**ABSTRACT**

**WORDS WORLD APP**

Education is evolving and is taking advantage of the digital society. Classrooms, study material, communication, etc. are in their traditional senses fading away and being replaced by their digital counter parts. Because of personal computers, faster Internet, and instant information access the need for Web-based education services has appeared.

In literature, traditional education is commonly referred to as “face-to-face” education. Most of us know what this type of education means; school houses, books, pens and papers, and so on. R. Hiltzet a briefly defines traditional education as “face-to-face courses using object visit, teacher –centered pedagogy”.

*Words World App* is an online platform for learning. It offers education and supplies all the necessary tools to learn whenever a user chooses to, and by his or her own preferences. In an on-line environment such as *Words World App* it seems suitable to lighten up the education and adopt the paradigm of “learning by playing”. Therefore it was decided to add a game to the platform.

Since the word quiz was a part of a greater project there were some parts that had already been decided on. A graphical profile had already been developed when this project was initiated. The intention for the profile was that it should be used for *Words World App* and all its parts. This meant that the word quiz also had to be designed according to it.

*Words World App* runs on a Front End platform. Therefore, so does the word quiz, and hence the word quiz must not be able to communicate with the server in order to keep track of players and games.

The word quiz should be founded on the same principles as *Words World App* and fit its environment, not only graphically, but also in terms of user friendliness, educational support, and level of entertainment.

Student name: K.JAYALAKSHMI

Reg.no:I19171017

Signature of the Guide

Dr. R. SATHISH BABU

Assistant Professor/Programmer

**CHAPTER - 1**

**INTRODUCTION**

Education has been evolving to the point of today’s distance education where the school is no longer just a physical place to read and write. Nowadays students can access necessary material, study, communicate with fellow students and teachers, and submit homework and such from almost anywhere.

Because on-line education is mostly exploited by universities and other entities for higher education and because of easy access and the possibility of studying whenever the user wants to, possibilities have evolved for Web-based education services. *Words World App* is such a service. It offers its users the possibility to study and learn on the user’s own demands. It will support and provide all the necessary tools to learn to read and write.

A game is a type of play where participants follow defined rules. (Houghton et al., 2013) discusses educational games as the utilization of games to support teaching and learning. Games can be used as a support tool to complement traditional teaching methods to improve the learning experience of the learners while also teaching other skills such as following rules, adaptation, problem solving, interaction, critical thinking skills, creativity, teamwork, and good sportsmanship. Learning should not feel dull and it should not only mean rote memorization where students learn and grasp concepts through repetition or cramming. Teachers can take advantage of the energy and innovative thinking that is provided by using technology in learning to improve student performance.

There are four points:

i. To develop an enthusiastic learner.

ii. To motivate and engage students.

iii. Reduce monotonous learning methods.

iv. To help students with focus, self-esteem, and memory.

Limited time to experiment. Using open source applications–Mood is an open source application which offers a demo mode where teachers can edit and upload content. The demo period expires after 60 minutes and all content will be reset. Pro Profs is also on open source application with restrictions which limit the

application development process.

**1.1 Overview of Games in Education**

Many different types of educational games are being applied and used in educational institutions, schools and homes. Using games in education mostly focuses on improving critical thinking skills while teaching a particular subject, by allowing students to think outside the box as they follow rules. There are other games that can be used which limit to improving knowledge in a specific subject and the most popular ones are math games. (Yue, & Zen, 2009) discussed that games like chess cannot be viewed as educational games as these improve logic skills, reasoning, and other traits valued in education but they are not considered educational because they do not deliver content or relay curriculum material. Games that incorporate curriculum content or other educational material are referred to as educational games (Michel, 2016).

**1.2 Roles of Games in Education**

(Boyle, 2011) posits that games play a vital role in building student’s self-confidence. As educational tools, games are constructive as they liven up teaching methods which are normally considered dull and boring.

**i**. **Engage students**

The major role of applying technology is to engage students and to encourage students to participate. The use of games in education plays an important role in engaging students by encouraging hands on approach. Journal of Education and Practice [www.iiste.orgISSN](http://www.iiste.orgissn/) 2222-1735 (Paper) ISSN 2222-288X (Online)Vol.8, No.15, 2017

**ii. Help students remember**

The use of games in education aims to help students remember what they have learnt as active participation is encouraged. Learning should not mean rote memorization but students can use games to remember the critical points which they can apply in their examinations as well as in real-world situations.

**iii. Visual and computer literacy**

This is something which is vital in light of the fact that we live in a world which is ruled by innovation. By playing games, students gain visual and computer literacy skills which will prepare them for the world of work.

**iv. Rule following and problem-solving skills**

Game drills are based on rule following and students are required to follow rules in order to achieve a high score and move to the next stage. Students can easily apply this knowledge in real world situations as they are encouraged to think outside the box

.

**v. Beneficial for students with attention disorders**

Using games can help capture student’s attention as this is considered to be a fun way of learning. Research conducted has discovered that web based games can assist kids who experience attention problems.

**vi. Teach other skills**

Games can also be used to teach other skills such as critical thinking, problem solving, sportsmanship, interaction and collaboration with peers. This helps in creating less stifled individuals who are not limited but can adapt to any real world situation.

**1.3 Disadvantages of using Games in Education**

However, gaming in education has set backs which need to be addressed.

i. Providing a platform for students to play revision games becomes a challenge when teachers or instructors cannot control such an environment. Students can have access to other platforms which are harmful.

ii. Students who rely on games are often secluded from real life interaction.

iii. Using computers and other electronic devices can cause health hazards such as eye strain and other physical problems.

iv. The technologies required for full participation can be quite expensive and this can create a gap between the students who have access to the technologies and those who do not have access. The diagram below summarizes the role of games in education.

**CHAPTER - 2**

**PROBLEM DESCRIPTION**

The project treated in this report is part of a larger project called *Words World App*. This is a project in progress with the purpose of developing an on-line education environment. The goal is to establish an internationally leading tool for education, for private persons, schools, and companies. The business idea is to introduce a new education concept for learning where a user has access to a pedagogic and rich education tool. It is an interactive platform. In the first version of *Words World App* will be the web interface along with all of the parts of the web site must support.

The education environment has a few fundamental principles such as interactivity, user friendliness, and that it should be fun to use. One goal is also to have the users learn in their own pace, and by the means of their choice. The target-group for *Words World App* is everyone with an interest in languages, which are people in the age’s from 12 and up.

The various tools in *Words World App* must fulfill the principles and at the same time be consistent in design and contribute to the learning experience.

**2.1 TASK MANAGEMENT**

The focus forth is project was to develop a word quiz for *Words World App.* The purpose of the quiz is to provide an entertaining alternative to rehearsing and practicing words and phrases, as well as to supply a possibility for the users to test their knowledge by themselves. There are several aspects to take into consideration when developing a game.

* The GUI (*Graphical User Interface*) has to be designed with the whole target group in mind.
* Different types of input (keyboard, mouse) and output (audio, video, image) need to be supported.
* Game rules such as scoring and timing.
* Implementation, the game should run smoothly without having the player wait too much during loading times and synchronization.
* Data bases and communication, technologies to use.

Since the word quiz was a part of a greater project there were some parts that had already been decided on. A graphical profile had already been developed when this project was initiated. The intention for the profile was that it should be used for *Words World App* and all its parts. This meant that the word quiz also had to be designed according to it.

*Words World App* runs on a Front End platform. Therefore, so does the word quiz, and hence the word quiz must not be able to communicate with the server in order to keep track of players and games.

The word quiz should be founded on the same principles as *Words World App* and fit its environment, not only graphically, but also in terms of user friendliness, educational support, and level of entertainment. Furthermore there were a number of goals and requirements for the GUI of the game, the game rules, and the implementation:

* The design of the GUI should fit the design of Words World App. It should be easy to understand and use. The texts and the click-able options displayed in the GUI should be descriptive and intuitive.
* There should exist some kind of help for a novice player that does not understand how to play the game.
* The game configuration consist of game length (in terms of number of questions), category for the questions, and by choosing to play with words or/and phrases.
* Questions should be presented as texts. A user may answer by type the correct letters in a free-text field.
* The score for each question should depend on how fast the player is answering. Furthermore the game should contain some kind of bonus system.
* All texts displayed in the GUI should be dynamically adapted to the users
* .The application should be developed in PIXIJS and JAVA SCRIPT.

The main goal for the project was that the quiz should work properly in time for the official launch of words world app.

**2.2 METHODS**

The project was divided into three phases, graphic design/game design, implementation, and report writing/in depth study. In order for the work to run as smoothly as possible, some time was spent setting up a time schedule with weekly goals. The first phase was the graphic and game design phase. The graphic design and the game design processor a somewhat in parallel because the graphic design was dependent on the game design to some extent, and because the scheduled Turbo Mode could benefit from treating both. By the end of this phase the goal was to have a design proposal.

**2.3 RELATED WORK**

The design proposal was not be thought of as a final design but rather as a template For the following implementation that could be revised. The design was based on a Turbo Mode session and usability testing.

The implementation phase was conducted with continuous communication with the developers of the *Words World App* project. A flow chart was created in order to make the implementation easier.

The tools used to develop the game were PIXIJS and JAVASCRIPT, HTMLL, and CSS.

The whole developing process somewhat followed a model of the design process (seeFigure4.1) and was documented in a working diary. The in-depth study was written as a part of the matters thesis which this project also is a part of. The scope for the in-depth study was chosen to fit the project.

* **Related Work**

There are also other commercial language education platforms. A fellow student conducted a market survey and identified the biggest rivals to *Words World App* as CD-ROM based courses. They have existed for a long time and are mostly published by large and famous publishing companies. There also exist sound books and such off-line material. When it comes to on-line material, there are less competition. The courses that exist are often two languages only. The tools are often very one-sided, with a number of text-based classes and perhaps some sound files. According to the student, the general standard for on-line learning is low.

The biggest difference between *Words World App* and the previously mentioned rivals are that *Words World App* is on-line and makes all data available on-line. Another difference is the importance of communication. Users are encouraged to communicate with each other and the teachers. A final difference is the arsenal of tools which *Words World App* provides to its users.

Even though the supply of on-line education is low, the supply of educational games on-line is huge. Strict learning games do however not exist to any larger extent. Games that make use of language however exist in large numbers. For this thesis a few of these became important, both a survivals and as inspiration. Some of them are listed below.

* [www.vetgirig.nu](http://www.vetgirig.nu/)
* [www.betapet.se](http://www.betapet.se/)
* [www.trivia.se](http://www.trivia.se/)

**CHAPTER - 3**

**HOW DO GAMES CONTRIBUTE TO MODERN EDUCATION?**

**3.1 INTRODUCTION**

Present education involves web based education environments. It also includes Internet accessed materials such as tutorials, literature, tasks, etc. A great deal of the traditional educational models’ tools and resources are digitalized. This platform opens up for new learning tools and among them, digital games.

**3.2 THE EVOLUTION OF EDUCATION**

Education is currently in an evolutionary state. It has been drastically evolving since the dawn of new technologies such as the Internet and personal computers. These new technologies has brought with them new means of communication and faster access to information.

**3.2.1 TRADITIONAL EDUCATION**

In literature, traditional education is commonly referred to as “face-to-face” education [7].Most of us know what this type of education means; school houses, books, pens and papers, and so on. R Hiltzet al. briefly defines traditional education as “face-to-face courses using objectivist, teacher-centered pedagogy” [7].

R Slavin calls traditional education the classroom instructional technology, which he describes as a combination of three elements:

* Task structure
* Reward structure
* Authority structure

The task structure is the activities that make up the school day. Lecture, class discussion, and seat work are task structures used in classrooms. Rewards may include grades, teacher approval, and tangible rewards. The authority structure refers to the control that students exercise over their own activities [18].D Leonard describes traditional education in what he calls “the industrial age model”. He compares the school model to the model of a factory consisting of administrators managing the buildings, the faculty, and the staff. The faculty makes sure that the workers (students) produce and is tested to meet some criteria. In such a factory there is no need for workers to share knowledge, think critically, and exercise creative problem solving [13].

Because of the ways of information distribution and communication, new need shave emerged. M. Resnick believes that it is no surprise that people see a natural connection between education and computers. Computers and Internet enable people to transmit, access, represent, and manipulate information in new and fast ways. Because education is associated with information and computers are associated with information, the two seem to make a perfect match [17].Leonard points out that, today, there is a need for workers (students) to share knowledge, think critically, and exercise creative problem solving. Furthermore he states that the movement from campus-based learning to web-based distance education is inevitable [13]. In another line of thinking, D.Klaila argues that conversation is a factor that draws people into learning. When they interact, learners can help each other work through simulations, games, and other interactive exercises and transfer what they have learned to the work place [11].

This brings us to the present time, and present time education. Let us call it modern education to keep things simple.

**3.2.2 MODERN EDUCATION**

The modern education mainly revolves around the World Wide Web [14, 22].

B. Means et al. state that on-line learning is one of the fastest growing trends in educational uses of technology [14]. S. Tucker claims that distance education is becoming a more vital part of the higher education family. Just about every major American university offers these types of courses [22].

An important point of interest among many authors is mobile access to learning and how Internet and the World Wide Web greatly increase the potential for reaching learners around the world [14]. Distance education is less a philosophy and more a method of education, according to A. Bates. Students can study in their own time, at the place of their choice (home, work, or learning center), and without face-to-face contact with a teacher. He states that technology is a critical element of distance education [1]. Tucker argues that even thought distance education reaches a broader student audience, better addresses student needs, saves money, and uses the principles of modern learning pedagogy, it does not mean that it is superior to traditional education. She does, however, believe that it is not worse than traditional education and that it can be an acceptable alternative [22]. The modern education model is not a completely, brand new model. Anything that evolves, keeps the best part of the old model and develops new elements to replace the not so good elements in the old model. Thorne suggests Blended learning as then ex-model.

It is basically just what it sounds like. It blends on-line learning and e-learning with more traditional methods of learning and development [21].

**3.2.3 E-LEARNING**

A recurring word in literature regarding modern education is e-learning. Learning is a consequence of education, it is natural that learning has changed along with education which is now found on-line. Horton presents a simple definition of e-learning which reads as follows:

* “E-learning is the use of information and computer technologies to create learning experiences” [9].

Klail a states that e-learning should make use of tools such as email, chat rooms, and Web postings to pick up tips, review strategies, and discuss out comes [11]. Horton adds that e-learning makes use of computer and network technologies to the task of education. Several other definitions of e-learning exist. Some hold that e-learning is limited to what takes place within a web-browser without the need for other software or learning resources [9].

**3.3 GAMES AND EDUCATION**

Pesky holds that one of the biggest concerns in all formal learning, whether it is in a classroom, on-line, distance, or e-learning, is to keep the students motivated. To have the students stick to the learning process to the end [16]. He adds schools or corporations, instructor-led or computer-based to the list of un-engaging parts of learning [15]. Furthermore he explains the de-motivation by claiming that all learning requires effort, and, like crime, people rarely do it without a motive [16].

**3.3.1 GAME DEFINITION**

Let’s take a closer look at games. Games could be anything from playing Tag with friends to playing Solitaire by oneself. One consistent characteristic for all games is that they have rules. Games are unique in that they have different set of rules that usually constrain action and are forcing players to manage resources and make trade-offs [20].Furthermore, some argue that a game may be defined as an activity that is engaging and entertaining, usually challenging, and an activity in which interaction with others usually occurs [23].

To separate the different types of games, let us put computer- and video games, and all electronic games in a category of their own, and let’s call it digital games.

D. Burgos et al hold that a digital game is a game played in an electronic platform fulfilling the following features [2].

* It is voluntary
* Imaginary
* Limited in time and space
* It follows a set of rules
* It provides an uncertain solution because of differences in the people playing

the game.

J. Kirriemuri and C. McFarlaneet al. point out that games that are developed for educational purpose all have some similarities. They are developed to be used as a tool to make learning fun and a belief that learning through doing in games offers a power full earning tool [10].

**3.3.2 ENGAGEMENTAND MOTIVATION**

Squire states that digital games are routinely listed as the most important and influential medium by those under the age of 35.Games are a powerful socializing force; those who play computer and video games have different attitudes about work, play, and their co-workers than do their peers [19]. H. Dreher et al believe that the target group is larger than that and that virtually all users from ages 4 to 60 plus are the market’s target groups [5].

While game players often have longer-term goals such as beating the game, and games also offer rewards such as scores, prestige, and prizes, it turns out these are not the principal reasons people play games [16]. People play games because the process of game playing is engaging. In fact, the top two reasons people say they play interactive games, according to the Interactive Games Association, is because they are challenging and relaxing [16].

The reason computer games are so engaging is because the primary objective of the game designer is to keep the user engaged. They need to keep that player coming back, day after day, for 30, 60 even 100+ hours, so that the person feels like he has gotten value for his money. That is their measure of success. The goal of keeping learners engaged is not the primary concern of educators. The primary goal of educators is to instruct, to get the material across. Learners are typically trapped, either physically in a class room, or by their goals. Fun in the learning process creates relaxation and motivation. Relaxation enables learners to take things in more easily and motivation enables them to put forth effort without resentment. Given this, it certainly makes sense that fun and learning should go hand in hand [16].Most of us would recognize that’ play’ very often has something to do with enjoyment and relaxation. That its meanings and relationships are different from those in the society around and that the consequences of what happens in the play world are not directly relevant to the real world [4].

Another factor regarding games and education is competition. Competitive learning activities are among the various learning activities that play a significant role in on-line learning environments. A competitive learning environment obviously stimulates different feelings in winners and losers, and it is imperative to consider how to design such an environment so as to motivate users [3]. Humans are instinctively competitive. On-line competitive learning activity is important in on-line learning, and thus it is imperative to consider how to design an on-line competitive learning environment that motivates both winners and losers. A competitive learning environment naturally stimulates different feelings in winners and losers. Winning is the most powerful confirmation of success, while losing represents a crisis of being extinct [3].

The attitude of today’s children toward their video and computer games is the very opposite of the attitude that most of them have toward school. This is the very attitude we would like our learners to have: interested, competitive, cooperative, result-oriented, actively seeking information and solutions. It therefore makes a great deal of sense to try to merge the content of learning and the motivation of games, and this is indeed what is happening [15].

**3.3.3 PLAYING AND LEARNING**

Another reason to put games in education are the concepts of “learning by playing” or “learning by doing”. Digital game-based learning is an alternative that is being used with amazing and increasing success. A new learning paradigm, learning via play, is gradually emerging [15].

Dreheret a point out that babies and young children react very effectively to the concept “learning by playing”. Such as imitation, and “trial and error”. They are in an environment where they are allowed to make mistakes. Unfortunately, particularly mistakes are not very often allowed in educational environments. To deal with that problem, games may provide artificial environments, where mistakes and “trial and error” are allowed. In general, people like to play games for enjoyment and pleasure. Obviously, games may strengthen different sorts of skills by playing in a safe virtual environment following a given set of rules [5].

Kirriemuiret a hold that learning through direct experience has, in many contexts, been demonstrated to be more effective and enjoyable than learning through, what they call “information communicated as facts” [10].

Pesky believes that small groups of trainers, teachers, content experts and game de-signers working together can create experiences that will radically improve the learning, and ultimately the competence and behavior of thousands, and potentially millions of learners[15].

Games have tremendous educational potential. A good educational game can enable players to explore ideas in virtual worlds. Good games are about choices and consequences, and good educational games force players to form theories and test their thinking against simulated out comes [20]. Games and simulations let people learn by playing. Games for learning can be fun, but they are always purposeful. They teach first and entertain second. Learning games can draw on the established conventions of quiz shows, board games, and video games to arouse curiosity and harness competitive urges. Games and simulation sallow learners top practice tasks, apply knowledge, and infer principles, all while having fun. Games and simulations may provide a complete model of a real world system or just a rapid-fire series of questions to answers [9].

Wright exemplifies the use of games for language learning. It is hard work and one must make an effort to understand, to repeat accurately, to adapt, and to use newly understood language in conversation and in written composition. Effort is required at every moment and must be maintained over a long period of time. Games help and encourage many learners to sustain their interest and work [23].

**3.4 DISSCUSSION**

There seems to been agreement that the educational model is changing. This is quite clear if one were to look on the reality of education and compare it to the education ten years back in time. The fundamental principles, however, are the same both in present and past time. Slavin mentions the task, reward, and authority structure which I believe is, and has been, the same for a long time [18]. What have changed the most are the tools and above all, because of Internet, the World Wide Web, and computers. It is now possible to collaborate from different location and at different times. Students can broadcast solutions to problems, knowledge, and tutorials. This, let us call it, user created knowledge base is the richest student literature there is. Aside from this, the digitalized education provides a great platform for new tools such as digital games.

Distance education is be coming more and more popular. Not only for distance learning, but also for home education. In order for web-based distance education to surpass, traditional face-to-face education, the spectra of tools has to be broadened. Games are therefore important.

The blended education model seems to be the most likely next model. This is because when something evolves, It is commonly keeping the best parts from the last generation and replaces the not so good parts with new parts, for the new generation. This is simply that the new generation of education will maintain the most effective part from the traditional education, such as questions and answers, and laboratory practical. The new generation would also compliment the traditional scheduled lectures, the student’s literature, and repetitive study. Instead the students attend class whenever he or she pleases. Lectures could be viewed at any time, and literature could be a shared digital document. The study could involve playing games, chatting with other students, reading and writing, watching video lectures, etc.

Games, which are the main subject of this article, have, as mentioned, great educational potential. Not only because they are very engaging, but also because they are very forgiving, because they offer a safe virtual environment. Furthermore, games offer a level of competition which traditional education tools do not. These three reasons along with the fact that the interest for games found at higher and higher age’s are, in my opinion, more than enough to give games a permanent place in the world of education. The list of the advantages of games in education is long, but the most important aspect is the “learning by playing” part. I believe that the most effective form of learning occur while playing or doing. By actually trying one is more likely to learn more than by just watching or reading. This is the games strongest advantage, by being able to provide virtual realities, safe environments for experiments and practice. Providing a never ending supply of virtual material that could be having as it do in reality.

There are not only positive aspects of games in education, there are of course also some draw backs. For instance does the new educational model requires a higher level of discipline in the students. Traditional education is clearly suffering from the problem of engaging students and the more freedom a student gets, the more responsibility will lie on the student. This could lead to students forgetting to study or not making time for it. Even though games are engaging, all games are not for everyone. It is the same thing as with movies or music. Everybody have personal tastes, which makes it hard to develop games for the general audience.

Games and education have a lot in common. For instance do most games, as most education, require some sort of introducing phase. Once this introduction is passed, the player or students, is exposed to a more serious task. The more the player plays, or the student study, the better they get, and the more advanced the task they can solve. They finally pass a test to finish the game or the course.

On the other hand, games and education have differences as well. Games are associated with leisure and entertainment, while studying could be associated with boring tasks and work. Games are played voluntarily at the time of the players choice. Studies often take place at scheduled time and are (aside from university studies) mandatory.

One thing that we can establish is that games would contribute, and already are contributing to some extent, to learning and education. Most authors seems to agree that games would mostly contribute in terms of motivation. In contrast, I believe that games would contribute the most as reality simulators. Having players / students practice and experiment in real life like environments, but without the risk of running out of precious materials or destroying equipment.

Until the day that educational games are being funded with larger budgets, their real potential will not be shown. In the mean time they will complement education as a fun tool.

**3.5 FUTURE WORK**

There seems to be a shortage of evidence confirming that games contribute to education. Therefore there exists a need for studies testing education with versus education without games. The studies also have to be conducted during along span of time and be tested in a number of sciences. Furthermore, educational games are often either accommodated for children or way too simple, graphically and narrative. It would be very interesting to see a high budget educational game with advanced game technologies.

**CHAPTER–4**

**PRACTICAL WORK**

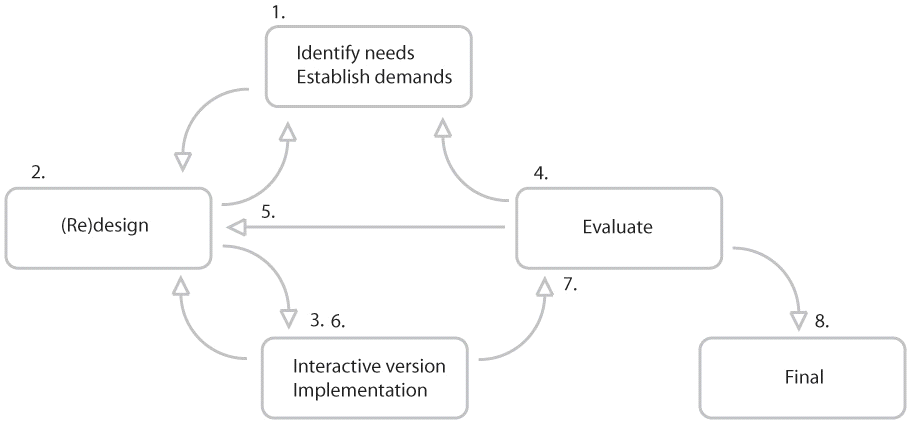
The following chapter contains a thorough walk-through of the practical work done during the masters’ thesis. From the early concept stages to the final most interesting technical solutions.

**4.1 METHOD**

The development process followed the design process to some extent.

The following item list describes the process which is also illustrated in Figure 4.1.

Before this process was initiated the task was specified as completely as possible



**Figure4.1:The design process**

**4.2 DESIGN**

**Requirements**

**Hardware:**

* Personal Computer with internet connection.

**Software:**

* PIXIJS gaming engine
* JavaScript
* HTML
* CSS

**4.3 GAMING DESIGN**

**PIXI.JS gaming engine**

**Pixi.js was started in 2013**

Pixi.js is a free open-source 2D engine used to make animated websites and HTML5 games. It can be used on all modern browsers - on both desktop and mobile. It was launched in February 2013 by Matt Groves and is free under MIT license**.**

Since then it has steadily expanded its features and has been picked up in projects by companies such as like Disney, PBS Kids, BBC, McDonalds and others.

Many games companies in Berlin and Hamburg have also started to use Pixi.js within their tech stack. Version 3 was released in April 2015.

**Pixi.js helps developers create fast, advanced content**

Groves created Pixi.js because he saw a lack of technology required to build content which would be easily accessible on multiple platforms, most notably mobile platforms.

“We created an engine that abstracts all the low-level, convoluted parts to enable anyone (including ourselves!) to create fast, advanced content on the web without needing a computer science degree.” Mat Groves

Pixi.js is used to animate scenes of graphical objects. You can load, move and rotate images, as well as change their color, tint and opacity. You can also add graphic objects to a container and move the container as a unit.

**The biggest advantage of Pixi.js is its speed**

Pixi.js is a devoted rendering engine. It uses Web GL for faster performance, making 2D rendering very fast. If Web GL isn’t supported, the engine falls back to standard HTML Canvas.

**Other advantages include:**

**Multi-platform:** Pixi.js is a great way to create multiplatform (mobile and desktop) experiences

**Power and simplicity:** Pixi.js gives you the power of complicated graphics technology (Web GL) in a simple library, meaning existing filters such as blurs and pixi.js can be created a side custom displacements and halftone effects.

**Good replacement for Adobe Flash**: For developers accustomed to java Script coding concepts Pixi.js will feel quite familiar

**Easy to use and learn:** easy to use API

**Full multi-touch input recognition**

**And other fun stuff:** Sprite sheet support, asset loader and text support

**What does it look like?**



**New features planned**

Matt and his team are running a crowd sourcing to build more features. Some of the things they have planned include:

**Pixi.js Flip**

“The 3D plug in for Pixi.js that lets you rotate all Pixi.js objects in 3D space and render simple models. Great for super slick UIs and simple 3D games/objects.”

**Pixi.js Lights**

“A real time lighting engine. We plan to include a deferred lighting system that will make your content look absolutely stunning!”

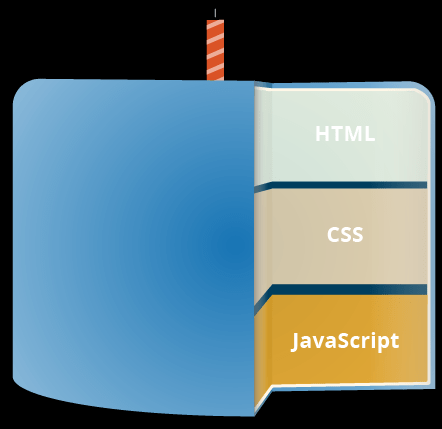
**Pixi.js Flash**

“We are also very keen to create a java Script exporter that will allow you to leverage the power of flash’s wonderful animation tools and publish straight to Pixi.js. This will empower designers to make amazing animations using the java Script IDE that can be easily exported to Pixi.js.”

**JavaScript**

**A high-level definition**

JavaScript is a scripting or programming language that allows you to implement complex features on web pages every time a web page does more than just sit there and display static information for you to look at displaying timely content updates, interactive maps, animated 2D/3D graphics, scrolling video jukeboxes, etc. you can bet that JavaScript is probably involved. It is the third layer of the layer cake of standard web technologies, two of which (HTML and CSS) we have covered in much more detail in other parts of the Learning Area.



**So what can it really do?**

The core client-side JavaScript language consists of some common programming features that allow you to do things like:

Store useful values inside variables in the above example for instance, we ask for a new name to be entered the store that name in a variable called name.

Operations on pieces of text (known as "strings" in programming). In the above example we take the string "Player 1: " and join it to the name variable to create the complete text label, e.g. "Player 1: Chris".

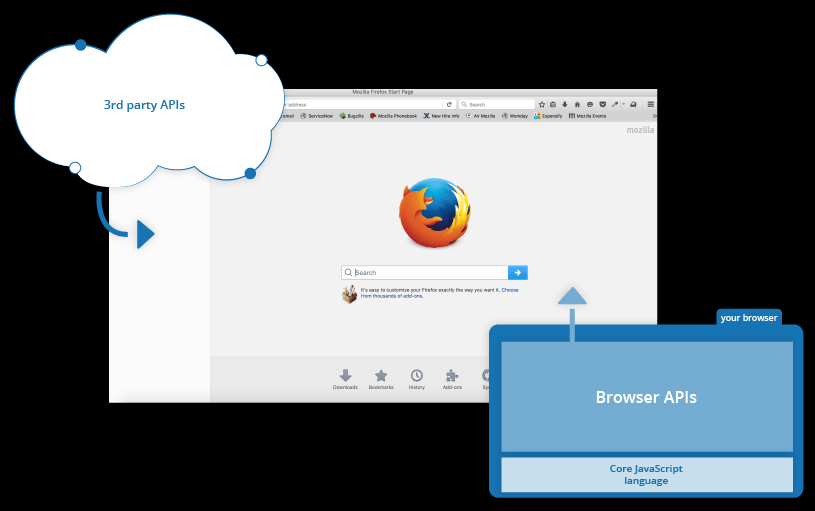
Running code in response to certain events occurring on a web page? We used a click event in our example above to detect when the label is clicked and then run the code that updates the text label.

And much more!

What is even more exciting however is the functionality built on top of the client-side JavaScript language? So-called Application Programming Interfaces (APIs) provide you with extra superpowers to use in your JavaScript code.

APIs are ready-made sets of code building blocks that allow a developer to implement programs that would otherwise be hard or impossible to implement. They do the same thing for programming that ready-made furniture kits do for home building it is much easier to take ready-cut panels and screw them together to make a bookshelf than it is to work out the design yourself, go and find the correct wood, cut all the panels to the right size and shape, find the correct-sized screws, and then put them together to make a bookshelf.

They generally fall into two categories.



Browser APIs are built into your web browser, and are able to expose data from the surrounding computer environment, or do useful complex things. For example:

The DOM (Document Object Model) API allows you to manipulate HTML and CSS, creating, removing and changing HTML, dynamically applying new styles to your page, etc. Every time you see a popup window appear on a page, or some new content displayed (as we saw above in our simple demo) for example, that's the DOM in action.

The Geo location API retrieves geographical information. This is how Google Maps is able to find your location and plot it on a map.

The Canvas and Web GL APIs allow you to create animated 2D and 3D graphics. People are doing some amazing things using these web technologies — see Chrome Experiments and web gl samples.

Audio and Video APIs like HTML Media Element and Web RTC allow you to do really interesting things with multimedia, such as play audio and video right in a web page, or grab video from your web camera and display it on someone else's computer (try our simple Snapshot demo to get the idea).

Note: Many of the above demos won't work in an older browser when experimenting; it's a good idea to use a modern browser like Firefox, Chrome, Edge or Opera to run your code in. You will need to consider cross browser testing in more detail when you get closer to delivering production code (i.e. real code that real customers will use).

Third party APIs are not built into the browser by default, and you generally have to grab their code and information from somewhere on the Web. For example:

The Twitter API allows you to do things like displaying your latest tweets on your website.

The Google Maps API and Open Street Map API allows you to embed custom maps into your website, and other such functionality.

Note: These APIs are advanced, and we'll not be covering any of these in this module. You can find out much more about these in our Client-side web APIs module.

There's a lot more available, too! However, don't get over excited just yet. You won't be able to build the next Facebook, Google Maps, and Instagram after studying JavaScript for 24 hours there are a lot of basics to cover first. And that's why you're here let's move on!

**What is JavaScript doing on your page?**

Here we'll actually start looking at some code, and while doing so, explore what actually happens when you run some JavaScript in your page.

Let's briefly recap the story of what happens when you load a web page in a browser (first talked about in our How CSS works article). When you load a web page in your browser, you are running your code (the HTML, CSS, and JavaScript) inside an execution environment (the browser tab). This is like a factory that takes in raw materials (the code) and outputs a product (the web page).

A very common use of JavaScript is to dynamically modify HTML and CSS to update a user interface, via the Document Object Model API (as mentioned above). Note that the code in your web documents is generally loaded and executed in the order it appears on the page. Errors may occur if JavaScript is loaded and run before the HTML and CSS that it is intended to modify. You will learn ways around this later in the article, in the Script loading strategies section.

**Browser security**

Each browser tab has its own separate bucket for running code in (these buckets are called "execution environments" in technical terms) — this means that in most cases the code in each tab is run completely separately, and the code in one tab cannot directly affect the code in another tab or on another website. This is a good security measure if this were not the case, then pirates could start writing code to steal information from other websites, and other such bad things.

Note: There are ways to send code and data between different websites/tabs in a safe manner, but these are advanced techniques that we won't cover in this course.

**JavaScript running order**

When the browser encounters a block of JavaScript, it generally runs it in order, from top to bottom. This means that you need to be careful what order you put things in. For example, let's return to the block of JavaScript we saw in our first example:

constpara = document.querySelector('p');

para.addEventListener('click', updateName);

functionupdateName() {

const name = prompt('Enter a new name');

para.textContent = `Player 1: ${name}`;

}

**Copy to Clipboard**

Here we are selecting a text paragraph (line 1), then attaching an event listener to it (line 3) so that when the paragraph is clicked, the update Name () code block (lines 5–8) is run. The update Name () code block (these types of reusable code blocks are called "functions") asks the user for a new name, and then inserts that name into the paragraph to update the display.

If you swapped the order of the first two lines of code, it would no longer work instead, you'd get an error returned in the browser developer console Type Error: Para is undefined. This means that the Para object does not exist yet, so we can't add an event listener to it.

Note: This is a very common error you need to be careful that the objects referenced in your code exist before you try to do stuff to them.

Interpreted versus compiled code:

You might hear the terms interpreted and compiled in the context of programming. In interpreted languages, the code is run from top to bottom and the result of running the code is immediately returned. You don't have to transform the code into a different form before the browser runs it. The code is received in its programmer-friendly text form and processed directly from that.

Compiled languages on the other hand are transformed (compiled) into another form before they are run by the computer. For example, C/C++ are compiled into machine code that is then run by the computer. The program is executed from a binary format, which was generated from the original program source code.

JavaScript is a lightweight interpreted programming language. The web browser receives the JavaScript code in its original text form and runs the script from that. From a technical standpoint, most modern JavaScript interpreters actually use a technique called just-in-time compiling to improve performance; the JavaScript source code gets compiled into a faster, binary format while the script is being used, so that it can be run as quickly as possible. However, JavaScript is still considered an interpreted language, since the compilation is handled at run time, rather than ahead of time.

There are advantages to both types of language, but we won't discuss them right now.

**Server-side versus client-side code**

You might also hear the terms server-side and client-side code, especially in the context of web development. Client-side code is code that is run on the user's computer when a web page is viewed, the page's client-side code is downloaded, then run and displayed by the browser. In this module we are explicitly talking about client-side JavaScript.

Server-side code on the other hand is run on the server; its results are downloaded and displayed in the browser. Examples of popular server-side web languages include PHP, Python, Ruby, and ASP.NET and... JavaScript! JavaScript can also be used as a server-side language, for example in the popular Node.js environment you can find out more about server-side JavaScript in our Dynamic Websites Server-side programming topic.

**Dynamic versus static code**

The word dynamic is used to describe both client-side JavaScript, and server-side language it refers to the ability to update the display of a web page/app to show different things in different circumstances, generating new content as required. Server-side code dynamically generates new content on the server, e.g. pulling data from a database, whereas client-side JavaScript dynamically generates new content inside the browser on the client, e.g. creating a new HTML table, filling it with data requested from the server, then displaying the table in a web page shown to the user. The meaning is slightly different in the two contexts, but related, and both approaches (server-side and client-side) usually work together.

A web page with no dynamically updating content is referred to as static it just shows the same content all the time.

Note: You can see this version on Get Hub as apply-javascript-internal.html (see it live too).

**Inline JavaScript handlers**

Note that sometimes you'll come across bits of actual JavaScript code living inside HTML. It might look something like this:

FunctioncreateParagraph() {

constpara = document.createElement('p');

para.TextContent = 'You clicked the button!';

document.body.AppendChild(para);

}

<button onclick="createParagraph()">Click me!</button>

You can try this version of our demo below.

This demo has exactly the same functionality as in the previous two sections, except that the <button> element includes an inline onclick handler to make the function run when the button is pressed.

Please don't do this, however. It is bad practice to pollute your HTML with JavaScript, and it is inefficient you have to include the onclick="createParagraph()" attribute on every button you want the JavaScript to apply to.

Using add Event Listener instead

Instead of including JavaScript in your HTML, use a pure JavaScript construct. The querySelectorAll() function allows you to select all the buttons on a page. You can then loop through the buttons, assigning a handler for each using addEventListener(). The code for this is shown below:

const buttons = document.QuerySelectorAll('button');

for (const button of buttons) {

button.addEventListener('click', createParagraph);

}

This might be a bit longer than the one click attribute, but it will work for all buttons no matter how many are on the page, nor how many are added or removed. The JavaScript does not need to be changed.

Note: Try editing your version of apply-javascript.html and add a few more buttons into the file. When you reload, you should find that all of the buttons when clicked will create a paragraph. Net, huh?

**Script loading strategies**

There are a number of issues involved with getting scripts to load at the right time. Nothing is as simple as it seems! A common problem is that all HTML on a page is loaded in the order in which it appears. If you are using JavaScript to manipulate elements on the page (or more accurately, the Document Object Model), your code won't work if the JavaScript is loaded and parsed before the HTML you are trying to do something to.

In the above code examples, in the internal and external examples the JavaScript is loaded and run in the head of the document, before the HTML body is parsed. This could cause an error, so we've used some constructs to get around it.

This is an event listener, which listens for the browser's DOM Content Loaded event, which signifies that the HTML body is completely loaded and parsed. The JavaScript inside this block will not run until after that event is fired, therefore the error is avoided (you'll learn about events later in the course).

In the external example, we use a more modern JavaScript feature to solve the problem, the defer attribute, which tells the browser to continue downloading the HTML content once the In this case both the script and the HTML will load simultaneously and the code will work.

Note: In the external case, we did not need to use the DOM Content Loaded event because the defer attribute solved the problem for us. We didn't use the defer solution for the internal JavaScript example because defer only works for external scripts.

An old-fashioned solution to this problem used to be to put your script element right at the bottom of the body (e.g. just before the </body> tag), so that it would load after all the HTML has been parsed. The problem with this solution is that loading/parsing of the script is completely blocked until the HTML DOM has been loaded. On larger sites with lots of JavaScript, this can cause a major performance issue, slowing down your site.

**A sync and defer**

There are actually two modern features we can use to bypass the problem of the blocking script async and defer (which we saw above). Let's look at the difference between these two.

Scripts loaded using the sync attributes will download the script without blocking the page while the script is being fetched. However, once the download is complete, the script will execute, which blocks the page from rendering. You get no guarantee that scripts will run in any specific order. It is best to use sync when the scripts in the page run independently from each other and depend on no other script on the page.

Scripts loaded with the defer attribute will load in the order they appear on the page. They won't run until the page content has all loaded, which is useful if your scripts depend on the DOM being in place (e.g. they modify one or more elements on the page).

Here is a visual representation of the different script loading methods and what that means for your page:

**Copy to Clipboard**

You can't rely on the order the scripts will load in. jquery.js may load before or after script2.js and script3.js and if this is the case, any functions in those scripts depending on j query will produce an error because j query will not be defined at the time the script runs.

Sync should be used when you have a bunch of background scripts to load in, and you just want to get them in place as soon as possible. For example, maybe you have some game data files to load, which will be needed when the game actually begins, but for now you just want to get on with showing the game intro, titles, and lobby, without them being blocked by script loading.

Scripts loaded using the defer attribute (see below) will run in the order they appear in the page and execute them as soon as the script and content are downloaded:

<script defer src="js/vendor/jquery.js"></script>

<script defer src="js/script2.js"></script>

<script defer src="js/script3.js"></script>

Copy to Clipboard

In the second example, we can be sure that jquery.js will load before script2.js and script3.js and that script2.js will load before script3.js. They won't run until the page content has all loaded, which is useful if your scripts depend on the DOM being in place (e.g. they modify one of more elements on the page).

**To summarize:**

A sync and defer both instruct the browser to download the script(s) in a separate thread, while the rest of the page (the DOM, etc.) is downloading, so the page loading is not blocked during the fetch process.

Scripts with an async attribute will execute as soon as the download is complete. This blocks the page and does not guarantee any specific execution order.

Scripts with a defer attribute will load in the order they are in and will only execute once everything has finished loading.

If your scripts should be run immediately and they don't have any dependencies, then use sync.

If your scripts need to wait for parsing and depend on other scripts and/or the DOM being in place, load them using defer and put their corresponding <script> elements in the order you want the browser to execute them.

**HTML** is the markup language that we use to structure and give meaning to our web content, for example defining paragraphs, headings, and data tables, or embedding images and videos in the page.

**CSS** is a language of style rules that we use to apply styling to our HTML content, for example setting background colors and fonts, and laying out our content in multiple columns.

**METHODOLOGY**

**The Flow Chart for whole game Flow and Features**

**Main Page**

**Select mode**

**Turbo mode Free mode**

**Points = 0**

**Time = 0:0:15**

**Points = 0**

**Query string, Type answer no time restrictions**

**Query string, Type answer in given time**

**Delete letter if selected**

**wrong letter**

**Delete letter if pressed**

**wrong letter**

**Exit if want to change mode**

**Exit if want to change mode**

**Next**

**Question**

**If**

**Ans=crt**

**if t==0:0:0 Ans = Correct Yes NO**

**Show win**

**Popup**

**Update**

**Points**

**No update**

**Points**

**Try**

**Again**

**Show win**

**Popup**

**Next question**

**Next question**

**FLOW DIAGRAM**

Button Creation

Main

Preload

Initializer

Index.html

Initializer.js

Preload.js

Main.js

Button

Creation

Points Table

Time Table

Delete button

To create empty space for answer

To create shapes for all letters

**HOW PROCESS WILL GO?**

**IIndex.html**

**Buttons Creator**

**Initializer.js**

**Main.js**

**Preload.js**

**Main page**

**Main Page**



* The main page contains mode of selection as Illustrated below.



**Turbo Mode**

* In order to get different perspectives, opinions, and ideas about the word quiz .Turbo Mode good in several ways. It is fun, ideas are developed very quickly, and a lot of progress is made. Furthermore it allows to quickly test several ideas for the system and for the participants to think of the system as a whole before designing each part. Turbo Mode is also a process that moves quickly between radically different designed as.
* This mode adds some difficulties in answering a question.
* In turbo mode user can see the timer .Basically the time for each question will be fifteen seconds.
* Since user wants to answer in specified time, the game gives some good and challenging experience after answered correctly.
* The game will never wait for user's correct answer for a question. Instead it will move to the next question after each 15 seconds.



* If a user not abled to perform well in this turbo mode , he/she can practice in free mode in which the time is not present.
* To switch game mode, user need to go for main menu by Clicking Exit Button.

**Free mode**

* Unlike turbo mode, Free mode has no timer .
* This game mode was implemeted for the purpose of practicing tough words for at beginning stager.
* In free mode User can see all the elements which is present in turbo mode , except time.
* Here is a delete button , by which user can delete a letter if typed wrongly.
* same background has maintained for same experince of both modes.



* Since the absence of timer, there is no time restriction to answer questions.
* Thus the beginners can develop their skill by this Free Mode.
* If user feels that he/she eligible for Turbo Mode, he/she can go for Turbo mode in which the timer is present.
* To Switch Free mode to Turbo Mode, user needs to go for main menu by clicking Exit Button.

**4.4 QUERY STRING**

* Query string is nothing; it is a question which is selected from Students subject.
* The questions were selected from student's ongoing subjects.
* This is the reason why the game will be popular among the students.
* Since answering multiple questions multiple times player will never forget words after good result in skill estimation.
* User can see every question at the top of the screen and if he/she gets good result in Skill estimation part then they will be certified by the team of Words World App.

**Jumbled Letters and Empty Answer place**

* At the center of the screen user can see the Jumbled letters in which the answer letters and some random letters.
* There is totally 30 letters in jumbled format.



* At the bottom of Screen, there is an empty cell which is place for input answer from user.
* These letters were placed in interactive rectangle shapes. So that user can touch any letter using cursor then the letter automatically fills the Empty answer input place.
* Another way to get input from user is key board. We have also implemented the interaction with keyboard.
* User can select letters either keyboard or jumbled letters so this game feels very cool.

**Delete Button**

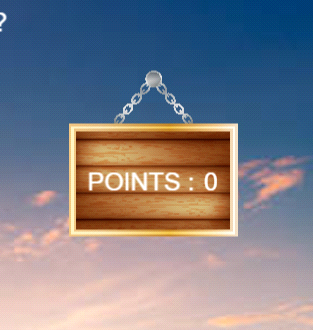
* There is many Chance to select wrong letter while playing with large letters
* The Delete concept is implemented for the purpose of remove letter from a cell if answered wrongly.
* We enabled user to delete letter either by using the delete graphics which in game screen or by simply pressing backspace.

**Exit Button**

* As we discussed earlier in Game Mode (free mode or Turbo mode) , user can reselect the game mode using the Exit button.
* There will be graphic to exit from current game mode. If once exit they can select intended mode of game.
* Thus both players (beginners and well performed players) can interact with this game in easy way. We know that the concentration is the key way for the treasure house of knowledge, thus this game will improve students concentration on their studies.

**Points Table**

* Points table developed to know the user that for how many questions they answered correctly.



* The points table will be placed at the right top corner of game screen.
* If user answered correctly the it increased by one value. otherwise it will never change.
* Initially the value will be zero.

**After answered correctly**

* After answered correctly each question player can see a win popup win animations.
* This can make some addition good feel of winning a question.
* And also the point’s table value increased, thus they can see for how many questions answered correctly.

**Estimate skill**

* Skill estimation is process of calculating all correct answers with respect to total number of questions.
* There is 50 questions will be shown for a game play.
* If any payer got 35 out of 50 he/she will be eligible for Turbo mode. Otherwise it’s better to continue with Free Mode.
* After Completion of 50 questions, user can see the result of his/her skill and directed to main page in which game modes present.

**CHAPTER–5**

**CONCLUSION**

* 1. **RESTRICTION**

The greatest restriction is that I could not follow the project all the way through because the whole *Words World App* project was delayed and the time for this thesis ran out. Another restriction is the multi-player part of the game. It is supposed to support one player. In the final solution the game is supporting two players to play against each other.

**5.2 LIMITATIONS**

The questions of the game have limited space, in terms of pixels, and therefore the questions cannot be longer than a certain number of characters in any language. External assets such as images and sound files must be formatted to use as little disc space as possible, because if the files are to large it will result in that the game will appear to freeze for a short period of time when it is actually loading the images.

This could be worked around by creating a loading screen, but as of now, there is none. When two players play each other there might occur problem due to differences in band-width and response times. The game has not been completely tested and may therefore contain bugs. The testing could not be completed because only some of the content in the databases were finished by the time this period was over.

**5.3 THOUGHTS**

Another point of interest is that I conducted the thesis alone. This, I found, had both advantages and draw backs. It demanded a lot of discipline and responsibility since I had to manage every part of the project by myself. From this I learned a lot. On the other hand I experienced some hard ship when I got stuck with a problem or needed input. I certainly realized the importance of having someone to widen my perspective and helping to carry out the work load.

It is hard to decide whether the project was a success or not due to the delay of the whole *Words World App* project. It was initially supposed to launch in the fall , but by the time I am writing this (March 2022) it has still not been launched. This made it impossible for me to test the game in its´ real environment and in its´ real context. As a learning experience however I will think of it as a success. I was over all been a good experience.

**5.4 FUTURE WORK**

The highest priority is on testing. Before publishing the game it should be thoroughly tested so it can be made sure that it works for all languages and that the multiplayer part works as well. The multiplayer part should also be further developed to support more than two players. The possibilities of developing more advanced games for this type of education should also be taken in to csonsideration.

**CHAPTER–6**

**SOURCE CODE**

Code to make a successful remote method call via java script. The “onResult” and “onFault” methods are manually configured.

responder=newResponder(onResult,onFault);connection.connect(gateway);

connection.call("DataGrid.getText", responder, sender.toString());

Thealgorithmforcorrectinganswers.

for(vari:Number = 0; i<s.length;i++)

{

if(s.toLowerCase().charCodeAt(i) !=

QItems.getItemAt(qNumber-1).answer.toLowerCase().charCodeAt(i))

{

returnfalse;

}

}

returntrue;

**INDEX**

<!doctype html>

<html>

<head>

<scriptsrc="https://pixijs.download/release/pixi.min.js"></script>

<div id="bgdiv"></div>

<style>

body {

background-color :rgb(7, 7, 7);

background-image :url('image/newpreload.png');

background-repeat:no-repeat;

background-size:cover;

top:"-200%";

}

</style>

</head>

<body>

<script type="text/javascript" src="Initializer.js"></script>

<script type="text/javascript" src="Images\_bytecoded.js"></script>

<script type="text/javascript" src="ButtonsCreator.js"></script>

<script type="text/javascript" src="main.js"></script>

<script type="text/javascript" src="preload.js"></script>

<!--<script type="text/javascript" src="main2.js"></script> -->

<script>

</script>

</body>

</html>

**INITIALIZER**

vartimestart = 1;

varLetterArr = new Array();

varLettercontainerArr = [];

varrandtemp;

varshuffeledArr = [];

varpositionX = 0;

varQntext

varpositionY = 0;

varAnswerWord;

varqnCount = 0

vartimeCount =16

varTimePosX =45

varTimePosY =100

varpointval =0;

varObj = new Object()

varClickedLetter = "";

varShapesContainerArr = [];

varShuffleArr= [];

varshapeAll = new PIXI.Graphics();

shapeAll.beginFill(0, 0.5).drawRect(425, 68, 500, 370);

shapeAll.y = 60;

var myshape1

varAnsShapeArr = []

var map2={}

varInteractiveflag = true

var count = 0

varUserAnsarr = []

varUserAnsText = ""

var Main;

varPointsPosX=925

varPointsPosY=-50

var ClearButtt,ClearButt,ClearBut,PointBench,PointBenchh,PointBenchhh;

varGametypeval = 0

//..............code to justify the letter of time ( helped to keedtimetext as 2 digit)....................

String.prototype.leftJustify = function (length, char) {

var fill = [];

while (fill.length + this.length< length) {

fill[fill.length] = char;

}

returnfill.join("") + this;

};

// varPreloadArr = ["image/wordimg2.png","image/wordimg.png"]

LetterArr = ["A","B","C","D","E","F","G","H","I","J","K","L","M","N","O","P","Q","R","S","T","U","V","W","X","Y","Z"];

var app = new PIXI.Application({

width: 1350,

height: 600 \* 1.25,

transparent: true,

});

document.body.appendChild(app.view);

varAnsArr=[

"Sun microsystem",

"Kernel",

"Linus Torvalds",

"System calls",

"Linux",

"Child shell",

"Login prompt",

"SA",

"Linux",

"Linus",

"Shell script",

"Linux",

"Minix",

"ANSI",

"Path name",

"Shell",

"Dead space",

"Dual",

"Input mode",

"Pipe",

"BAH shell",

"Verification",

"Beep sound",

"Variables",

"One line",

"Clear",

"Intel",

"Tree structure",

"Noclobber",

"Unmount",

"Group users",

"A user",

"Linux",

"Linux",

"Unix",

"Editors",

"Suitable linux",

"Linus Torvalds",

"Bar graphis",

"Mounting",

"Register",

"Structure variable",

"Repeat until statement",

"Dennis Ritchie",

"Bubble sort",

"Binary Tree",

"Cubic probing",

"Stack",

]

varQnsArr = [

"which of the following introduced the flavor of unix named solaris ?",

"what is the core of the linux operating system?",

"who is the founder of linux kernel?",

"what approach does an application use to communicate with the kernel?",

"which has excellent support for older hardware?",

"what is the other name of new shell?",

"where do the users arrive after entering exit in linux system?",

"who can change the password of any user?",

"which system provides excellent security features?",

"who was very much attracted by elegance and effectiveness of UNIX?",

"which text file contains linux commands?",

"which operating system was developed by the contribution of many people throughout the world?",

"which version of UNIX was handled by the students of the university of Helsinki?",

"which standard of unix was followed by linux?",

"which is called as full name of directory?",

"which program combines linux commands to solve the given problems?",

"what is the space between end of the line and end of the screen is called as?",

"which vi editor,makes the keyboard to play a role?",

"In what mode of editing ,the keyboard behaves as a normal typewriter with the exception?",

"which command takes data from one command to another command?",

"which is linus default shell?",

"what is the process does not allow any unauthorized person to access any of ourdirectories or file?",

"which indicate you to find the mode you are working?",

"who are placeholders to store values?",

"what type of editing is possible by Ed editor?",

"which command clears the screen?",

"what based of PC's ,linux was specifically designed?",

"where did the file system can be used to unambiguously identified and referenced?",

"which feature is used to prevent overwriting an existing file the redirection operation?",

"when we want replace a file system we have to present file system?",

"In Linux ,if all the members of the group share their files,they are called as ",

"who can change his/her password with the password command?",

"which shell provides many of the tools found in C language?",

"which was widely distributed over the Internet?",

"which was the operating system was created mainly by efforts of Ken Thomson?",

"what is mainly used for creating,deleteing and editing the files?",

"which version can run on any machine available now?",

"who developed an effective PC version of Unix for Minix users?",

"which of the following is not considered on of the main categories of computer graphics?",

"How to connect a file system on a storage device and our main directory tree by?",

"which of the storage class may help in faster execution?",

"when accessing a structure ,the identifier to the left of the operator is the name ?",

"which of the following is not a valid data type in C language?",

"who developed C language?",

"which of the following sorting algorithm is the slowest?",

"which of the following is not a linear data structure?",

"which of the following is not an open addressing technique to resolve collision?",

"which of the following data structure is more appropriate for implementing quick sort iteratively?",

]

**PRELOAD**

app.stage.pivot.x = app.screen.width / 2;

app.stage.pivot.y = app.screen.height / 2;

app.stage.x = app.screen.width / 2;

app.stage.y = app.screen.height / 2;

varImg = GTypeBg.src;

gametypebg1 = new PIXI.BaseTexture(Img);

gametypebg = new PIXI.Texture(gametypebg1);

gametype1 = new PIXI.Sprite.from(gametypebg);

gametype1.anchor.set(0.5, 0.5);

gametype1.x = (app.stage.width/2)+450 ;

gametype1.y = 100 +200;

gametype1.scale.x = 0.3;

gametype1.scale.y = 0.3;

FreeTxt = new PIXI.Text("Free Mode", { fill: "white" ,fontSize:"30px" });

FreeTxt.anchor.set(0.5, 0.5);

FreeTxt.x = gametype1.x;

FreeTxt.y = gametype1.y;

GameTypeCon1 = newPIXI.Container()

GameTypeCon1.addChild(gametype1,FreeTxt)

gametype2 = new PIXI.Sprite.from(gametypebg);

gametype2.anchor.set(0.5, 0.5);

gametype2.x = (app.stage.width/2)+850 ;

gametype2.y = 100+200 ;

gametype2.scale.x = 0.3;

gametype2.scale.y = 0.3;

TurboTxt = new PIXI.Text("Turbo Mode", { fill: "white", fontSize:"30px" });

TurboTxt.anchor.set(0.5, 0.5);

TurboTxt.x = gametype2.x;

TurboTxt.y = gametype2.y;

GameTypeCon2 = newPIXI.Container()

GameTypeCon2.addChild(gametype2,TurboTxt)

app.stage.addChild(GameTypeCon1,GameTypeCon2)

GameTypeCon1.interactive = true

GameTypeCon1.on("pointerup",function(){

Gametypeval = 1

setTimeout(() => {

main();

}, 1000);

})

GameTypeCon2.interactive = true

GameTypeCon2.on("pointerup",function(){

Gametypeval = 2

setTimeout(() => {//..............code to justify the letter of time ( helped to keedtimetext as 2 digit)...................

main();

}, 1000);

})

**MAIN**

function main() {

// ............creating a stage to keep all things ......................................................

app.stage.removeChild(GameTypeCon1)

app.stage.removeChild(GameTypeCon2)

app.renderer.view.style.opacity = "0";

app.renderer.view.style.transform = " scale(0.5)translate(0%,0.9%)";

app.renderer.view.style.transformOrigin = " centercenter ";

setTimeout(() => {

app.renderer.view.style.transform = " scale(1) translate(0%,0.6%)"; // implemented transioneffet

app.renderer.view.style.transformOrigin = " centercenter ";

app.renderer.view.style.transition = " 1s";

app.renderer.view.style.opacity = "1";

}, 100);

app.renderer.view.style.backgroundImage = "url("+"image/wordimg2.jpg"+")"//"url("+wordimage2.src+")";

app.renderer.view.style.backgroundSize = "cover";

//.............................................stage creation ENDED.......................................

app.stage.addChild(shapeAll);

// ..................creating points table ...............................................................

// varanimatationArr = ["sprite Images/Images/win1.jpg","sprite Images/Images/win2.jpg","sprite Images/Images/win3.jpg"]

// var textures = []

// for (i=0;i<animatationArr.length;i++){

// x = (new PIXI.BaseTexture(animatationArr[i]))

// textures.push(new PIXI.Texture(x))

// }

// let myanim = new PIXI.AnimatedSprite(textures)

// myanim.animationSpeed = 0.01

// myanim.play()

// app.stage.addChild(myanim)

varImg = Pointsbg.src;

PointBenchhh = new PIXI.BaseTexture(Img);

PointBenchh = new PIXI.Texture(PointBenchhh);

PointBench = new PIXI.Sprite.from(PointBenchh);

PointBench.anchor.set(0.5, 0.5);

PointBench.x = 210 + PointsPosX;

PointBench.y = 260 + PointsPosY;

PointBench.scale.x = 0.06;

PointBench.scale.y = 0.06;

Pointstext = new PIXI.Text("POINTS :", { fill: "white",fontSize:"25px" });

Pointstext.anchor.set(0.5, 0.5);

Pointstext.x = 190 + PointsPosX;

Pointstext.y = 270 + PointsPosY;

PointstextVal = new PIXI.Text("0", { fill: "white",fontSize:"25px" });

PointstextVal.anchor.set(0.5, 0.5);

PointstextVal.x = 260 + PointsPosX;

PointstextVal.y = 270 + PointsPosY;

app.stage.addChild(PointBench, Pointstext, PointstextVal);

// ................. creating table for time.............................................................

// .................. creating delete button..............................................................

ClearButtt = new PIXI.BaseTexture("image/clearbut.png");

ClearButt = new PIXI.Texture(ClearButtt);

ClearBut = new PIXI.Sprite.from(ClearButt);

ClearBut.x = 1030;

ClearBut.y = 320;

ClearBut.scale.x = 0.4;

ClearBut.scale.y = 0.4;

ClearBut.interactive = true;

app.stage.addChild(ClearBut);

ClearBut.on("pointerdown", function () {

Main.deleteFun()

});

//conrolling deleted letters

//....................creating function that used to create all letters for the game .....................

this.CreateShuffledLetters = function (qntxt, ans) {

// ......................qn text creation...............................................................

Qntext = new PIXI.Text(qnCount + 1 + ". " + qntxt, { fill: "white" });

Qntext.anchor.set(0.5, 0.5);

Qntext.x = 1350 / 2;

Qntext.y = 60;

app.stage.addChild(Qntext);

//......................................................................................................

AnswerWord = String(ans).toUpperCase();

ShuffleArr = AnswerWord.split("");

//........... inserting random letters and ans letters into an array.....................................

while (true) {

if (ShuffleArr.length == 30) break;

else {

randtemp = Math.floor(Math.random() \* LetterArr.length - 1);

if (randtemp> 0) {

ShuffleArr.push(LetterArr[randtemp]);

}

}

}

// ...............shuffling all the letters..............................................................

for (var i = ShuffleArr.length - 1; i > 0; i--) {

var j = Math.floor(Math.random() \* (i + 1));

var temp = ShuffleArr[i];

ShuffleArr[i] = ShuffleArr[j];

ShuffleArr[j] = temp;

}

//.............. creating shuffled letter shapes to see in stage ........................................

var map = {}

for (i = 0; i < 30; i++) {

if (ShuffleArr[i] == " ") {

ShuffleArr.splice(i, 1);

randtemp = Math.floor(Math.random() \* LetterArr.length - 1);

if (randtemp> 0) {

ShuffleArr.push(LetterArr[randtemp]);

i--;

}

} else {

map["name\_" + i] = LetterAndShape(

ShuffleArr[i],

450 + positionX \* 80,

90 + positionY \* 70,

1

); // using class that create all shapes............................................................

console.log('hello')

positionX += 1

if (positionX == 6) {

positionX = 0;

positionY += 1;

}

}

}

//.................. creating empty shapes to let the user to fill .....................................

varAnsAllshape = new PIXI.Container();

AnsAllshape.x = shapeAll.x;

AnsAllshape.pivot.x = shapeAll.x / 2;

for (k = 0; k <= AnswerWord.length - 1; k++) {

if (AnswerWord[k] == " ") {

continue;

} else {

map2["shape\_" + k] = LetterAndShape(

"",

1350 / 2 - (AnswerWord.length / 2) \* 70 + k \* 70,

450,

0

);

AnsShapeArr.push(map2["shape\_" + k]);

}

}

};

//.........................................................................................................

document.addEventListener('keypress', (event) => {

varKeyText = event.key;

Main.letterListener(KeyText.toUpperCase())

if (KeyText === "Backspace")

console.log("KeyText",KeyText)

}, false);

document.onkeydown = function(event) {

var key = event.keyCode || event.charCode;

if( key == 8 || key == 46 )

Main.deleteFun()

};

this.deleteFun = function(){

if (count > 0 && count <= AnswerWord.length) {

UserAnsarr.pop();

count -= 1;

Interactiveflag = false;

if (AnswerWord[count] == " ") {

UserAnsarr.pop();

count -= 1;

}

Main.AnswerPlace("");

}

}

this.letterListener = function(txt){

if(count>=0&&count<=AnswerWord.length-1){

Interactiveflag = true

if(AnswerWord[count]==" ")

{

UserAnsarr.push(AnswerWord[count])

count+=1

}

UserAnsarr.push(txt)

Main.AnswerPlace(txt)

count+=1

}

}

// ..... this function will be activated when user clicks any letter and it will add text to empty space...

this.AnswerPlace = function (text) {

map2["shape\_" + count] = LetterAndShape(

text,

1350 / 2 - (AnswerWord.length / 2) \* 70 + count \* 70,

450,

2

);

if (count == AnswerWord.length - 1) {

UserAnsarr.forEach((element) => {

UserAnsText = UserAnsText + String(element);

});

//..................to verify if user ans and inbuilt ans same.........................................

if (UserAnsText == AnswerWord) {

pointval += 1;

// PointText(pointval); //increments pointer text value

PointstextVal.text = "" + pointval;

setTimeout(() => {

ResetAll();

}, 500);

} else {

UserAnsText = "";

}

}

};

// ...................to call next qn and reset all the value at initial state............................

functionResetAll() {

qnCount++;

ShuffleArr = [];

app.stage.removeChild(Qntext);

shapeAll.children = [];

UserAnsText = "";

UserAnsarr = [];

positionX = 0;

positionY = 0;

timeCount = 16;

count = 0;

Main.CreateShuffledLetters(QnsArr[qnCount], AnsArr[qnCount]);

}

if(Gametypeval == 2){

timeBench = new PIXI.Sprite.from(PointBenchh);

timeBench.anchor.set(0.5, 0.5);

timeBench.x = 200 + TimePosX;

timeBench.y = 105 + TimePosY;

timeBench.scale.x = 0.06;

timeBench.scale.y = 0.06;

Timetext = new PIXI.Text("YOUR TIME", { fill: "white",fontSize:"25px" });

Timetext.anchor.set(0.5, 0.5);

Timetext.x = 193 + TimePosX;

Timetext.y = 100+ TimePosY;

TimetextVal = new PIXI.Text("0", { fill: "white",fontSize:"25px" });

TimetextVal.anchor.set(0.5, 0.5);

TimetextVal.x = 193 + TimePosX;

TimetextVal.y = 130 + TimePosY;

app.stage.addChild(timeBench, Timetext, TimetextVal);

//...............Interval to create time duration..........................................................

setInterval(() => {

timeCount -= 1;

timeCount1 = String(timeCount).leftJustify(2, "0");

TimetextVal.text = "00:00:" + timeCount1;

if (timeCount == 0) {

setTimeout(() => {

ResetAll();

}, 900);

}

}, 1000);

}

// if(gameT)

Main = this;

Main.CreateShuffledLetters(QnsArr[qnCount], AnsArr[qnCount]);

}

//...........................................................................................................

**BIOLIOGRAPHY**

* W. Bates. *Technology, E-learning and Distance Education*. Routledge,2 edition,2005.
* Daniel Burgos, Colin Tatters all, and Rob Khoper. Re-purposing existing generic

Games and simulations for e-learning.(1),2006.

* Li-Jie Chang, Jie-Chi Yang, Fu-Yun Yu, and Tak-Wai Chan. Development and evaluation of multiple competitive activities in a synchronous quiz game system.
* Guy Cook. Language plays, language learning. *ELT Journal*, 51/3, July 1997.

.

* Daniel Fallman. Design-oriented human-computer interaction.2003.
* Karen Holtz blatt and Hugh Beyer. Making costumer centered design work for teams.

*Communications of the ACM*,36(10),1993.

* William Horton. *E-learning by Design*. Pfeiffer,2006.
* John Kirriemuir and Ceangal Angela McFarlane. Literature review in games and learning. Technical report, University of Bristol, 2004.
* David Klaila. Game-based e-learning gets real.
* Barbara Means, Yukie Toyama, Robert Murphy, Marianne Bakia, and KarlaJones. Evaluation of evidence-based practices in online learning. 2009.
* MarcPesky.Thedigitalgame-basedlearningrevolution.2001.
* MarcPesky.Themotivationofgameplay.*OntheHorizon*,10(1),2002.
* RobertE. Slavin. Cooperative learning. *Review of Educational Research*, 50(2):315–342,1980.
* Kurt Squire. Game-based learning: Present and future state of the field. *E Learning Consortium*, February 2005.
* Kurt Squire and Henry Jenkins. Harnessing the power of games in education. *Insight*, 3(5), 2003.