adams, G & Duerst, M 1997, *Internationalization of the Hypertext Markup Language*, Standards Track.

YOYOGAMES 2014a, *Cross Platform from One Codebase* viewed 29-09-2014, <http://www.yoyogames.com/studio/multiformat>.

Yusoff, A, Crowder, R, Gilbert, L & Wills, G 2009, 'A Conceptual Framework for Serious Games', in Advanced Learning Technologies, 2009. ICALT 2009. Ninth IEEE International Conference on, pp. 21-3.

Zhang, W 2012, Software Engineering and Knowledge Engineering: Theory and Practice: Selected papers from 2012 International Conference on Software Engineering, Knowledge Engineering and Information Engineering (SEKEIE 2012), Springer.

—— 2014b, *Developer Services*, viewed 29-09-2014, <http://www.yoyogames.com/studio/developer-services>.

—— 2014c, *Engage*, viewed 29-09-2014, <http://www.yoyogames.com/studio/developer-services/engage>.

—— 2014d, *Focus on What Matters* viewed 29-09-2014, <http://www.yoyogames.com/studio/workflow>.

—— 2014e, *GameMaker: Studio* viewed 29-09-2014, <http://www.yoyogames.com/studio>.

—— 2014f, *Performance*, viewed 29-09-2014, <http://www.yoyogames.com/studio/performance>.