

An Enhanced Approach to Responsive Web Design in Fluid Grid Concept

Abdulrehman Mohamed¹, Dr. Cheruiyot W.K. PhD², & Dr. Richard Rimiru, PhD³.

*Jomo Kenyatta University of Agriculture and Technology, Institute of Computer Science and Information Technology,
P. O. Box 62000, Nairobi, Kenya.*

¹almutwafy@gmail.com, ²wilchery68@gmail.com, and ³rimirurm@yahoo.com

Abstract -Web design is a dynamic field which evolved around user viewing experience trends. The recent trends have revolutionized the web design approach technologies. The result of which, Responsive Web Design (RWD) concept was coined by Web Designer Ethan Marcotte in the year 2010. RWD is an approach of new paradigms and techniques to develop one single website which looks different for different screen sizes so that it is usable on any access web device. The core concepts of RWD are; media query concept, fluid grid concept, and fluid image concept. The study is inspired by the fluid grid concept. Lots of ink has been spilled on the concept, especially in areas of conversion of website's fixed grid layouts to fluid grid layouts. However, these approaches involved heavy customization, resulting into slow adaption by web designers. Therefore, the study's objective is to address this gap by implementing an alternative approach that will be non-customizable and accessible by a single line of code. Hence, study presented an enhanced approach to RWD by implementing an algorithm coined Liquidizer.js in a jQuery Framework. The study employed Matt Kersley RWD Tool: for testing Liquidizer.js, Bersoft Image Measurement (BIM) tool: for validating Liquidizer.js, and USE questionnaire for measuring usability: for evaluation of the Liquidizer.js algorithm in SPSS. The outcome of the study is threefold; to present the state-of-art survey of RWD technology, implement an algorithm Liquidizer.js that achieves an enhanced RWD, and register the Liquidizer.js in the jQuery Library under the GNU General Public License.

Index Terms – Responsive Web Design (RWD), jQuery, Fluid grid concept, Liquidizer.js and cascading style sheets (CSS)

1 INTRODUCTION

1.1 Background Study

In the more recent times the emerging technologies of computer science have spear headed the change of science fiction into science fact. Just a few years ago, technologies such as Internet of Things (IoT), cloud computing, fog computing, and just to mention a few, were science fiction and today are science facts. Most disciplines of sciences have reached their peak and are not experiencing any significant growth, but computer technology is a dynamic science that changes by the second. Computer technology is a science that had shape our past, shaping our present and would shape our future. Therefore, this study is intrigued by the computer technology dynamism, specifically in the area of Responsive Web Design (RWD) technology.

It is asserted by [14] that, RWD is a web design approach aimed at crafting sites to provide an optimal viewing experience; easy reading and navigation with a minimum of resizing, panning, and scrolling; across a wide range of devices (from mobile phones to desktop computer monitors). A site designed with RWD adapts the layout to the viewing environment by using fluid, proportion-based grids, flexible images, and CSS media queries, an extension of the @media rule.

Responsive Web Design was founded by Ethan Marcotte who is a developer and a web designer. He had a particular interest in architecture and he wanted to apply architectural principals in web design. Inspired by architecture way of thinking, he applied the architectural principal to web design; whereby a web site would adapt

itself to the users various devices (smart phones, laptops, desktop e.tc.). Hence, the idea of Responsive Web Design was coined

According to [18], Responsive web design is the terminology given to the concept of designing and developing a website so that the layout changes depending on the device/viewport on which the website is being viewed. By device, this could be a mobile phone, tablet, laptop, desktop computer, or even a smart TV.

According to [5], Responsive Web design is an approach that suggests that, design and development should respond to the user's behavior and environment based on screen size, platform and orientation. The practice consists of a mix of flexible grids and layouts, flexible images and an intelligent use of CSS media queries. As the user switches from their laptop to iPad, the website should automatically switch to accommodate for resolution, image size and scripting abilities. In other words, the website should have the technology to automatically respond to the user's preferences. This would eliminate the need for a different design and development phase for each new gadget on the market.

It is stated by [11] that; Responsive design is not a single technology but a set of techniques that allow web pages to serve the needs of both mobile and desktop users. The core components are: CSS @media queries, Fluid images and video, JavaScript, often triggered by window match Media, Server-side solutions, and Scalable Vector Graphic (SVG) to create resolution-free images.

It is suggested by [3] that, the following four core concepts to be apply for RWD implementation: firstly, the fluid grid concept – which calls for page element sizing to be in relative units like percentages, rather than absolute

units like pixels or points. Secondly, the flexible images – which calls also for sized in relative units, so as to prevent them from displaying outside their containing element. Thirdly, the media queries – which allows the page to use different CSS style rules based on characteristics of the device the site is being displayed on, most commonly the width of the browser. Fourthly, the RESS (Responsive Web Design + Server Side Components) – which in conjunction with client-side ones such as media queries can produce faster-loading sites for access over cellular networks and also deliver richer functionality/usability avoiding some of the pitfalls of device-side-only solutions.

It is further elaborated by [3] that, another name used to describe this set of techniques is Adaptive Web Design (AWD). The name would match more since the website really adapts to the device, rather than responding continuously to changes in its environment.

1.2 Types of Web Layouts Approaches

According to [19], classifies the web layouts as fixed-width layouts, liquid layouts, and elastic layouts.

1.2.1 Fixed-Width Layouts

It is described by [19] that, in fixed-width layouts, the width of the site is bound to a certain number of pixels. Generally, the measure chosen is 960 pixels. This is because with the passing of time, developers have found 960 pixels to be the best size for grid layouts, because the number is easily divisible by 3, 4, 5, 6, 8, 10, 12, and 15. However, the fixed-width layouts have some disadvantages. The designers who want to create a fixed-width website have to keep in mind that every aspect of their work has to be usable and clearly visible to a large number of screens, browsers, and devices. The wide variety of devices on the market at this time, as well as the consequently great variability of screen sizes makes creating one-size-fits-all content quite a challenging task, and arguably a challenge that outweighs the precision and control of fixed-width design.

1.2.2 Liquid Layouts

It is explained by [19] that, the first basic difference between the fixed-width type of layout and liquid layouts is the measurements of their size. The fixed-width layouts are measured in pixels, but liquid or fluid layouts, dimensions are defined in percentages, and as expected, this affords greater malleability and fluidity. In other words, by setting a percentage, one won't have to think about device size or screen width, and consequently, one can find a reasonable solution for each case because the design's size will adapt to the size of the device used. Liquid layouts are closely linked to media queries and special styles for optimization. Percentage-based widths alone will likely not be enough to accommodate one's design for a large variety of display sizes.

1.2.3 Elastic Layouts

It is elaborated by [19] that, elastic layouts are somewhat similar to liquid layouts. The main difference is once again the unit of measurement for size. The size indicator for elastic layouts is neither in pixels nor percentages; it's measured in ems. An em is the equivalent of the size (in pixels) defined in the font-size CSS rule. For example, if one styles text with a font-size of 20 pixels, 1 em would be equal to 20 pixels, 2 ems would correspond to 40, and so on. This types of layout gives the developer strong typographic control. Since the vast majority of layouts are predominantly populated with text, the precision of type treatments makes elastic layouts a strong contender for many projects. However, even with this type of solution, there is a risk of an unpleasant and unaesthetic horizontal scroll bar in some rare cases. The study is motivated to explore the fluid grid concept.

There exist various web development languages for implementation of responsive web design concepts. It is of paramount importance to discuss the strength and weakness of these languages, in order to present the most suitable one for Liquidizer.js implementation.

1.3 Web Development Technologies

According to [21], classifies the most common used technologies for web design are jQuery, Ajax, CSS and HTML.

1.3.1 jQuery

According to [7], jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

It is described by [16] that, jQuery is very compact and well written JavaScript code that increases the productivity of the developer by enabling them to achieve critical UI functionality by writing very small amount of code. It is a lightweight cross-browser JavaScript library. jQuery emphasizes interaction between JavaScript and HTML. It is used by 27% of the 10,000 most visited websites, jQuery is the most popular JavaScript library currently in use.

1.3.2 Ajax (Asynchronous JavaScript and XML)

According to [17], Ajax (Asynchronous JavaScript and XML) is a method of building interactive applications for the Web that process user requests immediately. Ajax combines several programming tools including JavaScript, dynamic HTML (DHTML), Extensible Markup Language (XML), cascading style sheets (CSS), the Document Object Model (DOM), and the Microsoft object, XMLHttpRequest. Ajax allows content on Web pages to update immediately when a user performs an action, unlike an HTTPrequest, during which users must wait for a whole new page to load.

It urged by [4] that, Ajax is not a programming language or a tool, but a concept. Ajax is a client-side script that communicates to and from a server/database without the need for a postback or a complete page refresh. Hence can be defined as; the method of exchanging data with a server, and updating parts of a web page - without reloading the entire page. Ajax itself is mostly a generic term for various JavaScript techniques used to connect to a web server dynamically without necessarily loading multiple pages. In a more narrowly-defined sense, it refers to the use of XMLHttpRequest objects to interact with a web server dynamically via JavaScript.

1.3.3 Cascading Style Sheets (CSS)

It described by [8] that, CSS is the language for describing the presentation of Web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language. The separation of HTML from CSS makes it easier to maintain sites, share style sheets across pages, and tailor pages to different environments.

1.3.4 Hyper Text Markup Language (HTML)

According to [8], Hyper Text Markup Language (HTML) is the language for describing the structure of Web pages. HTML gives authors the means to: publish online documents with headings, text, tables, lists, photos, etc.; retrieve online information via hypertext links, at the click of a button; design forms for conducting transactions with remote services, for use in searching for information, making reservations, ordering products, etc.; and include spread-sheets, video clips, sound clips, and other applications directly in their documents.

While, Extensible HTML (XHTML) is a variant of HTML that uses the syntax of XML, the Extensible Markup Language. XHTML has all the same elements (for paragraphs, etc.) as the HTML variant, but the syntax is slightly different. Because XHTML is an XML application, one can use other XML tools with it (such as XSLT, a language for transforming XML content).

1.3.5 Webkit

According to official website for the WebKit Open Source Project that, WebKit is an open source web browser engine, or the name of the OS X system framework version of the engine that's used by Safari, Dashboard, Mail, and many other OS X applications. WebKit is a layout engine software component for rendering web pages in web browsers. It powers Apple's Safari web browser and was previously used in Google's Chrome web browser. It provides a set of classes to display web content

in windows, and implements browser features such as following links that the user clicks on, managing a back-forward list, and managing a history of recently visited pages

1.4 Objective of the study

According to [10], it is asserted that web designers may not use fluid page designs for various reasons and further elaborated that, one of the reasons as being that: images, video, and other types of content with set widths, need to be set at multiple widths to accommodate different screen resolutions. The available framework solutions involve heavy customization, which discourage web designers to use RWD techniques. This has resulted into low adaption of the concept by web designers.

Hence, the study was inspired to address this gap by presenting an alternative enhanced approach to RWD by enhancing the existing algorithm BlocksIt.js to an automated algorithm coined Liquidizer.js that will be responsive and accessed by a single line of code to improve the adaptation of the technique by web designers. As a result of this, the study was directed and guided by the following specific objectives during the period of the study;

- 1) To conduct a literature study survey of the existing state-of-art RWD technologies.
- 2) To design an algorithm Liquidizer.js that achieves an enhanced RWD.
- 3) To develop and implement the algorithm Liquidizer.js that achieves an enhanced RWD.
- 4) To conduct experiments for testing, validating, and evaluating the Liquidizer.js algorithm.
- 5) To register the liquidizer.js algorithm in the existing jQuery library platforms.

2 METHODOLOGY OF THE STUDY

2.1 Algorithm Liquidizer.js Evaluation Method

In order to evaluate the algorithm Liquidizer.js, the following method was followed. The algorithm Liquidizer.js algorithm was given to web designers and shown how to use the model. They were instructed to include a line of code in a script tag to reference both the Liquidizer.js and jQuery.mini.js. The instructions were executed by coding the following line; `<script type = "text/javascript" src = "jquery.min.js"></script>` and `<script type = "text/javascript" src = "Liquidizer.js"></script>`. To conduct the research Table 3 shows the following web design households; Crablins Interactive, Jaffetek Computer Solutions and Mombasa Tech Community were engaged

TABLE 1: The Web Design Firms and Web Pages Tested

NO	Web Design Firms	Web Pages Tested
----	------------------	------------------

1	Crablinks Interactive	Sea View Academy
2	Jafftek Computer Solutions	Almutwafy ICT Consultancy
3	Mombasa Tech Community	SportsVision Magazine

The Crablinks Interactive; is a creative web design and optimization company based in Mombasa, Kenya. It designs and builds websites which depicts company's vision, brings out firms ideas and meets its business needs. It was chosen due to its rich portfolio in web design such as designing and developing in Lamu county website, Nawal center, shopping center in Mombasa with a customized e-commerce website, Shelter Tours, a Kenyan tour company, Sakafu Limited, a Kenyan floor safety company just to mention a few.

The Jafftek Computer Solutions; is a Kenyan based Software Development Company, focused on creating utmost qualitative, on time and cost effective software solution. It was chosen because of its development of successful online identities for various organizations in Kenya. It has vast of experience of over ten (10) years in designing website solutions, which meet customer expectations. It has specialized in providing comprehensive website design services in e-commerce, magneto designs and in joomla framework platform.

The Mombasa-Tech Community; is a Community Based Organization (CBO) which focuses on inspiring and developing upcoming innovative technology ideas through networking, technical training and support, and professional mentoring and coaching. It was chosen because, it brings together techpreneurs, entrepreneurs, developers, designers, and investors to provide them with opportunities to learn, share knowledge, be mentored, and initiate novel ideas which will lead to the development of new technologies in Kenya and globally at large. The study targets its professional mentors and coaches web designers, to conduct the review process. The web designers are then given a questionnaire to give their comments on observations they have made as regards the Liquidizer.js algorithm framework. The questions are based on software usability models.

2.2 Usability Model Types

This study presents usability perspective in most frequently used models of software quality characteristics, namely Factors Criteria Metrics (FCM) Model, and ISO 9126 Model. Furthermore, it discusses their limitations and strengths, in order to decide the best model to be adapted by the study

2.2.1 Usability in Factors Criteria Metrics (FCM) Model

Usability is an important factor of software quality. Usability has always been present, even in the very first

models of software quality. According to [20], the basic idea in all software quality models is to define software quality, which is a term that is too abstract to be studied directly, by dividing it into attributes (usually called quality factors). One of these factors is usability, a concept that is also abstract and therefore is usually divided into more specific attributes (usually called characteristics). In some models, these characteristics are further divided into sub-characteristics and so on, always with the same concept in mind: to divide an abstract term into better defined terms (terms that can be measured objectively or with a reduced level of subjectivity).

It is reported [15] that, usability is divided according to three Factors Criteria Metrics (FCM) model as: Operability, Training and Communicativeness. These criteria are associated only with the usability factor. Operability is associated to the user's effort for operation and operation control (for example mouse support, macro-commands, etc.). Training is associated to the effort required to teach the use of software to the user, while communicativeness is associated to how well the software communicates to the user the purpose for which it has been developed and the method to use it. Based on the FCM measurement method, each criterion could be associated with metrics in the form of questions allowing subjective "yes"

2.2.2 Usability in ISO 9126 Model

The usability in ISO 9126 Model is based on FCM model, international efforts lead to the development of the international standard ISO 9126 for software quality. ISO 9126 comprises of a basic set of 6 independent quality characteristics: Functionality, Reliability, Usability, Efficiency, Maintainability, and Portability. Unlike FCM, ISO 9126 is completely hierarchical and relates each sub-characteristic to only one of the basic characteristics.

According to [6], usability can be divided into understandability (which is the user's effort for recognizing the underlying concept of the software), learnability (which is the user's effort for learning how to use the software) and operability (which is the user's effort for operation and operation control like mouse support, macro-commands, etc.). ISO 9126 is currently used by many software developers to define quality goals and usability is always a prime concern in software quality. Table 4 summarizes the characteristics and sub-characteristics of ISO 9126.

TABLE 2: Source: ISO 9126 Characteristic and sub-characteristics (Abran, Khelifi, Suryan, & Seffah, 2003)

Characteristic	Sub-characteristic	Explanation
	Understandability	Does the user comprehend how to use the system easily?
Usability	Learnability	Can the user learn to use the system easily?
	Operability	Can the user use the system without much effort?
	Attractiveness	Does the interface look good?

After an extensive analysis of the three models of FCM Model, and ISO 9241 Model, it is clear from the above discussion that, FCM Model and ISO 9241 are enhanced models to form the ISO 9126 Model. Therefore, the study adapts the ISO 9126 model by using the quality characteristics of usability and sub-characteristics of understandability, learnability, operability and attractiveness to evaluate Liquidizer.js framework to be measured by web designers. Based on the FCM measurement method, each sub-characteristic is associated with metrics in the form of questions allowing the web designers a “yes” or “no” answers.

2.3 Algorithm Liquidizer.js Testing

Matt Kersley RWD testing tool is used for the study. It is described by [9], as a testing tool that allows viewing responsive website in various screen sizes simultaneously in a single screen, while building or designing websites. The study prefers this tool mainly because it shows all the screen resolutions side-by-side which makes it easier for debugging. It is an online tool, which can be accessed at <http://mattkersley.com/responsive/>. The Fig 1 below demonstrates how the interface of the Testing Tool works.

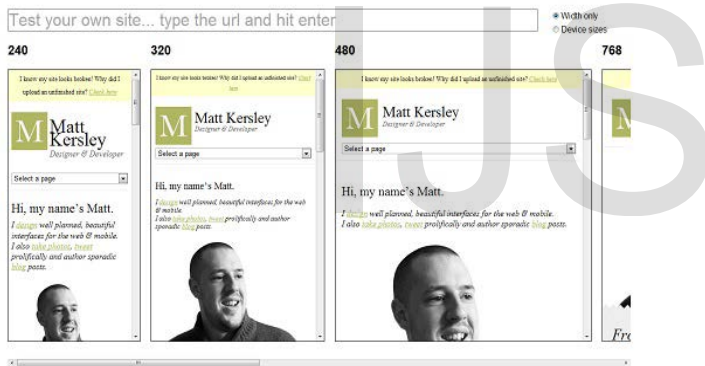


Fig 1: Matt Kersley Default Interface Adapted from (Kersley, 2012)

The tool has been built to help with testing responsive websites while designing and build. One can enter website's URL into the address bar at the top of the page (not browser's address bar) to test a specific page.

2.4 Algorithm Liquidizer.js Validation

The study used the Bersoft Image Measurement (BIM) tool for algorithm Liquidizer.js validation. BIM is a powerful image asset management tool that makes it easy to measure, acquire, store, compare and analyze digital images. BIM performs image analysis functions that include gray scale and 24 bits color measurements: angle, distance, perimeter, area, point, line, pixel profile, object counting, histogram and statistics. BIM supports DICOM, JPEG, TIFF, and most popular image formats. The Fig 2 shows the snap shot of BIM interface [2].

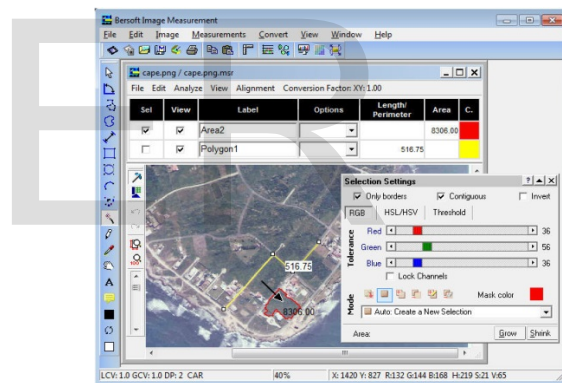


Fig 2: The BIM Interface adapted from (BIM, 1995)

3 DATA COLLECTION AND ANALYSIS

3.1 Introduction

As software becomes more and more pervasive, there has been a growing concern in the academic community and Business world about software quality. This concern arises from the acknowledgment that the main objective of software in industries is to balance price and quality to stay ahead of competitors. Some standard organizations, such as ISO and IEEE, tried the standardization of software quality by defining frameworks combining and relating software quality characteristics and sub-characteristics.

3.2 Measuring Usability with the USE Questionnaire

The Usability questionnaire was developed over time and it started out with a large pool of items. It was reported by [12] that, “The questionnaires were constructed as seven-point Likert rating scales. Users were asked to rate

agreement with the statements, ranging from strongly disagree to strongly agree. Various forms of the questionnaires were used to evaluate user attitudes towards a variety of consumer products. Factor analyses following each study suggested that users were evaluating the products primarily using four dimensions; Usefulness, Satisfaction, Ease of Learning and Ease of Use."

The questionnaires were constructed as seven-point Likert rating scales, e.g. from -3 (totally disagree) to +3 (totally agree), [12]. Therefore, the study adopts the USE Questionnaire for Measuring Usability to conduct a survey to evaluate the usability of the algorithm Liquidizer.js. The likert scale was coded as; 1 = "Strongly Disagree", 2 = "Disagree", 3 = "Slightly Disagree", 4 = "Neutral", 5 = "Slightly Agree", 6 = "Agree" and 7 = "Strongly Agree". The data collected was evaluated using Statistical Package for the Social Sciences (SPSS).

3.3 Data Collection

The study employed purposive sampling as its sampling technique to arrive at a sample of thirty (30) web designers. This concurs with a report by [12], who explains that, purposive sampling represents a group of different non-probability sampling techniques. Also known as judgmental, selective or subjective sampling, purposive sampling relies on the judgment of the researcher when it comes to selecting the units (e.g., people, cases/organizations, events, pieces of data) that are to be studied. Usually, the sample being investigated is quite small, especially when compared with probability sampling techniques.

Therefore the study purposively selected a sample of ten (10) web designers from each of the three (3) software households of; Crablins Interactive, Jaffetek Computer Solutions and Mombasa Tech Community to arrive at a sample of thirty (30) web designers. The web designers were instructed on how to use the Liquidizer.js. Moreover, the web designers were briefed on how to capture their perception on the Liquidizer.js using the questionnaires.

To evaluate the Liquidizer.js, the study conducted a survey to explore whether there is a relationship in the web designer's perception of the three dimensions of Usability; Usefulness, Satisfaction, and Ease of Learning (independent variables) and Ease of Use (dependent variable) of the Liquidizer.js. As a result of which the study presented the following hypothesis:

- a) H0 - There is no relationship between the perception of dimensions of usability and ease of use of the Liquidizer.js
- b) H1 - There is a relationship between the perception of dimensions of usability and ease of use of the Liquidizer.js

The questionnaires were immediately collected after the survey and analyzed using Statistical Package for Social Science (SPSS) version 20.

3.4 Data Analysis

To test the Liquidizer.js the study uses the Matt Kersley RWD Testing Tool. The tool presents visual output of the page in question either being responsive or not after or before implementation of the Liquidizer.js, by simultaneously simulating varying screen sizes of various devices.

To validate the Liquidizer.js the study uses Bersoft Image Measurement (BIM) tool. The tool is applied on both the generated digital images of frameworks of Liquidizer.js and BlocksIt.js algorithms. Three analytical tests of Lane Profile, Pixel Profile and descriptive statistics are conducted. The results are compared and interpreted.

To evaluate the Liquidizer.js descriptive statistics, correlation and regression analysis are conducted using Statistical Package for the Social Sciences (SPSS). The questionnaires are inputted in SPSS to create a model. The model is used to output results of different analysis tests.

4 FINDINGS OF THE STUDY

4.1 Introduction

In this section, the research findings and results of the study were presented into threefold; firstly, the testing of the Liquidizer.js by the Matt Kersley RWD Testing Tool Results, secondly, the validation of the Liquidizer.js by Bersoft Image Measurement (BIM) tool Results and thirdly, the evaluation of Liquidizer.js by Statistical Package for the Social Sciences (SPSS) Results. The SPSS tool conducted various tests to find relationship between variables, level of significance, reliability, and demographic data analysis. Specifically, the study used Cronbach's alpha test, descriptive statistics tests, chi-square, Pearson correlation and Regression analysis. As a result of this, the chapter is triggered and guided by the objective four of the study.

4.2 Results for Liquidizer.js Testing

The Matt Kersley RWD testing tool was used to simulate varying screen size from mobile phone to a main frame computer. The Fig 3 (a) presented the result before implementing the algorithm Liquidizer.js, while the Fig 3 (b) showed the findings after implementing the algorithm Liquidizer.js. The results are interpreted according to visual observation, whereby the page before integrating the Liquidizer was not responsive since the graphics and the menus are distorted and not fitting a single page. While Fig 3 (b) is responsive since the page can be viewed in single view without the graphics or menu being distorted.



Fig 3: (a) Before implementing Algorithm

Moreover, further responsive tests were conducted using Matt Kersley RWD testing and simulator tool on various screen sizes; 240 x 320 pixels smart phone, 320 x 480 pixel iPhone, 480 x 640 pixels tablet, 768 x 1024 pixels iPad - Portrait and 1024 x 768 pixels iPad - Landscape. The Fig 4



Fig 4: (a) Smart Phone (b) Small Tablet (c) iPad

4.3 Results for Liquidizer Validation

The study used Bersoft Image Measurement (BIM) tool to performed three validation analytical tests on the digital images generated by both Liquidizer.js and BlocksIt.js. The following are the tests which were conducted during validation.

- i. Lane Profile Test
- ii. Pixel Profile Test
- iii. Descriptive Statistics Test

4.3.1 Lane Profile Test

The lane profile test analyses the pixel intensity against the pixel position. Both algorithms; the Liquidizer.js and BlocksIt.js's digital images were analyzed and their results compared against each other's. The purpose of the test is to compare the pixel intensity of the two algorithms. High pixel intensity indicates that the digital image is of high quality and non-distorted while; low pixel intensity indicates that the digital image is of low quality and distorted. The high quality and non-distorted is inferred to as responsive, while the low quality and distorted is inferred to as non-responsive.



(b) After implement Algorithm

(a), 4 (b), 4 (c), 4 (d) and 4 (e) showed that, the liquidizer.js model is responsive at various screen sizes. The result can be explained by visual observation of the framework fitting in all screen size without being distorted.



(d) iPad-Portrait (e) iPad-Landscape

4.3.2 Lane Profile Results

The lane profile was conducted using Bersoft Image Measurement (BIM) on the model generated by BlocksIt.js. The results are shown in Fig 5 (a). It is observed from the results of four rows and columns pixel section of the digital image that the pixel position of each row and column divided in to 100px, corresponded to low pixel intensity of (R,G,B) (55, 55, 55) (shade of black) which was fairly constant distributed at these regions. The result infers that, most of the pixels are cramped together at these ranges. The three colors; red, green and blue are not evenly distributed hence displaying dark quality picture since the range values of pixel intensity are from (0,0,0) (black) to (128,128,128) (gray) to (255, 255,255) (white).

Therefore, the results can be explained by the visual observation of the distorted picture model, which is non-responsive. The non-responsive digital image will not fit in various screen sizes such as from mobile to framework computer without being distorted, while a responsive digital image will completely fit in various screen sizes and be non-distorted.

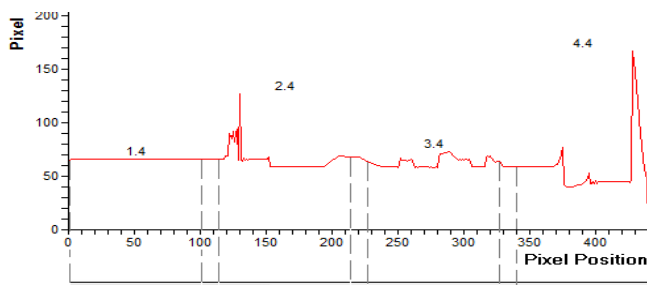


Fig 5: (a) BlocksIt.js Lane Profile

The lane profile was also conducted on the model generated by Liquidizer.js. The results are shown in Fig 5 (b). It is observed from the results of four rows and columns pixel section of the digital image that the pixel position of each row and column divided in to 100px, corresponded to high pixel intensity of (R,G,B) (160, 160, 160) (light gray) which was fairly constant distributed at the non-distorted picture framework, which is responsive. A responsive digital image will completely fit in various such screen sizes and be non-distorted hence displaying a high quality image.

From the above two results of Fig 5 (a) and (b) observations, it can be clearly concluded that the image generated by Liquidizer.js is more responsive as compared to image generated by BlocksIt.js. The conclusion is arrived after analyzing the results of the two algorithms. The BlockIt.js results indicated that the image is of low quality and distorted as compared to that of Liquidizer.js indicating the image of high quality and non-distorted. Therefore this infers that the Liquidizer.js is more responsive as compared to BlocksIt.js algorithm.

4.4 Pixel Profile Test

The pixel profile test evaluates the pixel values against the pixel distance of a digital image. Both the frameworks of Liquidizer.js and BlocksIt.js generated digital images which were evaluated for pixel profile.

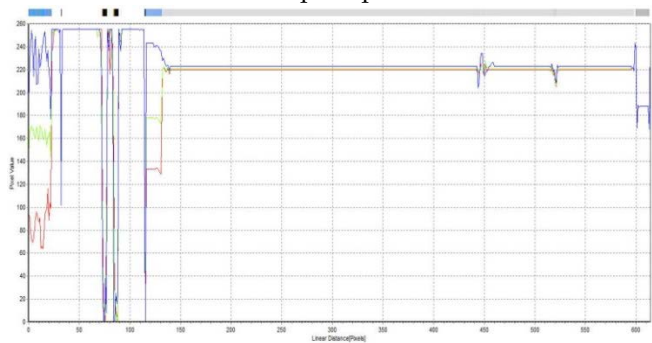
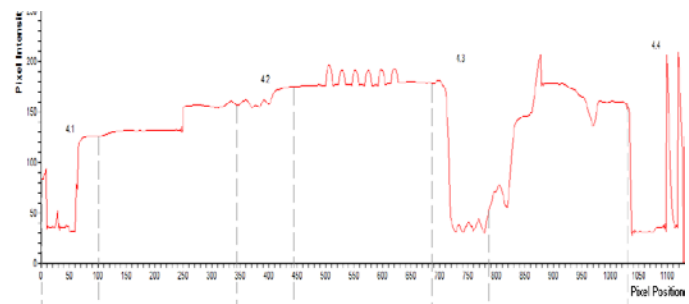


Fig 6: (a) BlocksIt.js Pixel Profile

The pixel profile was also conducted on the framework generated by Liquidizer.js. The results are shown in Fig 6 (b). It was observed from the results of the slice pixel section of the digital image that, the pixel distance of between 150px - 600px, corresponded to the pixel values of between 0 - 220, which are evenly distributed throughout the image. The result infers that,

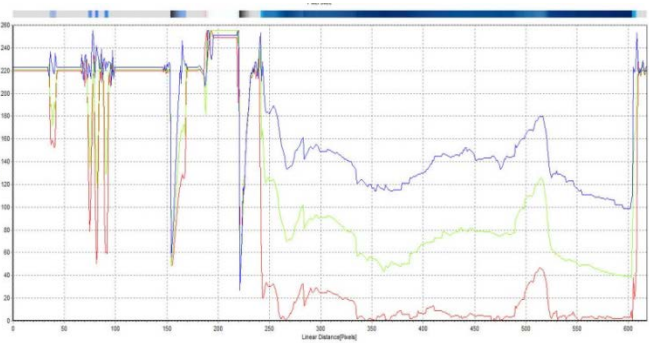


(b) Liquidizer.js Lane Profile

these regions. The result infers that, most of the pixels are highly dispersed at these ranges. The three colors; red, green and blue are evenly distributed hence displaying brighter quality picture since the range values of pixel intensity are from (0,0,0) (black) to (128,128,128) (gray) to (255, 255,255) (white). Therefore, the results can be explained by the visual observation of

4.4.1 Pixel Profile Results

The pixel profile was conducted using Bersoft Image Measurement (BIM) on the framework generated by BlocksIt.js. The results are shown in Fig 6 (a). It is observed from the results of the slice pixel section of the digital image that the pixel distance between 150px - 600px, corresponded to the pixel value of 220 which was fairly constant at this region. The result infers that, most of the pixels are cramped together at this range. The three colors; red, green and blue are not evenly distributed hence displaying poor quality picture. Therefore, the results can be explained by the visual observation of the distorted picture framework, which is non-responsive. The non-responsive inferred that, web page framework will not automatically detect various devices' screen sizes when accessed by different users without page being disfigured, while a responsive web page framework will automatically detect various devices' screen sizes and be non-disfigured.



(b) Liquidizer.js Pixel Profile

most of the pixels are fairly disperse over the region. The three colors; red, green and blue are highly distributed hence displaying good quality picture. Therefore, results can be explained by visual observation of the non-distorted picture which is responsive.

From the above two results of Fig 6 (a) and (b) observation it can be clearly concluded that the image

generated by Liquidizer.js is more responsive as compared to image generated by BlocksIt.js

4.5 Results for Liquidizer.js Evaluation

The descriptive statistics tests were conducted on both the algorithm Liquidizer.js and BlocksIt.js. The study evaluated and compared the skewness of the algorithm to ascertain the degree of distortion. The Skewness measure indicates the level of non-symmetry. If the distribution of the data is symmetric then Skewness will be close to 0 (zero). The further from 0, the more skewed the data. A negative value indicates a skew to the left.

4.5.1 Statistical Results

The Table 3 shows various descriptive statistics test, the study had a particular interest on values of skewness. The

TABLE 3: Descriptive Statistics

Statistics	Liquidizer.js	BlocksIt.js
N	737580	416021
MaxValue	169852	133517
Min	0	0
Max	255	255
Median	175	222
Mode	221	221
Mean	159.191962905719	213.507094593783
St. Dev	64.3144448743522	46.2189489446875
Skew	-0.488686363274612	-2.07774911184475
Kurtosis	1.93133960607452	8.51071458293398

4.6 Evaluation Results

The questionnaires were coded in SPSS to create a statistical model. The model was used to conduct various analytical and statistical tests on the information collected from web designer's perception of the Liquidizer.js algorithm.

4.6.1 Survey Respondent Perception

Mean in statistical analysis and probability, are used to refer to measures of the central tendency either of a probability distribution or of the random variable characterized by that distribution. Therefore, the study sampled 30 respondents, and evaluated the distribution of the web designer's perception of the three dimensions of Usability; Usefulness, Satisfaction, Ease of Learning (independent variables) and Ease of Use (dependent variable) of the Liquidizer.js. The data was analyzed using descriptive statistics by computing means of each Usability determinates. The means were then evaluated against the questionnaires approximate survey value coded to the survey labels, as shown in Table 4

TABLE 4: Respondent's Perception of the survey

results indicate that BlockIt.js had a skewness value of -2.07774911184475 which infers a large negative skew from the mean, media and mode as compared to Liquidizer.js skewness value of -0.488686363274612 which is a score near to zero inferring to near perfectly normal distribution. The normal distribution of the primary colors (RBC) indicated that the pixel intensity is sparsely distributed on whole image. This inferred that a high quality image which is non-distorted. These results concurred with lane profile test above. Therefore, results can be explained by visual observation of the high degree of distorted picture quality of BlocksIt.js digital image as compared to Liquidizer.js digital image.

Respondent's Perception of the survey. The responses were mapped to questionnaires as; 1 = "Strongly Disagree", 2 = "Disagree", 3 = "Slightly Disagree", 4 = "Neutral", 5 = "Slightly Agree", 6 = "Agree" and 7 = "Strongly Agree".

The findings showed that, the respondents asserted an average (mean = 6.50) for Satisfaction inferring to Survey Value (7), coded as "Strongly Agree" on the likert scale response, average (mean = 6.37) for Ease of Learning, inferring to Survey Value (6), coded as "Agree" response, average (mean = 6.43) for Ease of Use, inferring to Survey Value (6), coded as "Agree" response, and average (mean = 6.45) for Usefulness, inferring to Survey Value (6), coded as "Agree" response.

The inferential statistics indicate that, average (mean = 6.44) for Grand Perception of all Usability determinates, inferring to Survey Value (6), coded as "Agree" response. Therefore, it can be inferred that, respondents asserted a general agreement to the usability determinates of; Usefulness, Satisfaction, Ease of learning (independent) which positively influence Ease of Use (dependent) of the Liquidizer.js, hence leading to a linear relationship.

Usability Determinates	N	Mean	Survey Value	Survey Label
Satisfaction	30	6.50	7	Strongly Agree
Ease of Learning	30	6.37	6	Agree
Ease of Use	30	6.43	6	Agree
Usefulness	30	6.45	6	Agree
Grand Perception	30	6.44	6	Agree

There are various contributing factors that may lead to linear relationship between variables, but more conclusive and affirmative tests such as correlation analysis should be consider for reliable output results

4.6.2 Correlation Analysis

The study conducted correlation analysis and computed four usability determinates of; Usefulness, Satisfaction, Ease of leaning (independent variables) which positively influence Ease of Use (dependent variable) of the Liquidizer.js for 30 respondents. The results are shown in Table 5 and suggested that, there was strong relationship of .817 between dependent variable Ease of Use and independent variable Satisfaction, which is statistically significant at $r(30) = 0.817, p < .01$, two-tailed. This infers that, the strong relationship between the variables was statistically significant at 99%. Moreover, the result further suggested that, there was a strong relationship of .814

between dependent variable Ease of Use and independent variable Ease of Learning at $r(30) = 0.814, p < .01$, two-tailed.

This also infers that, the strong relationship between variables was statistically significant at 99%. Finally, the result suggested an existence of a strong relationship of .630 between dependent variable Ease of Use and independent variable Usefulness at $r(30) = 0.630, p < .01$, two-tailed. This also infers that, the strong relationship between variables was statistically significant at 99%.

In summary the result infers that, there was a strong positive linear relationship between the usability determinates of; Usefulness, Satisfaction, Ease of leaning (independent variables) and Ease of Use (dependent variable). The increase of Usability determinates, increases the Ease of Use. The Table 5 summarizes the correlation output computed by SPSS Version 20 for Usability Determinates of Ease of Use.

TABLE 1: Correlations for Usability Determinates of Ease of Use

		Satisfaction	Ease of Learning	Ease of Use	Usefulness
Satisfaction	Pearson Correlation	1	.424*	.817**	.630**
	Sig. (2-tailed)		.020	.000	.000
	N	30	30	30	30
Ease of Learning	Pearson Correlation	.424*	1	.814**	.551**
	Sig. (2-tailed)	.020		.000	.002
	N	30	30	30	30
Ease of Use	Pearson Correlation	.817**	.814**	1	.630**
	Sig. (2-tailed)	.000	.000		.000
	N	30	30	30	30
Usefulness	Pearson Correlation	.630**	.551**	.630**	1
	Sig. (2-tailed)	.000	.002	.000	
	N	30	30	30	30

*. Correlation is significant at the 0.05 level (2-tailed).

**. Correlation is significant at the 0.01 level (2-tailed).

Therefore, it was clear from the discussion that correlation can only indicate the presence or absence of a relationship, not the nature of the relationship. Correlation is not causation. There is always the possibility that a third variable influenced the results. Therefore, other

confirmatory tests such as regression should be conducted for reliable and conclusive results.

5 RECOMMENDATIONS

Even though the findings of the study showed that, the algorithm Liquidizer.js implementation was effectively responsive, an enhanced version is recommended whereby more dynamic attributes to be incorporate to increase acceptability by the web designers. Moreover, the study revealed that, the future development of the algorithm should emphasis on the usefulness of the web designers, since the variable usefulness contributes negatively for easy of usability and adaption.

In the future studies the algorithm can be improved by considering development of algorithm using CSS only, since as a layout language, CSS is more creative and flexible in design as compared to JavaScript. Moreover, the study recommended the optimization of the algorithm to increase the page loading time. Finally, the evaluation sample used in the study consisted of a limited numbers of web designers due to budgetary and time constraints. These constraints may have introduced some biasness in the study's findings. Therefore, the study recommended for a more thorough global research in order to explore further the usability and adaptability of the algorithm Liquidizer.js by web designers.

BIBLIOGRAPHY

- [1] Abran, A., Khelifi, A., Suryan, W., & Seffah, A. (2003). Usability meanings and interpretations in ISO standards. *Software Quality Journal*, 11(4), 325-338.
- [2] BIM. (1995). Bersoft Image Measurement. Retrieved May 13, 2015, from <http://image-measurement.com/>
- [3] Doyle, M. (2011, September 30). Responsive Web Design Demystified. Retrieved May 3, 2014, from [http://www.elated.com/articles/responsive-web-design-demystified./](http://www.elated.com/articles/responsive-web-design-demystified/)
- [4] Fote, W. (2013). What is Ajax and Where is it Used in Technology? Retrieved September 22, 2014, from <http://www.seguetech.com/blog/2013/03/12/what-is-ajax-and-where-is-it-used-in-technology>
- [5] Harb, E., Kapellari, P., Luong, S., & Spot, N. (2011). Litratione Study: Responsive Web Design. Retrieved from <http://courses.iicm.tugraz.at/iaweb/surveys/ws2011/g3-survey-resp-web-design.pdf>
- [6] ISO/IEC 9126-1. (2000). Product Oriented Standards.
- [7] John, R. (2006). jQuery. Retrieved from <http://jquery.com/>
- [8] Keio, E. M. (2013). HTML & CSS - W3C. Retrieved September 22, 2014, from <http://www.w3.org/standards/webdesign/htmlcss>
- [9] Kersley, M. (n.d.). Matt Kersley | Designer & Developer. Retrieved May 12, 2014, from <http://mattkersley.com/>
- [10] Knight, K. (2009, June 2). Fixed vs. Fluid vs. Elastic Layout: What's The Right One For You? | Smashing Magazine. Retrieved from <http://www.smashingmagazine.com/2009/06/02/fixed-vs-fluid-vs-elastic-layout-whats-the-right-one-for-you/>
- [11] Knight, K. (2011). Responsive Web Design: What It Is and How To Use It. Retrieved June 29, 2014, from <http://www.smashingmagazine.com/2011/01/12/guidelines-for-responsive-web-design/>
- [12] Lund. (2012). Purposive sampling | Lærd Dissertation. Retrieved April 8, 2015, from <http://dissertation.laerd.com/purposive-sampling.php>
- [13] Lund, A. M. (2001). Measuring Usability with USE Questionnaire. *STC Usability SIG Newsletter*, 8(2).
- [14] Marcotte, E. (2010, May 25). Responsive Web Design. A LIST APART. Retrieved from <http://alistapart.com/article/responsive-web-design>
- [15] McCall, J. (1977). Factors in software quality. *Tech. Inf. Service. Springfield*, 1,2,3(AD/A-049-014), 015 - 055.
- [16] Narayan, S. (2011). What is jQuery and How to Start using jQuery? - CodeProject. Retrieved September 22, 2014, from <http://www.codeproject.com/Articles/157446/What-is-jQuery-and-How-to-Start-using-jQuery>
- [17] Rouse, M. (2007). What is Ajax (Asynchronous JavaScript and XML)? - Definition from WhatIs.com. Retrieved September 22, 2014, from <http://searchwindevelopment.techtarget.com/definition/Ajax>
- [18] Solanki, K. (2012, May 17). Responsive Web Design - Ideas, Technology, and Examples. Retrieved from <http://www.onextrapixel.com/2012/05/17/responsive-web-design-ideas-technology-and-examples/>
- [19] Tranfici, A. (2013, January 29). Responsive Web Design: Fluid Layouts. Retrieved from <http://www.sitepoint.com/responsive-web-design-fluid-layouts/>
- [20] Xenos, M. (2001). Usability Perspective in Software Quality. In *Usability Engineering Workshop*. Southern Cyprus.
- [21] Y. Na, D. (2012). The What, Why, and How of Mobile Applications. *Noblis Publication*.