Honours Final Report

"An Evaluation of the Suitability of HCI in Educational Software"

By

Aladdin

BHCG_1

2008xxxxx

Project Supervisor: Dr Richard Foley
Second Marker: June McCrae

Except where explicitly stated, all work in this document is my own.

Signed: ______________________________ Date: _______________

Abstract

Research has indicated that educational software and e-learning technologies receive a mixed opinion within the industry. Many researchers believe that e-learning technologies have the potential to completely replace traditional teaching methods, where others believe e-learning technologies are not advanced or clever enough to address an individual's unique requirements effectively.

HCI (Human Computer Interaction) and multimedia plays a large role in how effectively educational software communicates with the user, and many are quick to dismiss the idea of implementing educational software within the classroom due to the high criticism of HCI within educational software. However, some companies are making large sums of money through successful deployment of their educational software and it is evident that many schools have already implemented educational software within their curriculum. It has been claimed that educational software has the potential to support a large variety of learning styles through good use of HCI and popular learning models such as VARK have indicated that the majority of people possess more than one learning style.

This project identifies several mathematical educational software packages at Key Stage 2 level which include different aspects of multimedia and learning models within each package and evaluates the effectiveness of HCI using an analytical evaluation method which created guidelines specific to these packages, influenced by other design principles and heuristics. This experiment will help provide an indication to which elements of learning models and multimedia best influence HCI, and aim to create a set of design principles specific to Key Stage 2 educational Mathematics software, based on the outcome of the results.

Acknowledgements

I would like to thank my Project Supervisor, Dr Richard Foley, for investing hours of his time at meetings, where I received a lot of support and advice; without this advice I would not have developed such a rigorous attitude towards this project.

Contents

Table of Figures

1. Introduction

1.1 Project Background

Reports, surveys and studies (Lapin et al, 2007; Plimmer, 2006; Mueller et all, 2007) have suggested that e-learning technologies can vastly improve the way in which one learns. Whilst earlier studies such as (Tscheligi, 2005) have highlighted various limitations and impracticalities of e-learning material, it is undeniably apparent that technology has drastically evolved in the last ten years alone.

This “leap” has been particularly demonstrated by (Kopf, 2007), revealing that the e-learning market in the US was worth $17.5 USD in 2007 and has been projected at a colossal $56.2 billion USD by 2010. Many large companies such as IBM are already investing large amounts of money into 3-Dimensional virtual worlds for business and education as the success of existing social learning technologies has already become apparent (Shen, 2009). As a result of such rapid growth, it is apparent that educational e-learning has the potential to feature more frequently alongside “uni-modal”, traditional learning to enhance and enlighten the way students intake information.

Traditional education, that is the “chalk and talk”, submissive approach to learning, has dominated the 20th Century and still features very preferably within education today. However, as technology has emerged very rapidly in the past few decades, findings by reports suggest that technology could become a primary role within education. The research of (Rashty, 1999) suggests that e-learning can possess all the traits of traditional learning with the added advantage of not being limited to short timescales as one could work around the clock which may entice higher motivation and lead to an extended degree of involvement in the learning process.

In addition to this, (Allison, 2007) believes that using a higher degree of technology in everyday teaching can be more cost effective, where less time is spent getting the message across to students. Of course, limitations, past and present have played a large role in the influence of educational technology within the classroom. Nonetheless, it still is widely believed that learning-based technology is only one of many factors which contribute to making learning more efficient.

HCI (Human Computer Interaction) reflects the development and design within educational software. HCI and its cognitive psychology with the user can greatly influence the way in which a user interacts with the software. Elements of multimedia within effective HCI of educational software is thought to support learning through frequent activity in multiple ways, where students have very little time to do so manually and can be used to enhance problem solving scenarios such as those within Mathematics, where abstract concepts can be animated or simulated in simpler ways (Roschelle, 2001).

While there are a wide variety of factors which are suggested to impact the ways in which one learns such as gender, experiential differences, cognitive differences environmental surroundings etc., (Inkpen, 1997) research has shown that an individual’s learning style is an important factor. The effectiveness of HCI within educational software is very much dependant on which learning styles an individual possesses, from many commonly established learning styles (Anon, 2009). There are many learning models which incorporate these learning styles into unique categories – among these are David Kolb’s Model (Kolb, 2009) and Fleming’s VARK system (Fleming et al, 2009). David Kolb’s Model has often been used to relate learning styles along with e-learning software, and has been refined and modernised over the last three decades (Arthurs et al, 2007). However, research has indicated that this model must be used in conjunction with others to fully capture online learning styles.

Other findings, supported by (Leite, 2009) make a rigorous attempt at validating Fleming's VARK system. VARK's system has four learning categories (Visual, Aural, Read/write, Kinaesthetic), where learners ultimately fit into one, several or all of these categories. A kinaesthetic learner (e.g. learning through a gaming environment) may also fit into the category of a visual learner (such as picture stills, diagrams etc), thus the model can be very flexible in assigning learning styles. Leite Et Al analysed a sample of 14,211 US individuals who took part in VARK's online questionnaire and results showed that 35.8% of the examinees were classified by the algorithm to benefit from all four of the established learning styles as opposed to 29.7% to just have one. This suggests that many contain multiple learning styles which could be best suited to multimodal environments which often exploit several learning styles at once (Clark, 2008). Learning styles can be easily evaluated through VARK's website, which contains a questionnaire with 16 questions, with 4 testlets each (more than one option can be selected) in order to conduct an extensive analysis of one's learning style (Fleming et al, 2009).

Bongers (Bongers, 2009) believes design heuristics play a large part in establishing what works well for e-learning and students who ultimately use the software and are a valuable source for determining what works well. It is commonly known that subjects such as Mathematics are not perceived as interesting to children, (Cole, 2007) discusses an experiment, where Mathematics software was piloted across 14 classrooms in Texas and at the end of the course, over 75% of the children involved in the process who previously didn't find the subject interesting were now finding it a fun exercise. Additionally, (Sedighian, 1997) describes an experiment involving 15 children using an exemplar piece of Mathematics software using a game incorporating several multimedia techniques, such as interactive animations and sound, lead to an average improvement of 50% in test scores.

With it therefore being apparent that different factors which contribute to the success of HCI for children, it is important to consider these elements (Moroz-Lapin, 2008). Lapin discusses some of these features and believes the first and foremost of successful Human Computer Interaction lies within elements such as bright colours, self-evident icons and a simple, clutter-free layout are said to contribute to a successful engaging interactive experience for the child. HCI experts such as Jakob Nielsen and Gehardt-Powals have created design principles which further elaborate on these aforementioned elements (Nielsen et. Al, 1990; Gerhardt-Powals et. Al, 2006).

It will therefore be beneficial for this project to look into methods of experimenting with different aspects of HCI and multimedia within educational software in a domain such as Mathematics, particularly with children, as there are further needs to discover what works well and what doesn't.

1.2 Project Outline & Research Question

The main aim of this project is to gain an insight into the impact of which HCI has on educational software within the domain of Mathematics at Key Stage 2 level. Through the use of 4 different educational e-learning packages/environments, an experimental HCI evaluation took place, involving deep analysis and scrutiny on several aspects of these packages, including the software's use of multimedia and underlying learning models amongst a host of other important aspects.

As stressed within the background of this report, HCI within software is often under-evaluated, making the selection process difficult to conduct. In order for this project to carry out a well established evaluation, all HCI evaluation methods and techniques must be reviewed. Guidelines for establishing which educational packages were included within the evaluation have been concluded from literature research in order to follow a well-justified systematic process. A set of HCI Guidelines for evaluating these packages were then created and applied to the evaluation. This will allow for these packages to be reviewed efficiently and aims to derive a set of results which will gain insight into the appropriateness of both the Mathematics educational software and the applicability of the HCI evaluation guidelines.

This project will therefore adopt an experimental approach and use the following Research Question to achieve its goal:

"How suitable is the HCI within typical Mathematics Educational Software, in regards to the use of multimedia and underlying learning models, for Key Stage 2 Primary School students?"

1.2.1 Project Objectives

The objectives of the project are as follows:

Objective 1:

- Conduct Literature Review

Purpose

In order to fulfil the Research Question, one of the first milestones in this project is to conduct a literature review. The literature review has three main objectives: discuss the appropriateness of HCI within educational software, explore the uses of multimedia within mathematics educational software generally aimed at children and to discuss underlying learning models which exist within educational software. Examples, problem areas, benefits and ways to address solutions are explored for each of these objectives and overall conclusions/findings are drawn from the literature review which will influence the methods of the project.

Objective 2:

- Develop a set of guidelines for selecting educational software

Purpose

There is an endless list of educational software commercially available, making it unrealistic and unfeasible to evaluate them all. In order to make the experiment as efficient and varied as possible, it is beneficial to set out some guidelines/principles for selecting these packages which can be applied in a systematic manner. The nature of this process was inspired from Nielsen (Nielsen et. Al, 1990), who gathered his requirements for what attributes must exist within the software for his evaluation, which became the basis of a selection process. Many packages will be reviewed through this process, and as a result, the best 4 software packages adhering to these guidelines will be selected.

Objective 3:

- Select appropriate HCI evaluation methods for carrying out the experiment

Purpose

There are various ways of conducting an HCI evaluation and as such these evaluation methods need to be reviewed in order to select a format which will best answer the Research Question. It is important to consider the benefits and drawbacks of each evaluation method in order to measure its applicability against the project's requirements.

Objective 4:

- Construct a set of HCI evaluation guidelines to be used within the experiment on the chosen software

Purpose

There are many ways in which the effectiveness of HCI can be measured (e.g. Nielsen's Heuristics, Gerhardt Powals' cognitive engineering principles, which are discussed within the Literature Review) and in to fulfil all areas of the Research Question, several different principles will need to be reviewed and adapted. In order to analyse the results and outcome of the evaluation, a scoring process within the HCI guidelines is also necessary.

Objective 5:

- Evaluate these packages with the guidelines and scoring system

Purpose

The experiment will then involve evaluating each of these educational software packages. Every aspect of the software's HCI will be measured against the aforementioned HCI guidelines and in addition to this, a "score" will be decided for each guideline in relation to the package. This will allow for in depth analysis at the results stage.

Objective 6:

- Investigate and analyse results

Purpose

Results from the experiment can then be analysed. These will identify areas in each of the packages where HCI is ineffective or successful. It will also indicate the ways in which the underlying learning model has influenced HCI behaviour in Mathematics software. These results can be analysed, the problem areas which the guidelines revealed can be discussed and evaluated, the overall usefulness of the learning environments can be "ranked" and possible solutions can be proposed.

Objective 7:

- Develop guidelines/principles for selecting and designing Key Stage 2 Mathematics Software

Purpose

After the results have been discussed and an overall conclusion has been reached, overall principles can be created and discussed which were subsequent from research and analysis. These will be specifically aimed at guiding educational bodies or developers to what are the best and most effective features of Key Stage 2 Mathematics Software.

1.3 Report Structure

1.3.1 Literature Review

The Literature Review, in Section 2 of this report explores three specific objectives resulting from initial research. These are:

- The appropriateness of HCI within educational software
- The uses of multimedia within educational software for children
- Underlying learning models within educational software

These objectives form a discussion on the issues involved in poor HCI engagement within software and look evaluation methods which have been used to improve HCI and the impact they currently have.

The impact of which multimedia has with children is then discussed and issues and benefits are explored.

Finally, a discussion on why underlying learning models within educational software is important to the HCI engagement of educational software leads to proposed models which best aid in the HCI evaluation process.

The outcomes are then concluded and how they will be brought through to the evaluation stage of the experiment is discussed.

1.3.2 Methods

The methods section outlines the how the experiment will be carried out, using evaluation techniques and adaptations subsequent from the literature research. This section explains why Analytical methods were chosen over Empirical Methods then describes and selects techniques from analytical methods.

This section also explains how the educational software search process is carried out and explains remaining tasks between this report and the final report.

1.3.3 Results

The results section of the report is presented and analysed in section 4. This section deals with the results of the guidelines against the four software packages which were chosen. Results are presented and grouped in charts and the problems or success which were derived from the guidelines against the software are discussed.

1.3.4 Discussions and Conclusions

Section 5 concludes and summarises the report. It provides a general summary of the main issues which were encountered, as well as successful aspects within the software which were discovered during the evaluation. Motivated from this general overview, findings from the Literature Review are put in context with the results of the evaluation in order to create guidelines/principles which can be used for developing or selecting Key Stage 2 Mathematics software. Lastly, a discussion takes place, considering further areas of research, where the scope of this project could be altered to form the basis of a whole new project for other developers in order to extract more information in this specific domain.

2. Literature Review

2.1 The appropriateness of HCI within educational software

In their search for teaching large numbers of students, educational establishments have acknowledged new ways of utilizing information and rapidly growing technology. With potential to execute learning material through many modes of multimedia, educational software requires increased HCI engagement and as such, it's applicability within today's society is very much a hot topic (Hartley, 2010). Human-Computer Interaction is a common practice worldwide – a staple of modern day's digital society. A study revealing one in three people within Europe and one in two people within the United States owning a Personal Computer within their home suggests regular HCI is evident from a range of backgrounds – from the inexperienced to the experts (Inkpen, 1997).

Poorly designed interfaces have often lead to a host of complex problems. There have been incidents within the industry where this led to disaster. A severe case of where this has occurred was the classic example of the Three-Mile Island accident, where in Pennsylvania in March 1979, there were critical failures with a Nuclear Power Plant control system which were caused due to ambiguous design within the power plant's User Interface (NRC, 2010). As this lead to uncontrolled amounts of toxic waste being released into the area, it almost lead to serious health risks to a dense population within the immediate area. In this case, it was therefore evident that poor HCI design practices had been followed which has carried on to the functioning of the software.

Ofcourse, where educational software is concerned, poor practices in evaluating HCI and interface design is very unlikely to compromise someone's life, but in a comparative nature, poor HCI and design considerations can result in the software lacking "user friendly" features, failing to "connect" with the end user and this can cause huge negative impacts to large companies who produce large commercial educational software packages (Jones et al, 1999).

Preece et al (Preece, 1999) discusses how poor HCI within a software package can lead to the user feeling inadequate, where they aren't likely to feel inclined to use similar types of software in the future and are also unlikely the software to peers. Thus, in the light of educational software, where technology clearly has the potential to offer educational establishments a stepping stone from traditional teaching methods into technology-based information – HCI design principles must be understood to engage with the target audience (Jones et al, 1999**).**

As technology has emerged within the past 20 years, educational software has surfaced and transformed within the market and the need for HCI design principles has been developed by many authors. In-depth studies of HCI (Dix et al, 2004) has discussed and reviewed various HCI design rules which are commonly known as "HCI guidelines". These guidelines have been created by publishers who are experienced in key fields such as Interaction Design and through their research, they have uncovered a well-established set of "rules and recommendations" to be followed which is encouraged which is encouraged to either lead to a well-designed, user-friendly system or evaluating for these properties which cater to the exact needs of the end-user. The most used Heuristic evaluation method for such purposes are Nielsen's Heuristics, which, according to researchers (Davies et al, 1996), have contributed to well-developed commercial software, boasting successful HCI with its users, such as Maths Whizz educational software for children.

An example of one of Nielsen's famous 10 Heuristics states "Match between system and real world", where the terminology used should be based on the target user's language for any one task. For example, in the case of educational software aimed at children in the domain of Mathematics, language should be simple and clear of "elaborate" wording and number schemes, in order to comply with the child's natural language. (Bongers, 2009).

This context can be applied to most of Nielsen's Heuristics, such as:

- **User control and freedom** – This would be of particular importance within a children's educational software environment. Children are particularly more motivated when they are the initiator of actions as they feel more in control of outcomes and enjoy the process of feeling involved. This would be achieved by allowing the child to initiate an action as opposed to just responding to one.
- **Error prevention** – Error prevention in the context of Mathematics software would involve having control over the way in which the user interacts with the environment. The child should not feel discouraged to explore and entering mathematical answers, for instance, should be controlled by some verification and validation so that the user is aware when they are entering foreign characters.

Another well known expert within this field is Gerhardt-Powals, who has established and refined his own set of guidelines in the same systematic process that Nielsen used; Gerhardt-Powals' Cognitive Engineering Principles (Gerhardt-Powals et. Al, 2006). Like Nielsen's Heuristics, Gerhardt-Powals' principles can also be put in context of children's educational software. For example:

- **Automate unwanted workload** – Any extra in-take of information which is not necessary to the child when using the educational software should be avoided. This can be achieved using cognitive resources, and in the case of mathematics, features such as on-screen calculators may be used to aid the child in completing calculations which would otherwise be worked out externally from the software (Moroz-Lapin, 2008).
- **Include in the displays only that information needed by the user at a given time** – The user should not be distracted from their task. In order to avoid distraction, elements should be condensed where necessary and the screen should contain no information which is external to the task at hand. Furthermore, although cognitive resources would be useful, this principle would suggest that if these cognitive resources are not needed for that current task then they should not be available to select in that particular session.

It was discussed at a general HCI conference (Interact**,** 2009), how developers who engineer their software against HCI principles often fail to adapt pre-set guidelines (such as Nielsen's). One of the main faults uncovered from this was developers were only adapting from a single set of HCI design principles and it was considered how adapting methods from several HCI design principles could better address this issue and help developers create software "connect to their end users" more.

In particular, it was discussed how in-depth evaluation methods could be adapted with Heuristics to focus a developer's aim more tightly. Global Usability Analysis, which "breaks down" all observable elements of HCI organised into categories with questions, such as "are all elements of the user-interface consistent throughout the software?" e.g. file menu always available in the same area of every screen. Thus, adapting principles from the likes of heuristics and selectively extracting questions from Usability Analysis has the potential to address common HCI issues which still exist today.

2.2 The uses of multimedia within educational software for children

Research suggests that there are still issues concerning children's experience with multimedia technology which are still under-supported in the Primary School stage. Lack of awareness is still very common among academic staff who may not be fully able to understand or appreciate the educational ability that current multimedia technologies can provide to Primary School children (increasing awareness, 2010). Similarly, research has discovered that most regions of Europe, students who transfer from Primary School to Secondary School have experienced some difficulties, where different styles of teaching, namely multimedia-based are explored to a greater extent (Multimedia, 2010).

Reports have explored how a multimedia-based learning environment can increase the quality and efficiency of a child's learning ability (Gupta et al, 2009). These reports discuss how better learning can be achieved through visual stimulation which occurs with the use of colours, shapes, moving images etc. The use of multimedia with an educational message/information will allow the child to be captivated by the use of these techniques. Furthermore, the child is likely to enjoy the "fun side" of this medium of education, the becoming unaware that they are learning at the same time, yet still extracting the relevant educational information (Green et al, 1996).

Captivation through multimedia already has many examples of benefits over traditional methodologies. Technology such as the "DS Lite" has spawned world-wide multimedia-based games around the goals of quitting smoking and losing weight and through exploring these methods through captivating environments, people have used this software to distract them from their daily bad habits. (Hartley, 2010).

Therefore it is evidential that through research and popular examples of multi-media being applied into everyday life, that multimedia can usefully be deployed for the purposes of education and has notable potential within educational software at children and that the industry could benefit from further addressing these issues which still exist within schools where children have insufficient exposure to multimedia-based education material.

2.3 Underlying learning models within educational software

Educational software aims to deploy its material through a variety of techniques, involving the use of multimedia. In the traditional teaching industry, this is known as a "pedagogy", however, when discussing its involvement in educational software, it is often referred to as a Learning Model (Lilley et al, 2004). Learning models are intended to be designed around particular end user's style of learning. For example, studies have investigated that the adults studying within academic institution's preferred style of learning is usually through the use of text and picture based information, where children, such as those within Primary School's tend to favour game and animation-based learning materials (Zaphiris & Ang, 2007).

As discussed within the introduction of this report, there are many popular processes (e.g. David Kolb's Experiential Model and Fleming's VARK model) which exist and categorise these learning models using different descriptive terms and techniques. It is often overlooked that learning models can directly moderate the outcome of Human Computer Interaction behaviour, and when evaluating HCI within software with an educational purpose, it has been indicated that it is beneficial to include methods for evaluating underlying learning models (Fleming, 2009).

Extensive research has discussed the benefit of the aforementioned VARK (Visual, Audio, Read/Write, Kinaesthetic). A researcher adapted the VARK model which took it's basic concept and adapted it to describe the four types of educational software which exists (Fleming, 2009). Similar to this is a representation of the idea in Figure 1 and 2, below:

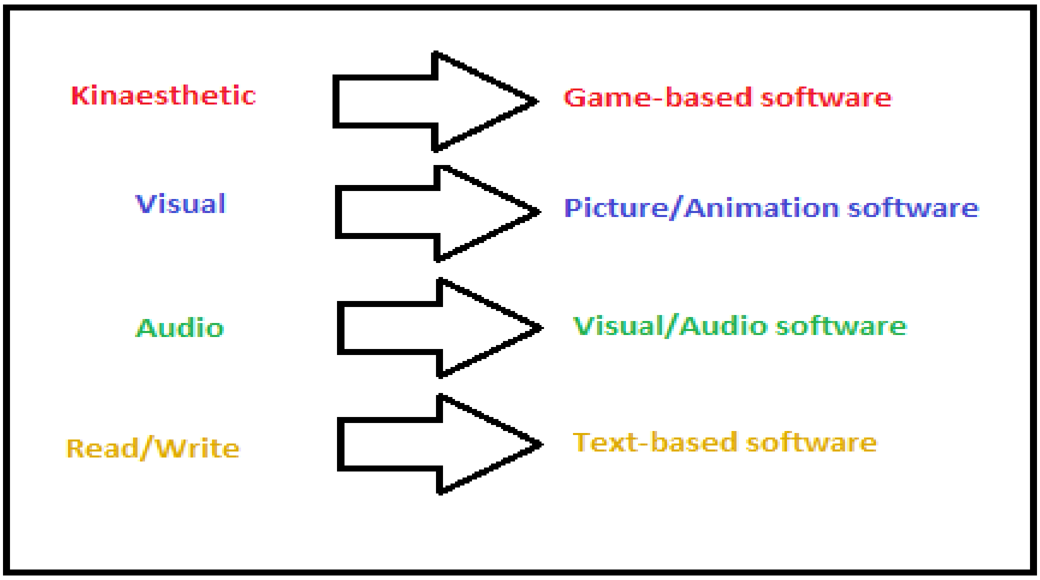


Figure 1: Adapted VARK model representing educational software package types

Therefore, when considering VARK'S elements of learning styles in terms of educational environments, they can be summarised as following:

- **Game-based software:** This type of educational software will teach through the aid of gaming. The gaming exercises would deploy learning material through the high use of interactivity, thus where the "kinaesthetic" element would exist. By using this type of educational environment, strong emphasis is put on "learning by doing".
- **Picture/Animation software:** This environment communicates its learning material through still pictures or animations. These animations may also have an element of interactive. Anything representing visual elements which are not of a video nature or solely classed as a game would fall into this category. Text may also accompany Picture/Animation software, although environments which mainly promote textual material would fall into another category.
- **Video/Audio:** This environment would simply deploy learning exercises through within video sessions followed by audio commentary. This does not support activities kinaesthetically, however stimulates the user sound and vision.
- **Text-based software:** The remaining element promotes a read/write nature. This isn't likely to support any interaction and would purely rely on it it's narrative to teach the user. If a "read-write" nature is fully desired, then the user would likely have to externally take notes off-screen whilst using the learning material

2.4 Literature Review summary

Through this literature research, it is evident that there are still issues within the industry regarding HCI engagement with educational software, however, it is proven to have potential if HCI evaluation methods are selected and adapted properly. As a result, this report will look into all the possible HCI evaluation methods and techniques in order to effectively evaluate Mathematics educational software aimed at Key stage 2 students.

Additionally, through researching the uses of multimedia within educational software for children, this research has demonstrated how multimedia has had a positive impact on the way children learn and absorb information. This literature objective helped re-iterate the importance of this project's Research Question and could provide quality, useful information to other researchers who are particularly interested in developing or researching this particular age group and domain. It is hoped that the outcome will produce a set of results which could refine the guidelines specific to this domain and age group.

Finally, through research it has been decided that learning models which influence learning styles within educational packages also influence the behaviour and impact of HCI, therefore it will be incorporated and expanded on in the evaluation, which is explained within the Methods section

3. Methods

This chapter looks at HCI evaluation methods and reviews the different techniques which are available within these methods. This project therefore discusses alternative methods and justifies the use of selected methods and demonstrates how these methods have been adapted to suit the goals of the project. This chapter then goes on to discuss the process of how the experiment will be carried out and describes what has yet to be completed for the remaining months of the project.

3.1 Methodology Review

An HCI evaluation can be conducted in several ways. There are generally two specific ways in which these evaluations can be addressed: Empirical methods, which involves the participation of human subjects and Analytical methods which is a systematic process, often described as "pencil and paper" (Dix et al, 2004). The methodology review will look at the most significant techniques from both of these processes and discusses their suitability in terms of the project.

3.1.1 Empirical Methods

An empirical approach involves "analysing how the user performs when interacting with a system" (Dix et al, 2004**).** An empirical evaluation can be achieved using "Experiment", "Cognitive Process" and "Queries" techniques.

An empirical experiment involves sourcing human participants in order to perform tests to investigate whether a particular hypothesis is correct. This technique can be of benefit to evaluators with intermediate expertise and allows the evaluator to record a large amount of information regarding the hypothesis (Meira, 2004). The downside to this technique, however, is that human subjects must be carefully selected, as it requires familiarity and experience with the interface.

A cognitive process involves the evaluator observing participants using the interface and then records these actions and sequences through notes, audio or video. Popular examples of these techniques are: "Think Aloud" which involves the participant being encouraged to describe whatever they are looking at and whatever they are thinking or feeling when engaging with the application (Green, 1996). "Remote Testing" is where the evaluator indirectly observes the user while they are interacting with the application. Again, these techniques can provide qualitative information as the results are similar to the experience of the participants. However, the environment may not feel natural to the participants and they may feel uncomfortable by the presence of the evaluator or by the process of "being watched" (Meira, 2004).

Queries are a technique which the evaluator uses to gather opinions about the interface of the system. This is commonly achieved through Focus Groups where the evaluator guides a discussion about the application with a group of participants (Green, 1996) or questionnaires where multiple choice answers are presented to the participants. The advantages of Queries is that it can provide a balance of both qualitative and quantitative information, however, a large number of participants are required for this technique to be effective (Meira, 2004).

Therefore, due to one of the main goals of the project involving the evaluation of several packages (4 or 5), empirical evaluation methods wouldn't be best suited on the basis that it would be far too time consuming to evaluate this many packages with human participants effectively. In addition to this, analysis of the packages would require a degree of familiarity, and as these packages have been pre-selected by the evaluator, it would be too venturesome to assume participants of the target age group (Key Stage 2) would have sufficient experience with these particular packages.

3.1.2 Analytical Methods

In comparison with Empirical methods, Analytical methods tend to not be as time consuming. Analytical methods usually don't directly involve the participation of human subjects and many of the popular techniques do not require human subjects at all. This method requires the evaluator to "analyse the design of the software and systematically assess the impact it will have on its particular audience" (Dix et al, 2004). The most common types of techniques which fall into this category are Model-based evaluations and Expert Appraisals, such as Heuristic Evaluations and Usability Analysis.

Model-based evaluations are largely based on previous research related to the software's interface and the user's behaviour with the interface and which habits entail from the interactions. One of the most popular models which is an example of this is "GOMS" (Goals, Operators, Methods and Selection Rules) which was created by Card, Maran and Newell in 1983, which is used to predict the effects of errors on task performance of a user and redesign the interfaces accordingly (Green, 1996).

There are some disadvantages of using GOMS. None of the techniques proposed by GOMS account for user unpredictability, functionality of the system is not considered by this model (only usability) and GOMS is only really useful for systems which are in the process of being developed and allow for adaption. As the project must investigate both functionality and usability of the educational packages, and they are fully developed packages already, this evaluation technique will not be used.

The other category of analytical methods is concerned with establishing guidelines and evaluating software against them is known as an "Expert Appraisal" (Inkpen, 1997). The "Heuristic Evaluation" was created by Nielsen in 1990 which involves the evaluator examining the interface and "grading" it's conformity against well-established guidelines. Heuristic Evaluations can provide some quick and cheap feedback for the evaluator, however, it does require a high level of expertise. Furthermore, Heuristic evaluations are most effective when the same process is followed by several people (Dix et al, 2004**).** Therefore, this technique is somewhat relevant for this project and some of Nielsen's Heuristics will be adapted and used within the evaluation. As this technique will only be partly used, another technique (discussed below) will be adapted and used along with this one in order to create a comprehensive set of HCI evaluation guidelines.

Usability Analysis is almost similar to a Heuristic Evaluation, only it's a more in-depth systematic and recorded process. A Usability Analysis determines how easy it is, through HCI, to find, understand and manipulate information on an application and how well these aspects conform to guidelines and design principles (Dix et al, 2004). Usability, only technically requires one evaluator, as it provides a comprehensive checklist of questions (over 200) directly related to the HCI and interface design of the software, which can be easily adapted to the evaluator's investigatory purpose of their project.

3.2 Nature of experiment

This section of Methodology entails the way in which the educational software packages were sourced and the instruments which will be used to evaluate the packages. Following this is the systematic process which was used to choose the various packages, using selection guidelines as a moderator. Lastly, the aforementioned HCI evaluation techniques are discussed and how they will be applied to the evaluation is demonstrated.

3.2.1 Instruments & resources

The instrument used to carry out the search for educational packages is solely the use of the internet. Using search engines and well established educational websites, the internet is able to adequately offer a large variety of educational software fit for the aims of this project. The process in which the software has been selected is detailed in the next section.

The resources which will be used to carry out the evaluation will be a PC with specifications capable of running all of the software packages/environments to their full ability hence not compromising the quality of the experiment. As the PC belongs to the evaluator, the resource is readily available for the flexibility of the experiment and all software has been reviewed and confirmed as attainable.

Additionally, Microsoft Excel, which is comprehensive spreadsheet software, will be used extensively throughout this project, in order to record the software selection process and all of the results of the software and thus Microsoft Excel will also be useful for dynamically creating charts and graphs to graphically represent the results within section 4 of the report.

3.2.2 Sourcing educational software

The process used to source the educational software involved creating a set of package selection guidelines which would assist in selecting software which is relevant to the particular objectives which are related to the research question. This process was used to source over 20 potential software packages and after inspection, the four packages which best fit the guidelines and purpose of the investigation were selected for involvement for the experiment.

The software selection guidelines work as follows:

- Package selection is primarily sorted by category of underlying learning model. There are four main learning models which have been identified through research:
  - Text-based (the application mainly deploys it's learning material through a text-based environment)
  - Game-based (the application mainly deploys it's learning material through a game-based environment)
  - Picture/animation-based (The application mainly deploys it's learning material through pictures/animations/models)
  - Video/audio-based (the application mainly deploys it's learning material through videos and audio)
- Packages are then searched and sorted into one of the four categories which best describes its primary function. Within each category, several packages are listed and assessed through the guidelines which are:
  - The name of the educational software/environment

- Conditional criteria:
  - Is the software available at Key Stage 2 level?
  - Does the software in question offer a free trial or evaluation copy?
  - Does the sample cover sufficient content? (this is required in order to fulfil all aspects of the HCI guidelines within the evaluation)
- Any notes about the particular software which may be relevant, thus beneficial to record.

Below is Figure 2 which shows the template of the software selection guidelines. Each instance of educational software which has been researched is sorted into one of the four aforementioned categories and matched against the conditional criteria - this is done in Microsoft Excel which allows an easy overview of all the software.

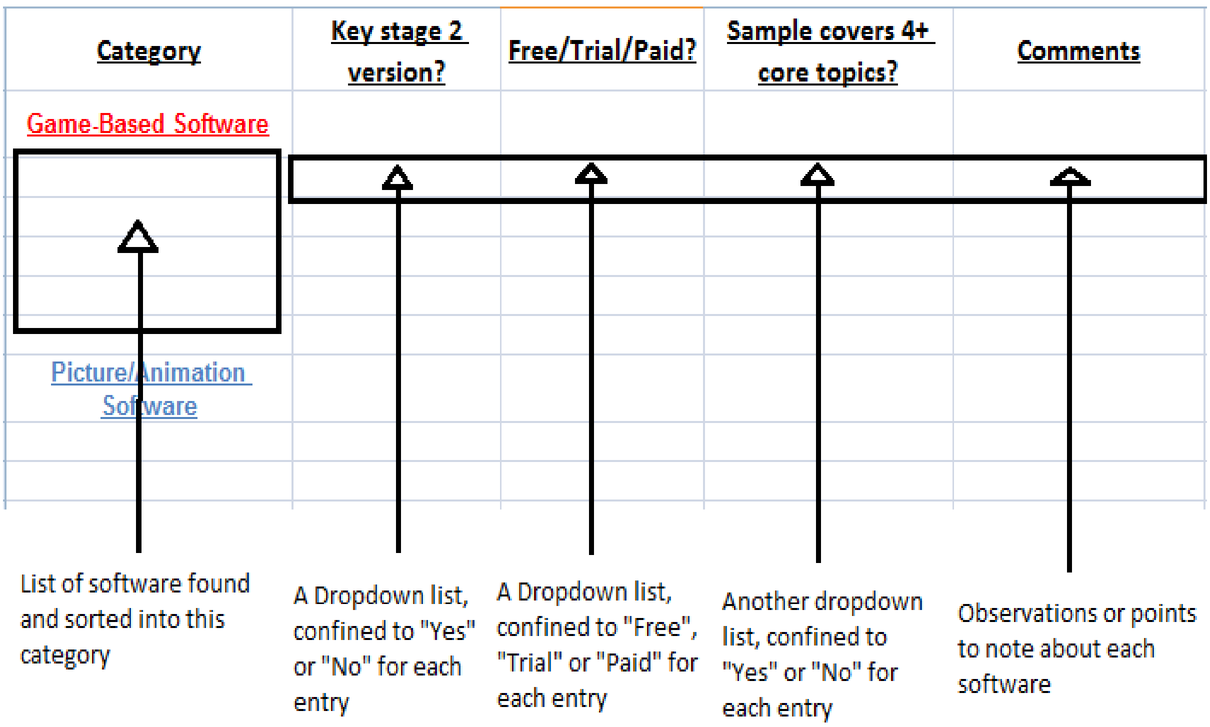


Figure 2: Software Selection Guidelines

The complete outcome of this process and the final chosen software for each category is discussed and evaluated in section 3.3 of this report.

3.2.3 HCI evaluation techniques

As reviewed and discussed in the earlier stages of the Methods section, this report will be conducting an evaluation, adapting guidelines from both Nielsen's Heuristic Evaluation and Gerhardt-Powals' Cognitive Engineering Principles. These guidelines and principles have been reviewed and transformed into in-depth questions which are the basis of the guidelines for this HCI experiment. Four categories have been created for these guidelines, which helps sort these guidelines by relevance and will also be useful for analysing the results of the experiment (Section 4). The structure and nature of the evaluation itself is presented as an adapted usability checklist, using Usability Analysis techniques which consists of a scoring system and in-depth comments for each guideline, all of which is discussed within this section of the report.

Below is Figure 3 which is an overview of the categories which have been created to suit this evaluation and their subsequent guidelines. These are further discussed in sections 3.2.3.1 – 3.2.3.5. However, for the complete process of how these guidelines were derived, refer to Appendix E within section 7 of the report.

HCI Evaluation Categories	Number of Guidelines
Practicality of the software	5
Simplicity of the software	9
Communicating the subject	8
Efficiency of the software	7

Figure 3: HCI evaluation categories

The structure of the evaluation checklist was built in the spreadsheet software Microsoft Excel and is shown in Figure 4 below. A separate sheet exists for each of the four categories/software chosen which helps separate, isolate and record data from the experiment effectively. This will also allow dynamic graphs and charts to be created and presented, which will automatically deal with changes and modifications in any of the results.

This figure shows an example of how a category of guidelines would be evaluated against a piece of software where each score is compared against a scoring system between 0-4 and comments are recorded which further support this score. The score for this software (Actual Score) is calculated and compared against the overall maximum score for the category and a percentage is displayed for each of these categories. After the evaluation is fully complete, a total overall score and percentage is calculated and summarised at the bottom of the sheet.

The scoring system works as follows:

- **Score 0:** The requirements of this guideline are not achieved at all.
- **Score 1:** There were a few cases where this guideline was demonstrated, but mostly it wasn't.
- **Score 2:** The guideline was achieved in some areas but was equally absent in others.
- **Score 3:** The guideline was mainly achieved throughout the software, however, there were a few cases where it wasn't.
- **Score 4:** This guideline is fully demonstrated within the software.

Game-Based Software	Maths Whizz		
Category/Criteria	Max Score	Actual Score	Comments
Practicality of the Software			
1. Does the software provide adequate facilities for the user to address the core concepts of Mathematics?	4	2	Some of the harder games could provide additional tasks before starting the exercise
2. Is the information provided to the user relevant to the task and context?	4	4	Always in context and directly related to task
3. Are the functions provided to the user relevamt to the task and context?	4	4	Always in context and directly related to task
4. Are commands, actions and terminology followed consistently throughout the software?	4	3	Commands, actions and terminology mostly stick to a convention, however, this is not followed in some of the more advanced games
5. Is information presented in natural and logical order for this particular age group?	4	4	All relevant information is attention-grabbing and always followed by animations and sound
Category Total	20	17	
Category Percentage		85.00%	
Simplicity of the Software			
6. Is each screen free of clutter? (not too much information displayed on one screen)	4	3	The screens don't appear to be over-cluttered, in some screens there could be less information/controls

Figure 4: HCI evaluation template

In total, 29 comprehensive guidelines have been created and numbered and these will be discussed along with their categories in the following sections.

3.2.3.1 Practicality of the Software

For successful educational software aimed at children, it must be as practical as possible: It must simply provide all functions and elements that are necessary for Key Stage 2 Mathematics. All information must directly relate to the subject and information should be presented in an order which is naturally perceived by this age group of 7-11. Therefore, the following guidelines (1-5) are as follows for this category:

Guideline 1: Does the software provide adequate facilities for the user to address the core concepts of Mathematics?
Guideline 2: Is the information provided to the user relevant to the task and context?
Guideline 3: Are functions that are provided to the user relevant to the task and context?
Guideline 4: Are commands, actions and terminology followed consistently throughout the software?
Guideline 5: Is information presented in natural and logical order for this particular age group?

3.2.3.2 Simplicity of the Software

This category deals with how effectively the software can deliver its material, whilst remaining as simple as possible, which is required for this age group. This category achieves this by evaluating the placement of functions on the screen, the general usability of the software and the level to which the software provides resources which help make the process of mathematical exercises simpler. Therefore, the following guidelines (6-14) are as follows for this category:

Guideline 6: Is each screen free of clutter? (Not too much information displayed on one screen?)
Guideline 7: Does each screen condense elements which are not central to the task at hand? (Free of unnecessary elements?)
Guideline 8: Are the most commonly used elements and options for a given task visible on each screen (with other options easily accessible)?
Guideline 9: Is the general design of the software "self-evident" (e.g. Would you expect the target age group to be able to navigate and locate functions without prior experience and detailed instructions with the software)?
Guideline 10: Is all "unwanted workload" automated? (e.g. Is it made faster, simpler, easier and more fun where possible?)
Guideline 11: Are the instructions and data "straight to the point" (consolidated and summarised) where expected, with this particular age group in mind?
Guideline 12: Is new information/topics introduced with aids to interpretation, which would be meaningful to such a young age group?
Guideline 13: Does the software reduce the need to remember information, where possible? (Such as progress on calculations, displaying times tables where appropriate, etc)
Guideline 14: With this particular age group in mind, are high level tasks such as calculations and estimations supported by cognitive resources?

3.2.3.3 Communicating the Subject

This category is concerned with how adequately the subject of Mathematics is communicated to its target age group and how well the methods and techniques are used to support this. The guidelines aim to investigate how rich in feedback the software is, how transparent the software is in communicating the sequence of the exercise, if the software contains any features which track and display the user's accomplishments and progress and in which way it introduces examples and new information using metaphors and conventions as well as how appropriate mathematical terminology and symbols are used with this age group. Therefore, the following guidelines (15-22) are as follows for this category:

Guideline 15: Does the software provide appropriate, clear and timely feedback for the user so that they can see what is going on and know the result of their actions?
Guideline 16: Is a structure within the software adopted? Are related items put together and unrelated items set apart?
Guideline 17: Is the user aware of the sequence of the task at hand? For example, are they clearly informed of their progress at all times? (Beginning, middle and end)
Guideline 18: Help and documentation at hand? Are these instructions concise and do they focus properly on the task at hand?
Guideline 19: Does the environment keep track of the user's accomplishments? (Scoreboards, graphs and charts depicting progress over time, etc)
Guideline 20: Does the interface properly communicate between pre and post submission of answers?
Guideline 21: Where appropriate, are common metaphors, concepts and real-world conventions followed and understandable for the target age group?
Guideline 22: Are all mathematical symbols and terminology used limited to this target age group?

3.2.3.4 Efficiency of the Software

This section aims to evaluate how efficient the software is in adhering to the target age group's requirements. The software must continuously aim to challenge the user and thus detect advancement in skill, the software must provide shortcuts for the user more familiar with the environment, and the software must be interactive, exercises allowing manipulation of results. The software must gain the user's trust, so they feel eager to explore all of the functions and the design of the software should be pleasant, colourful, fun and playful in order to best connect with the user. Therefore, the following guidelines (23-29) are as follows for this category:

Guideline 23: Does the software accommodate the user's continuous advancement in knowledge and skill? (E.g. Less help pop-ups once the user begins to re-use common elements within the software)
Guideline 24: Are all alternative shortcuts available to allow the user to access different areas of the software, once they become more experienced with frequent use?
Guideline 25: To which extent does the environment allow the user to feel "in control" of their actions? (E.g. Initiating an action as opposed to responding to one)
Guideline 26: The environment should employ "forgiveness" where possible. Is the user prevented from making serious errors, wherever possible? Is there confirmation in place before the user performs a "destructive action"?
Guideline 27: Error recovery? Are clear and concise error messages displayed when there are disconnects or internal errors? Does it prompt to restart? (Only deduct in the event of an error, where it's apparent there's no error recovery options)
Guideline 28: Does the environment promote "undo" and "redo" actions, where beneficial? The ability to reverse actions encourages the user to explore unfamiliar options
Guideline 29: How well does the interface design relate to the target audience? (E.g. Attractive, bright, positive etc)

3.4 Chosen educational software packages

The outcome of the package selection process and the subsequent choices of educational software for each of the categories is discussed in this section. No software which was released before 2005 was included within the selection process and if the resources well not developed commercial company or experts within the field of education then they were also not included within the selection process.

3.4.1 Game-based choice

As shown in Figure 5 below, there were 5 entries within the game based category. Braintastic and Britannica were not selected due to no trials being available and Braintastic was only available at an expensive price, whilst Britanicca was more outdated than other software within this category, being released in 2006. Further to this, the trial for Kid's Academy was only available as a trial which did not offer enough core topics to satisfy the requirements of the experiment. This left Education City and Maths Whizz. Both were qual candidates and had good commercial value, although the final choice boiled down to Maths Whizz due to access being completely free as opposed to a 21 day trial.

Category	Key stage 2 version?	Free/Trial/Paid ?	Sample covers 4+ core topics?	Comments
Game-Based Software				
BRAINtastic! Maths KS2	Yes	Paid	Yes	An up-to-date commercial piece of software. Although, a free or trial version cannot be found. Expensive to buy (£11.99).
Education City	Yes	Trial	Yes	Up-to-date online learning resource which offers unlimited access for 21 days.
Kid's Academy - KS2	Yes	Trial	No	Several releases of the software would have to be bought to cover sufficient material for this experiment. 2009 release.
Maths Whizz	Yes	Free	Yes	Online learning resource, fully up-to-date and commercially developed. Appeared regularly during software selection process
Britanicca Education Series	Yes	Paid	Yes	No trials or free versions have been sourced. Software is very cheap to purchase although release date is 2006.

Figure 5: Selecting game-based software

Maths Whizz, demonstrated in the figure below, is online software which centred on deploying its teaching material through interactive games. This software rewards users for completing daily exercises with credits which can be spent on various collectible items as a virtual currency within its virtual shop. The child will be awarded with medals and trophies for completing exercises within certain times and correctness and their scores can be compared with their “buddies”. Maths whizz offers the child to personalise their own profile and tries to encourage the child to learn the maths lessons by retaining statistical information about their progress and achievements over time. This is information is exclusive to each account which is initially managed and created to by the child’s parent and the account is linked to their parent’s e-mail address.

Figure 6: Maths Whizz

3.4.2 Picture/Animation-based choice

Four instances of software was found for the picture/animation category, all of which had quite a commercial factor within the industry. Both of the desktop software which was found, Maths 1.0 and Lett's Maths were not chosen due to the other options being less restrictive than these which only offered a paid or trial version. Bitesize Revision and Topmarks were both equally capable of taking part within the experiment and were both up-to-date, but the final choice was Topmarks, as a BBC title was chosen for the Video/Audio category and it was preferred that each of the software was owned by a different source.

Picture/Animation Software	Key stage 2 version?	Free/Trial/Paid ?	Sample covers 4+ core topics?	Comments
Maths 1.0	Yes	Trial	Yes	Downloadable desktop evaluation available. Release date 2007. Not entirely sure how commercial the software is.
Lett's Maths Test Primary	Yes	Paid	Yes	No active evaluation copies available, although cheap to purchase. Most up-to-date version of this is 2006. However, it is a popular commercial title.
Topmarks	Yes	Free	Yes	All topics are fully covered for KS2 and completely free. This software is an online learning resource. Content is up regularly updated.
Bitesize revision	Yes	Free	Yes	Up-to-date and an online learning resource. Also a very commericial title (Created by BBC).

Figure 7: Selecting picture/animation software

Topmarks, displayed in the figure below, is an online learning resource which offers a variety of Key Stage 2 Maths topics and many exercises within each of these, when selected. These exercises are taught through pictures, animations or interactive animations (like the one below) where all the information is displayed.

Figure 8: Topmarks

3.4.3 Video/Audio-based choice

As shown in Figure 9 below, there were five considerations for the Video/Audio category. Themathsfactor.com would have been the perfect candidate for this category, with a strong commercial appeal of Carol Vordeman and it appearing to have a lot of money and time invested into its development. It would have been interesting to evaluate this software within the experiment, however, due to the requirement of an expensive monthly subscription, this was not possible. Europress only offered a five minute evaluation of the software (and was quite dated) and Learning Ladder seemed quite dated with very little commercial development to it. BBC Learning Zone was the best pick for the category, with a well known commercial company behind it's development and being up-to-date it seemed more appropriate than teachers.tv which was suited to teachers as well as students.

Video/Audio Software	Key stage 2 version?	Free/Trial/Paid ?	Sample covers 4+ core topics?	Comments
Europress	Yes	Trial	Yes	This software only allows a downloadable 5 minute free trial which would not be sufficient for the requirements of this experiement. 2006 release date
themathsfactor.com	Yes	Paid	Yes	Online resource with interactive video based maths exercises, using the popular presence of Carol Vordeman. Very commercial and up-to-date, however requires monthly subscription to access.
Learning Ladder KS2	Yes	Trial	Yes	Covers sufficient content with an evaluation copy, however it is very dated and not the most commercial.
BBC Learning Zone	Yes	Free	Yes	Online resource which is completely free and is very commercial (BBC) and up-to-date.
teachers.tv	Yes	Free	Yes	Up-to-date learning resource with some commercial presence. This resource can be used for both teachers and KS2 students, which may be a bit contradictory for the purpose of the experiment.

Figure 9: Selecting video/audio software

BBC Learning Zone, shown in the figure below, offers the user a catalogue of videos which are sorted into all of the core topics within Key Stage 2 Mathematics. These videos were all developed by a sizable group of educational representatives, which regularly re-occur throughout the videos and aim to deliver these topics through a discussion style at the child's level of intellect; aiming to make these videos as fun as possible. BBC Learning Zone has a general "Youtube" layout and everyday objects are used within the videos which are familiar to this age group.

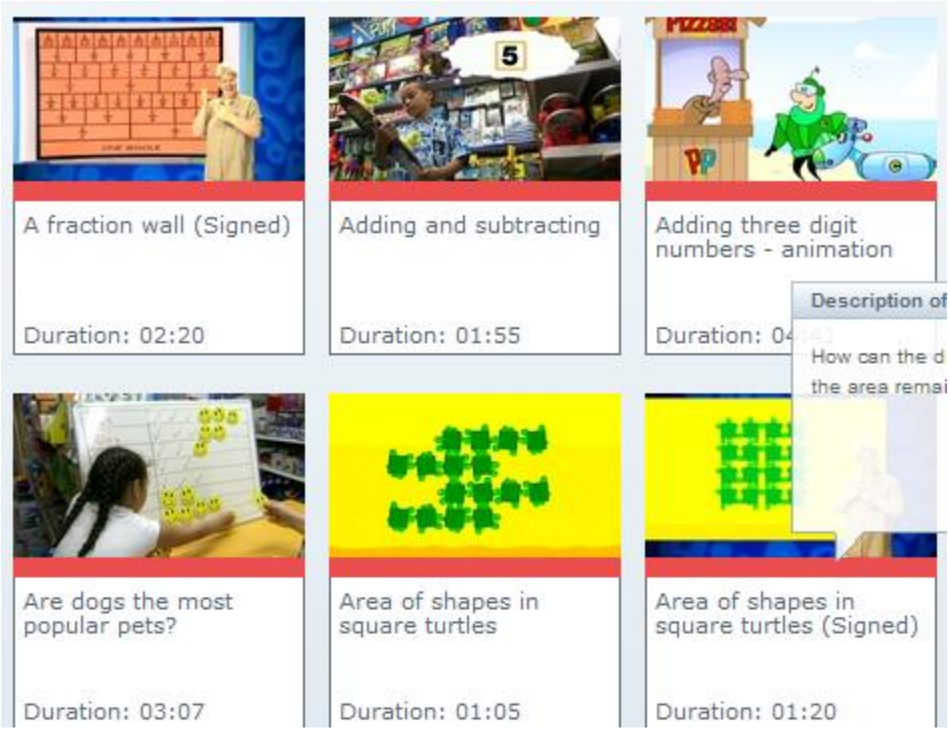


Figure 10: BBC Learning Zone

3.4.4 Text-based choice

Below is Figure 11 which represents the results of software selection for text-based software which sports a text-based environment. Only these three which were sourced, adequately satisfied the requirements of the experiment, therefore it seemed that this category was becoming more redundant as a teaching method, compared to the other three categories. The text-based version of BBC Bitesize was not chosen, again, due to a BBC title already being chosen for another category, and after further inspection, it was clear that MathsNet did not supply enough learning topics for the experiment. Therefore, Happychild.org was chosen as the software to be included within the experiment.

Text-Based Software	Key stage 2 version?	Free/Trial/Paid ?	Sample covers 4+ core topics?	Comments
Bitesize Revision (text version)	Yes	Free	Yes	A text alternative from BBC Bitesize. If another BBC title is chosen from another category, it would be better to pick a different title in this category, however.
happychild.org	Yes	Free	Yes	A commercially researched and developed website devoted to teaching KS2 students in a simple textual manner.
MathsNet	Yes	Free	No	Although this is a commercial resource aimed to teach KS2 students, this doesn't seem to provide sufficient learning topics.

Figure 11: Selecting text-based software

Happychild, represented in the figure below, is a text-based teaching environment which uses stories aimed at the child's level of intelligence in order to familiarise them with the concepts within Key Stage 2 Mathematics. The interface is very simple, providing links to both readable exercises and printable worksheets which the child is expected to use in order to interact with the exercises.

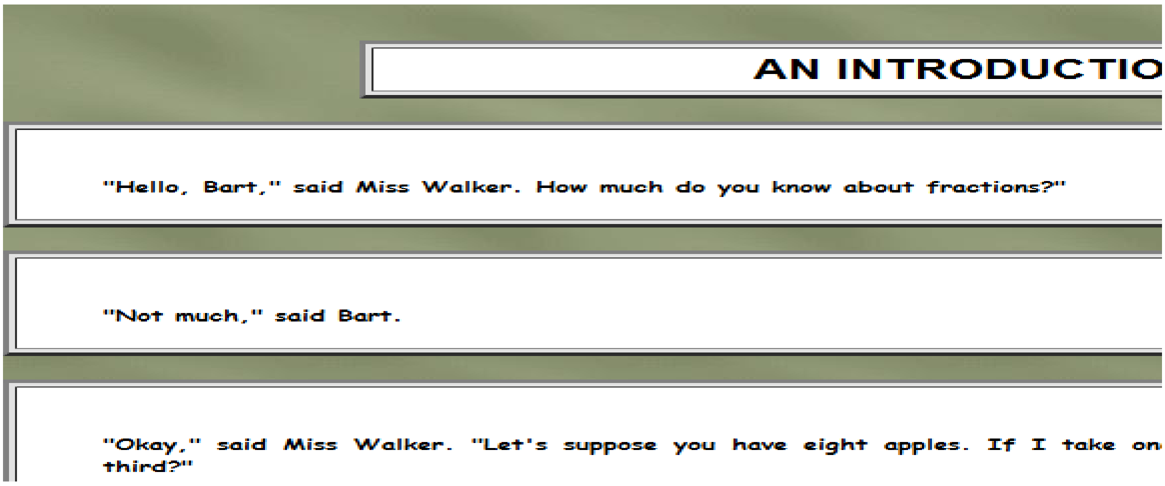


Figure 12: Happychild

4. Results

This section of the report deals with analysing and evaluating the results of the experiment, which was the main objective of this report. The experiment aims to uncover what succeeded and what each of the software lacked in terms of HCI and their multimedia elements. It is important to consider which software offered the richest HCI experience overall, nevertheless, it is also crucial to further analyse these results under each category to scrutinise the variation of success within each guideline. The analysis of results follows the structure of the checklist of guidelines which were discussed within the Methods section. Due to chart and graph sizes, category names have been shortened and are thus referred to in this chapter as follows:

- Maths Whizz - Game-based category = Maths Whizz (Game)
- Topmarks.co.uk – Picture/Animation-based category = Topmarks (Pic)
- BBC Learning Zone – Video/Audio-based category = BBC Learn (Vid)
- Happychild.org – Text-based category = Happychild (Txt)

For the full results of each software that was evaluated within this experiment, refer to Appendix A – D within section 7 of this report.

4.1 Overall score for each software/category

It is clear and certainly not of surprise that all of the software contained some problem or another. The overall scores did largely vary, however. Maths Whizz (game) scored an overall 85.34% with the least amount of issues, while Happychild (Txt) was at the other end of the spectrum with 52.59%. Topmarks was not far off this low score with a marginally higher score of 56.90%. BBC Learning Zone received an average score of 68.10%.

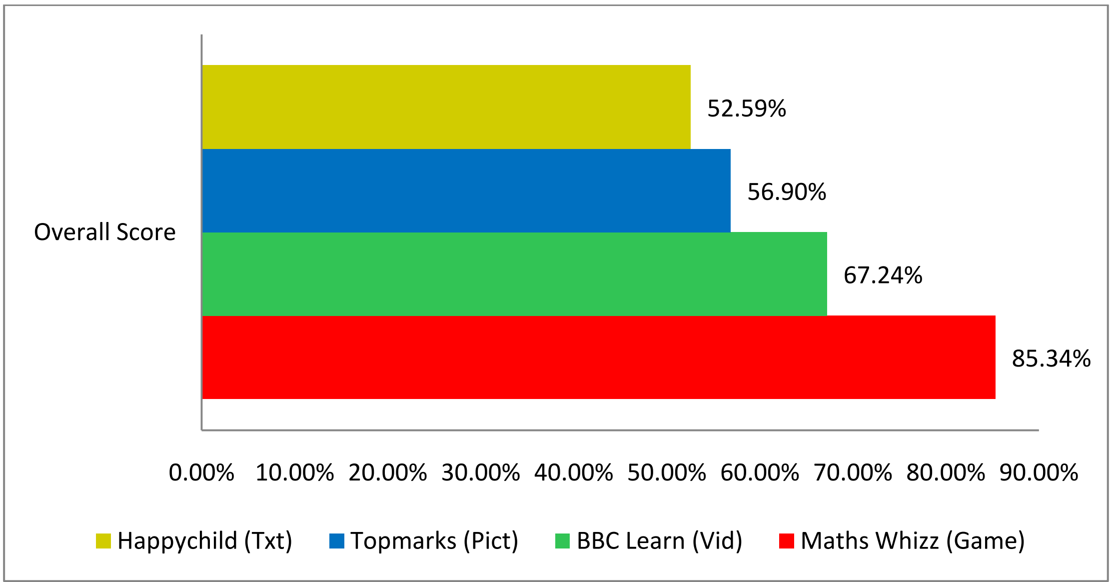


Figure 13: Overall software scores

Maths Whizz seemed to score highly and consistently in each of the four sections, which attributed to its high overall score. At face value, deploying learning material this way seemed to succeed, primarily due to the game based nature, which managed to both visually and interactively allow the user to grasp concepts and effectively "learn by doing" (Fleming, 2009). The game-based environment was rich in cognitive resources (on-screen calculators, timetable charts etc.) and by having an in-built achievement and score tracking system, it dealt well with encouraging the target age group to advance in knowledge and skill.

BBC Learning Zone had some notable flaws such as lack of navigation, and perhaps wasn't as simple, bright and attractive enough. As will be further discussed, its main drop in score was associated with the design of the environment which wasn't as engaging, mainly due to the lack of interactivity and direct control of outcomes.

Topmarks and Happychild seemed to score rather similarly and this was mainly due to the lack of features to effectively engage with the user and keep them interested. Whilst the narrative style was very interesting and intuitive within Happychild for this particular age group, it was bordering on dull and plain and didn't promote interaction. Topmarks seemed to share many of these pitfalls, although it did score a little higher due to promoting some interactivity.

4.2 Results for "Practicality of the Software"

Most of the software environments seemed to deal well with and did promote practicality, BBC Learning Zone and Maths Whizz with 85% and Happychild with 80%. However, the exception to this was Topmarks, which only scored 45%. There was little consistency with placement of functions and all exercises seemed to be developed by different external developers, which resulted in an inconsistent placement of functions and information. These results are displayed in Figure 14 below.

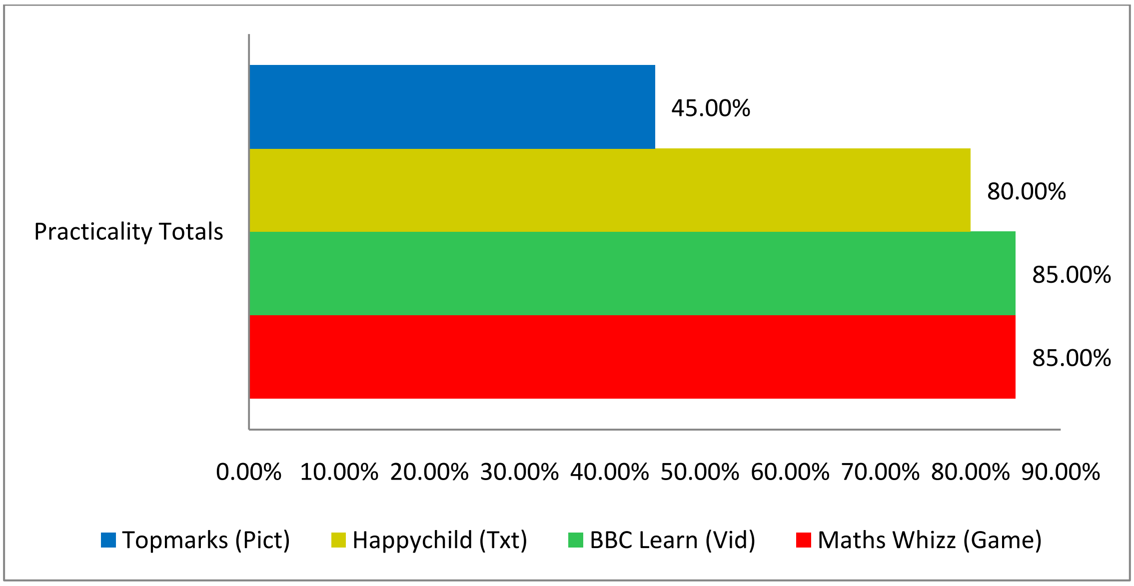
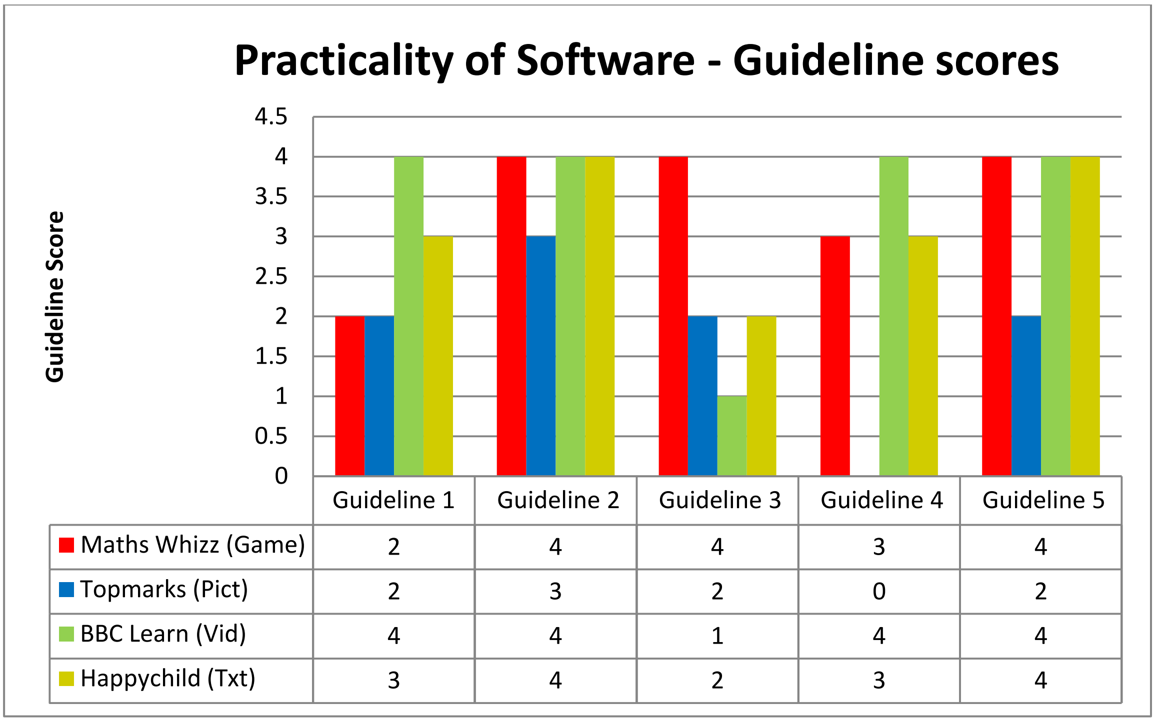


Figure 14: Category totals for "Practicality of the Software"

Below this is Figure 16, which isolates the five guidelines within this category, where the results of each guideline are discussed.



	Guideline 1	Guideline 2	Guideline 3	Guideline 4	Guideline 5
Maths Whizz (Game)	2	4	4	3	4
Topmarks (Pict)	2	3	2	0	2
BBC Learn (Vid)	4	4	1	4	4
Happychild (Txt)	3	4	2	3	4

Figure 15: Guideline totals for "Practicality of the Software"

Guideline 1, which is "Does the software provide adequate facilities for the user to address the core concepts of Mathematics?" was only fully achieved by BBC Learning Zone, where these concepts were fully demonstrated, both audibly and visually. Happychild scored 3, where these concepts were delivered well throughout its story-based narrative style, however, printable worksheets which were offered did not cover all of the core concepts it was teaching. Topmarks and Maths Whizz both scored 2 – the former for inconsistently providing these facilities throughout the software and the latter for lacking in practice exercises which weren't available in some of the harder games, yet were present in the easier ones.

The information provided to the user was fully relevant to the task and context (Guideline 2) for Maths Whizz, BBC Learning Zone and Happychild, achieving this through its games, videos or narrative text, although Topmarks scored less, due to the information occasionally coming across as unclear within its interactive activities..

Only Maths Whizz managed to provide functions which were fully relevant to the tasks and context (Guideline 3), although all of the other software failed to achieve this well. Happychild had a lack of functions due to the simplicity of its environment, however, the functions it did provide managed to fully communicate the task at hand (e.g. a control to move from screen to screen for each exercise).

Topmarks fully failed employ consistency within its design (Guideline 4), commands, actions and terminology were inconsistent and designed completely different from exercise to exercise. This was again down to these exercises being developed from many different sources. Maths Whizz and Happychild had some minor issues with this guideline – within the game-based environment, some of the higher difficulty games seemed to be designed in a slightly different format from all the other levels of games, where some of the actions were omitted, where it did not seem logical to do so (such as the confirm answer pop-up dialogues). The textual environment seemed to adopt two different design structures; one for the exercises and another slightly different structure for the worksheets. While it didn't seem logical to have two separate structures for these tasks, these structures were consistent with actions, commands and terminology within themselves.

Three of the software managed to fully present information in a natural and logical order for the target age group (Guideline 5), where information was appropriately attention grabbing within the games and the videos and even the narrative style within Happychild managed to achieve a natural discursive style with this age group. Topmarks, however, didn't fully grasp this idea, where many of the exercises presented information in an order which would better suit older age groups.

4.3 Results for "Simplicity of the Software"

Figure 16, below, demonstrates how Maths Whizz and Topmarks achieved well in this category (80.56% and 72.22% respectively), thus managed to effectively deploy its learning material whilst remaining relatively simple. Maths Whizz excelled at automating all workload, synchronously updating with the users actions and always displayed the necessary elements to a given task within each screen. Topmarks didn't achieve all of this as well as Maths Whizz, but scored surprisingly high when comparing results to the practicality section of the experiment.

BBC Learning Zone, again, scored averagely with 63.89%. Whilst satisfying some guidelines just as thoroughly as the aforementioned software, its main pitfalls laid with cognitive resources and no methods reduce the user's need to remember information. Happychild's educational environment struggled to satisfy any of the guidelines within this section beyond "middle ground" and scored 47.22%. Amongst other issues, the main ones with this software was a lack of "self-evident" design, and again, there were no cognitive resources available to help the user.

Below Figure 16 is Figure 17 and Figure 18, which displays the results of each guideline within this section against each of the software.

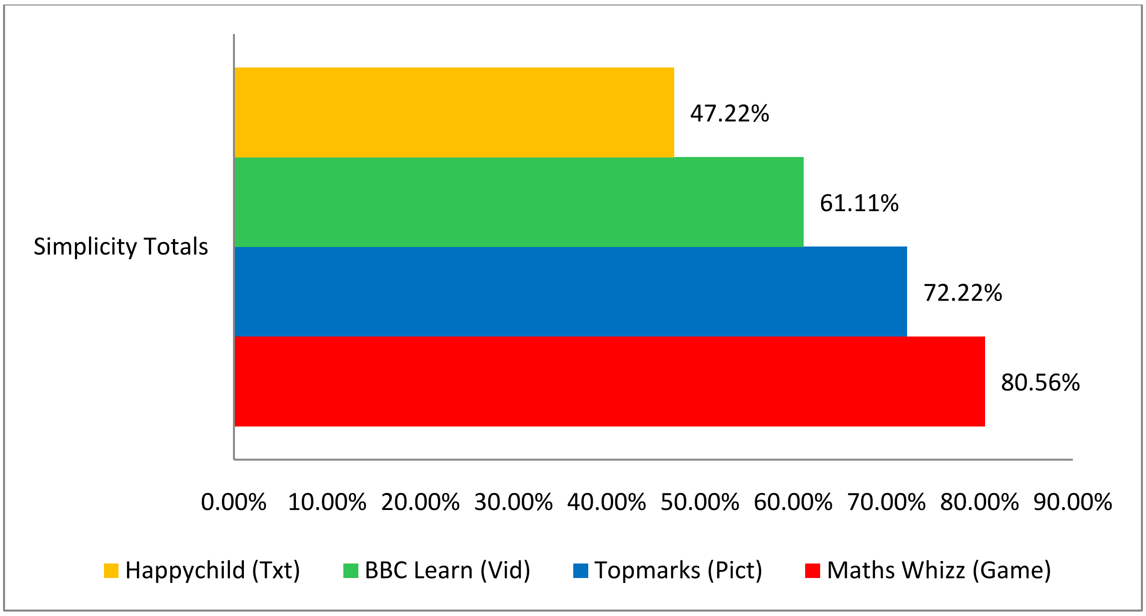
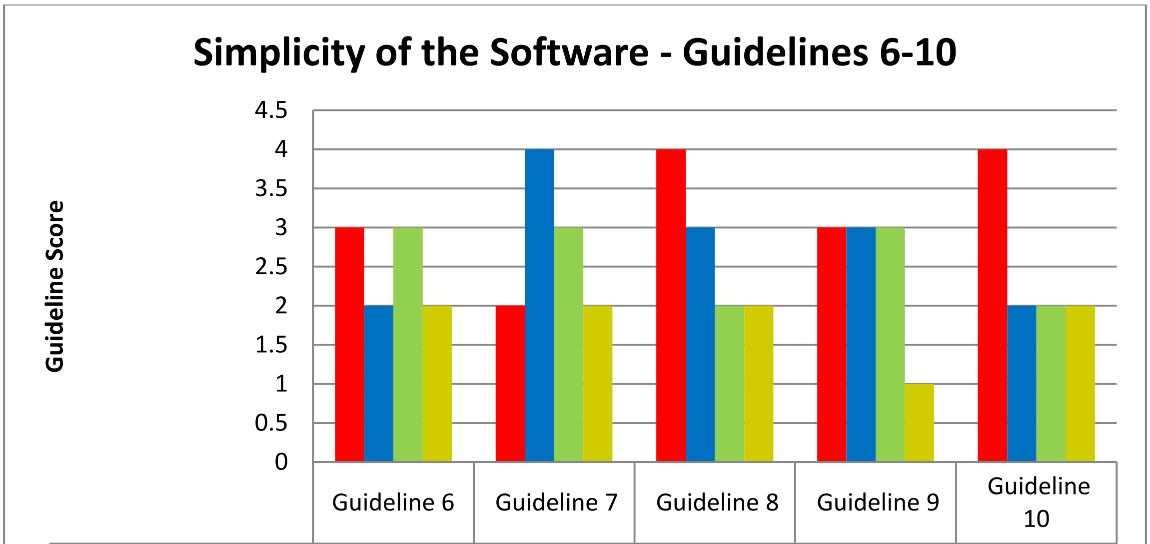
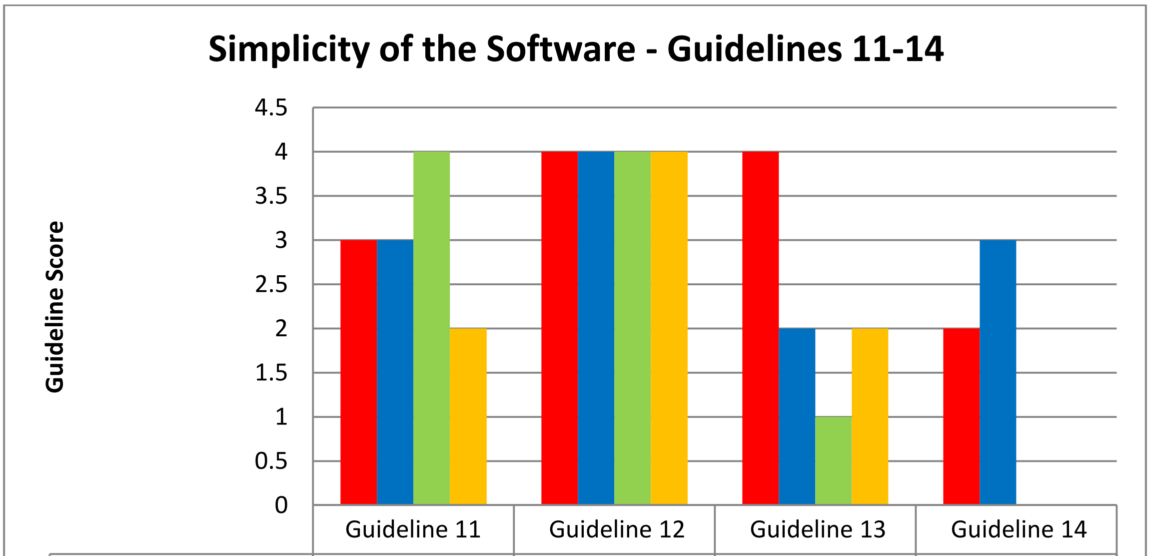


Figure 16: Category totals for "Simplicity of the Software"



	Guideline 6	Guideline 7	Guideline 8	Guideline 9	Guideline 10
Maths Whizz (Game)	3	2	4	3	4
Topmarks (Pict)	2	4	3	3	2
BBC Learn (Vid)	3	3	2	3	2
Happychild (Txt)	2	2	2	1	2

Figure 17a: Guidelines 6-10 totals for "Simplicity of the Software"



	Guideline 11	Guideline 12	Guideline 13	Guideline 14
Maths Whizz (Game)	3	4	4	2
Topmarks (Pict)	3	4	2	3
BBC Learn (Vid)	4	4	1	0
Happychild (Txt)	2	4	2	0

Figure 18: Guidelines 11-14 totals for "Simplicity of the Software"

None of the software fully achieves the requirements of Guideline 6, which requires each screen to be free of clutter. Both Maths Whizz and BBC Learning Zone dealt best with this, Maths Whizz only cluttered a few of the screens with some extra controls which could have been omitted and BBC Learning Zone's only "slip-up" was that each screen linked to other Key Stage 2 subjects which were not relevant to Mathematics - this same problem was even more apparent within Topmarks and thus it scored an even lower mark of 2. The problem with Happychild's interface is that many of the textual scenarios within the exercises were lengthy yet presented on a single page, thus requiring excessive scrolling. If this information was divided into several pages, this would also allow the user to easily return to that point within the exercise without excessive scrolling and remembering exact parts of the page.

Topmarks scored well with Guideline 7, which dealt with condensing any elements which were not central to the task at hand. Some elements within BBC Learning Zone could have been condensed whilst watching a particular video, such as the information boxes, one of which described the video currently playing and was to the right of the video and the other one below the video suggested further ideas with the task – both of these information panels consumed a lot of space and could not be condensed by preference.

Commonly used elements and options for a given task (Guideline 8) were fully available and accessible within Maths Whizz and the interface was very intuitive and timely with these options and elements which would only appear when appropriate. Within Topmark's environment, the options and elements were available in most tasks, although they weren't present where expected in a few exercises. Both of the other software were inconsistent with this guideline, fully providing options and elements in one area, yet omitting them in another.

Most of the software environments managed to promote a "self-evident" design, BBC Learning Zone adopted a "Youtube-style" design which appeals to all age ranges, yet it was more simplified, therefore the target age group would be able to navigate with little previous experience. Topmarks' interface design was a little complicated in some areas of the software and Maths Whizz had no issues with its self-evident design within the in-game exercises, however, the main menu was a little "scattered" and could have been designed a little simpler for this age group. Happychild's textual interface came across as far too dull and formal in design for this type of age group and would be somewhat difficult to navigate around, where nothing stands out and grabs the user's attention.

Happychild, BBC Learning Zone and Topmarks didn't deal well with automating unwanted workload. The simple text-based environment of Happychild made it difficult for the intake of information and lacked the "fun factor" and speed of which information could be delievered. Similarly, Topmarks did require a lot of note-taking, despite the presence of pictures and animations and BBC Learning Zone also required similar periods of external note-taking. Maths Whizz dealt with these issues by reducing a lot of the workload on-screen by exploiting its gaming nature to make the task more enjoyable and faster than any of the other software environments. All of the information is recorded in-game and did not require any external note-taking out with the software.

BBC Learning Zone managed to clearly communicate its instructions unmistakably through its video lessons, and as a result, this came across as "straight to the point" for this age group (Guideline 11). Maths Whizz and Topmarks managed this rather well as their environments offered some interaction and manipulation of outcomes. However, in both cases, instructions for these exercises could be too lengthy in some areas and lacking in others. The text-based nature of Happychild meant that it struggled to minimise the length of the explanation process within its concepts and techniques to learning various mathematical concepts, therefore tended to take the "long route" explaining these – something which was done visually by the other software in a fraction of the time.

All of the software environments had their own individual techniques which successfully introduced new topics with aids to interpretation which are meaningful to this particular age group (Guideline 12). Within Happychild's narrative style, a story was always cleverly aimed at this age group to introduce them to a new idea. Maths Whizz always introduced new information at the beginning of each game with animated demonstrations and Topmarks used a similar technique. Everyday objects which are common to the target age group were cleverly used within BBC Learning Zone’s video exercises in order to help apply meaning to the exercise at hand.

Maths Whizz Managed to fully reduce the user’s need to remember information (Guideline 13), with techniques such as having a confirmation are on the screen which notified the user on their progress within calculations and what actions they had to perform before this was complete. Topmarks and Happychild only received half marks for this guideline, the former where some exercises would detail progress but other tasks were ongoing and had no progress of completion but failed to communicate this to the user. Happychild partly achieved this in its narrative style which would use phrases like “nearly half way there!” and “almost finished...”, although, with no resources, there were no other attempts made to reduce the need to remember information. BBC Learning Zone scored the least by only vaguely achieving this by using short summaries of what was covered in the topic at the end of the videos.

Only Topmarks scored well with being rich in cognitive resources, which support high-level tasks for this age group (Guideline 14); offering on-screen calculators, notepads, interactive whiteboards and fact sheets in most of the exercises, when expected (with only a few exceptions). Despite Maths Whizz generally being rich in other resources, it was less consistent than Topmarks within this guideline and this was due to Maths Whizz only offering these resources out with the game exercises as a standalone feature, where it would have been more beneficial to include an option to use these within the exercise themselves. BBC Learning Zone and Happychild did not include any of these resources.

4.4 Results for “Communicating the Subject”

Most categories seemed to fail to communicate effectively with the user. As can be demonstrated from Figure 18 below, the only clear exception to this was Maths Whizz which achieved an almost perfect score of 93.75%. Again, BBC Learning Zone remained with an average score of 62.5%, with its main pitfalls being that there was a lack of help or advice for the user to help them navigate, the user was not able to interact or control outcomes and because of this, there was no way for the user to track accomplishments or achievements. However, this environment dealt very well with explaining the task at hand with good use of metaphors, terminology and symbols.

Both Topmarks and Happychild received a low score (53.13% and 46.88% respectively) and in the former’s case, this was mainly due to the lack of clear feedback for the user when completing exercises. The latter failed to incorporate an achievements system and progress tracking, and the user generally had no control of the exercises and they could not interact with the outcome.

Below Figure 19 is Figure 20 and Figure 21, which displays the results of each guideline within this section against each of the software.

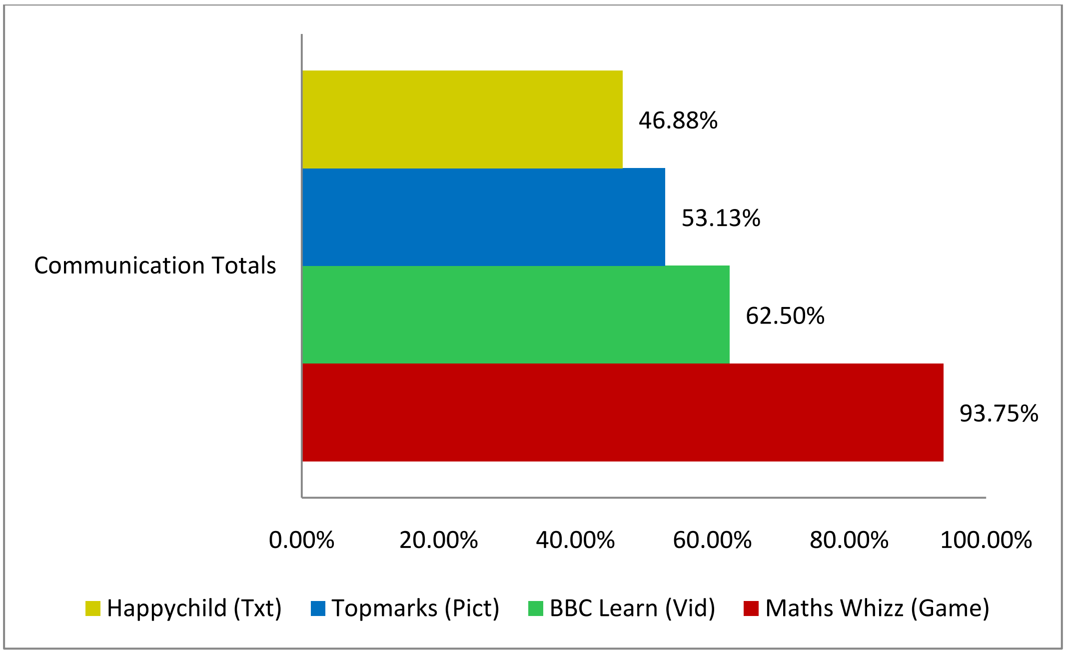
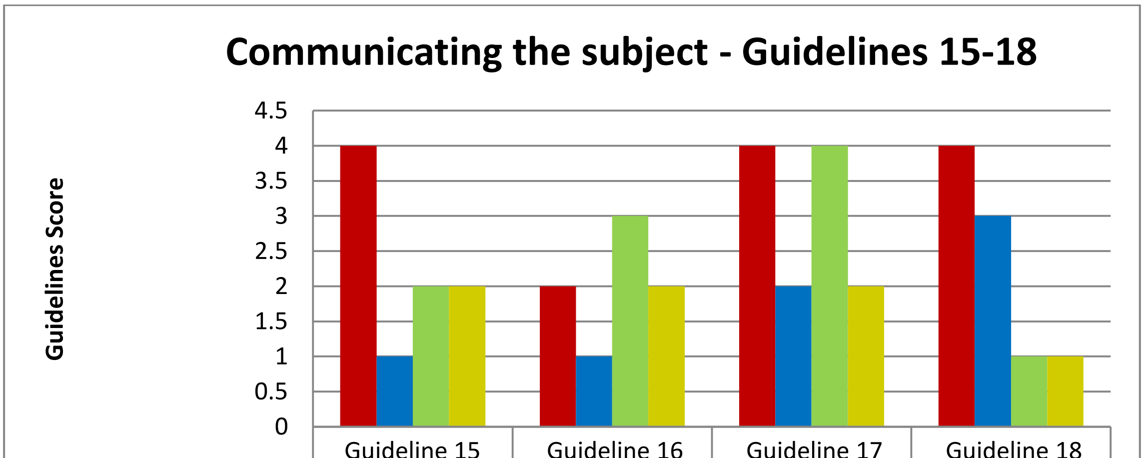
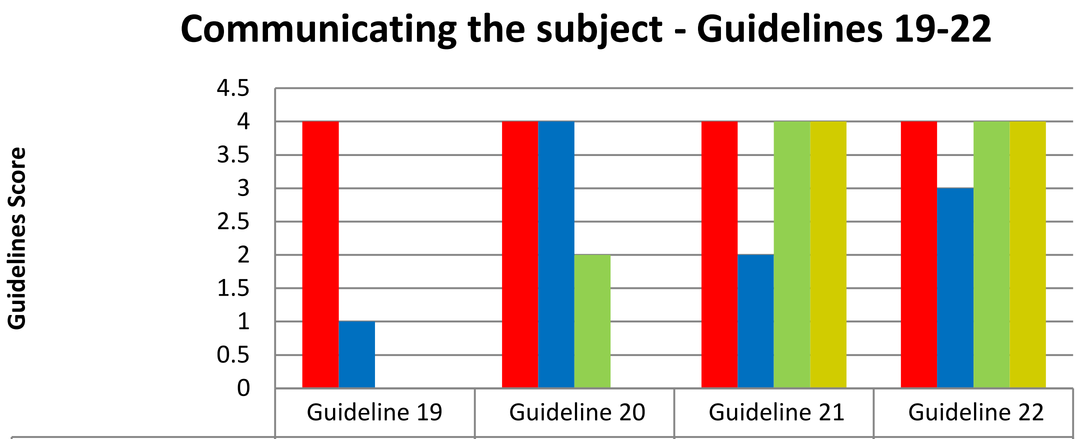


Figure 19: Category totals for "Communicating the Subject"



	Guideline 15	Guideline 16	Guideline 17	Guideline 18
Maths Whizz (Game)	4	2	4	4
Topmarks (Pict)	1	1	2	3
BBC Learn (Vid)	2	3	4	1
Happychild (Txt)	2	2	2	1

Figure 20: Guidelines 15-18 totals for "Communicating the Subject"



	Guideline 19	Guideline 20	Guideline 21	Guideline 22
Maths Whizz (Game)	4	4	4	4
Topmarks (Pict)	1	4	2	3
BBC Learn (Vid)	0	2	4	4
Happychild (Txt)	0	0	4	4

Figure 21: Guidelines 19-22 totals for “Communicating the Subject”

Maths Whizz managed to provide appropriate, clear and timely feedback for the user, keeping them well informed with feedback through pop-ups and several different progress bars during exercises (Guideline 15). BBC Learning Zone didn’t achieve this so well and was only moderately achieved as short pauses or summaries within the video, which wasn’t very effective (these pauses often weren’t long enough). As Happychild was a text-based environment, this was only achieved through the text itself, although allowing no interaction meant that there was no feedback available. Topmarks was very inconsistent and only a few of the exercises (ones which were perhaps developed by the same source) provided some feedback during interaction.

Topmarks also struggled with Guideline 16, which requires a structure within the software to be adopted, where only the main menu appeared to have a structure, where related items were put together such as options. Maths Whizz dealt with this a little more successfully by having some structure within their exercises and their menus. BBC Learning Zone, however, coped with this the best, which mainly stuck to a consistent structure (one central menu of video links with every video using the same structure).

Both Maths Whizz and BBC Learning Zone communicated the sequence of the task at hand well (Guideline 17). This was communicated well with the videos themselves and the time progress bar within the video for BBC Learning Zone and Maths Whizz achieved this with all of its in-game exercises where there were two progress bars, with the progress bar on the top of the screen informing the user of how many questions they had completed (and subsequently how many were left to go) and the bottom bar would inform the user of their current score in a percentage of correctness, thus the user would remain well informed of the beginning, middle and end of exercises. Topmarks occasionally communicated progress, although inconsistently and Happychild lacked interactivity, with the only method of communicating the progress of a task was the position of the scroll bar on the page.

Happychild and Topmarks also lacked help and documentation (Guideline 18) and a notable pitfall about both of these was new mathematical symbols introduced in newer topics provided no explanation or help to allow the user to familiarise with. In contrast to this, Topmarks did offer extra help for any new elements within a task (although some of it came across as unclear for this age group), but Maths Whizz dealt with this most communicatively, where mathematical concept and controls contained a small “i” which could be hovered over with the mouse pointer if the user felt they needed to acquire more information about an object.

Neither Happychild or BBC Learning Zone provided features which kept track of the user’s accomplishments (Guideline 19) and Topmarks only stored some statistical information in a few of its exercises. Contrary to this, Maths Whizz fully supported this feature with scoreboards, charts, graphs and a medal system for each exercise. An additional feature was that these scores could be compared against the user’s “buddies” scores (this software included a friends list feature) if they have completed the same exercise. These features were made possible due to each user owning their own account and could therefore retain their own scores over any period of time (as opposed to being dependant on browsing history and cookies).

Guideline 20, which is “Does the interface properly communicate between pre and post submission of answers?” was achieved in full by Maths Whizz and Topmarks. Maths Whizz would synchronously update after the user performed an action and used on-screen prompts intuitively, advising the user to submit once they had entered numerical values. Topmarks dealt with this through all exercises analysed during the experiment, which was a surprising observation due to the issues with inconsistency. This method was rather weak within the videos of BBC Learning Zone and there were no functions or interactivity within Happychild to even warrant this feature.

Guidelines 21 and 22 almost accomplished identical results. It was apparent that, if the software scored highly with communicating metaphors, concepts and following real world conventions for this age group (Guideline 21), then mathematical symbols and terminology would also have been used appropriately (Guideline 22). Both of these were fully achieved by Maths Whizz, BBC Learning Zone and Happychild who managed to deliver this in separate ways: vocally within the videos, through story-based examples in the text-based environment and explanations were used with demonstrations before each exercise. Topmarks seemed to have exercises which were using analogies which wouldn’t be understood properly by such a young age group (such as the money counting exercises which were often discussed in a complex manner), however, Topmarks did deal a little better with mathematical symbols and terminology, only occasionally offering on-screen scientific calculators, as opposed to a standard calculator which is all that would have been required.

4.6 Results for “Efficiency of the Software”

With a similar order of results, Maths Whizz received a decent score of 82.14%, BBC Learning Zone scored rather well with 67.86%, Topmarks and Happychild scored low (50% and 46.43% respectively), as demonstrated in Figure 21, below. The only major pitfall for Maths Whizz was often failing to increase the level of difficulty of exercises with the user’s advancement in skill and knowledge.

Due to the nature of the video environment within BBC Learning Zone, it didn’t really allow the user to feel in control of the outcomes within the exercises and this could perhaps be a little discouraging. The environment wasn’t rich in shortcuts for the regular users, however, did adopt a “view style” from the menu. Topmarks failed to offer shortcuts and was “50/50” with many aspects within this category, such as a positive appearance for the age group, supporting the user's advancement in knowledge and skill and not allowing the user to feel in control.

Figure 23, directly below Figure 22 details the results of the guidelines within this category.

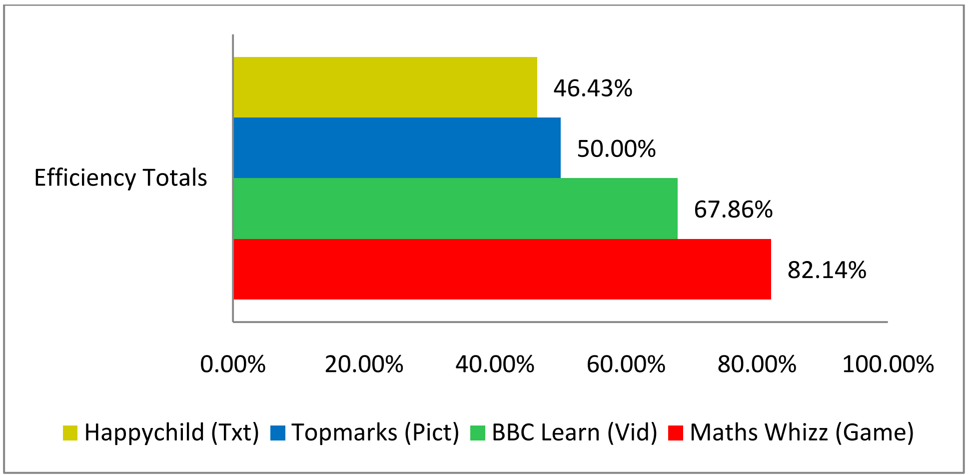
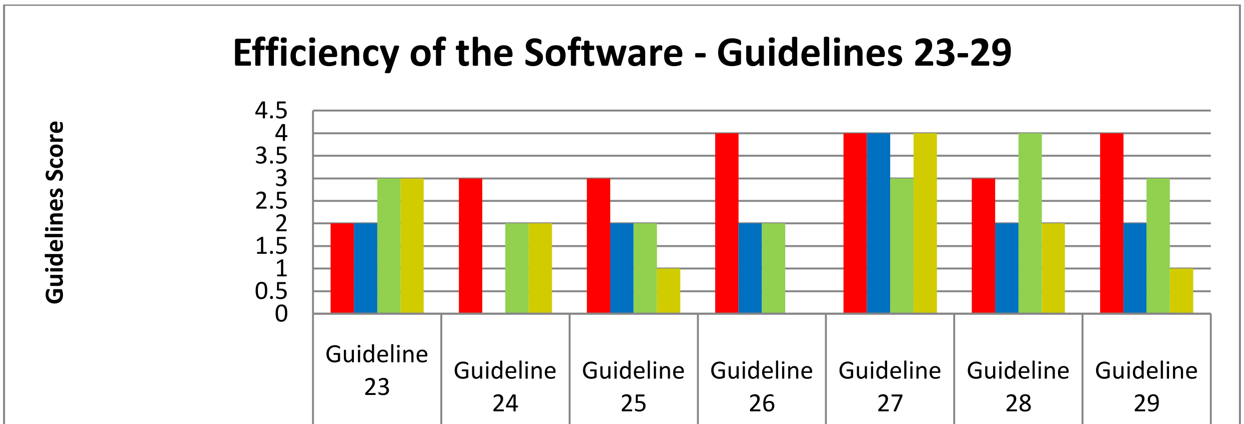


Figure 22: Category totals for "Efficiency of the Software"



	Guideline 23	Guideline 24	Guideline 25	Guideline 26	Guideline 27	Guideline 28	Guideline 29
Maths Whizz (Game)	2	3	3	4	4	3	4
Topmarks (Pict)	2	0	2	2	4	2	2
BBC Learn (Vid)	3	2	2	2	3	4	3
Happychild (Txt)	3	2	1	0	4	2	1

Figure 23: Guidelines 23-29 totals for "Efficiency of the Software"

Guideline 23 dealt with the accommodation of the user's continuous advancement in knowledge and skill. BBC Learning Zone and Happychild dealt with this comparatively better than the other software, where the difficulty of the exercises naturally increased during the length of the video and the text within Happychild dealt well with continuously increasing the difficulty of the topic throughout the exercise. No skill tracking was available within Topmarks itself and Maths Whizz partially achieved this by eventually calculating a "Brain Age" which would eventually increase the difficulty of the games, yet the games themselves, which were quite lengthy (usually around 12 questions per game) never seemed to increase in difficulty – thus repetitive.

None of the software achieved placement of shortcuts to accommodate the users who gained more experience from repetitive use to full efficiency (Guideline 24). Topmarks provided no shortcuts at all, thus the user had to rely on the use of the browser shortcuts to make their way about. BBC Learning Zone offered some shortcuts by allowing the user to switch to a grid view with the videos, in order to view more of them at once (although less information would be available for each video this way). Nonetheless, Maths Whizz dealt with this a little more efficiently by allowing the user to exploit shortcuts to different topics if they don't want to access the daily exercises in sequence.

It is unsurprising, that due to the text-based nature of Happychild, there was no means of on-screen interaction with the exercises to allow the user to feel responsible and thus "in control" of their actions (Guideline 25). Maths Whiz succeeded with this the best, which was naturally attributed to the high involvement and interaction that comes with gaming. Therefore, Topmarks and BBC Learning Zone scored less due to little interaction causing less control of outcomes.

Math Whizz also fully employed a "forgiveness" strategy (Guideline 26) by disallowing invalid entry (e.g. only allowing the entry of numerical values within the answer boxes). Confirmation boxes were also presented upon entering some answers. Again, validation appeared in some of the exercises for Topmarks, yet there was no sign of this in others. There was not enough interaction with BBC Learning Zone for much forgiveness to be employed and Happychild allowed no entry at all and could not employ this guideline.

The environment of Maths Whizz handled "error recovery" well (Guideline 27), where an error was encountered during an in-game exercise, where the software saved the progress and allowed the exercise to be continued after this error was recovered. No errors were encountered with Topmarks or Happychild, however, there were a few errors with BBC Learning Zone, where videos no longer existed but were still showing as available from the main menu.

"Undo" and "redo controls were available (Guideline 28) throughout Maths Whizz menu's although they could have been present in-game, where the user would benefit from the option to reverse their answer before the pop-up confirmation box. These features varied within Topmarks and Happychild's interface was too limited to offer this. BBC Learning Zone's information was always delivered on video; therefore this feature would not be required in this environment.

Guideline 29, which was concerned with how well the overall interface design within each of the software environments related to the target audience, provided some interesting results. Maths Whizz scored full marks with this as the environment came across as very appealing to the target age group; sporting bright, colourful and attractive design within the menus and in-game. Furthermore, the games, animations and attractive, informative pop-ups also promoted an environment which ultimately was most optimal for this age group.

Next in line was BBC Learning Zone which didn't promote the same level of usability but still promoted this well with its simple, yet attractive "youtube style". Topmarks only promoted optimal interface design within the animated exercises, although the general menu design outside these exercises was generally quite dull and unappealing for this age group. This dull and uninspiring appearance was, however, carried throughout all of Happychild's interface design and thus would not connect well with the user.

4.7 Summary of Results

Whilst all of the software showed a certain level of problems within this evaluation, it is a noticeable observation that Maths Whizz generally succeeded overall and did promote an engaging Human Computer Interaction experience. It seems, that from a project of this scale, that a gaming environment best accommodates interaction with this age group when deploying the subject of Mathematics. This was greatly due to the nature of a gaming environment promoting the most interaction, which is, afterall, the basis of HCI. The discussions and conclusions section within section 5 further discusses the outcome of the issues and success which was touched upon within the results section.

5. Discussions and Conclusions

This section discusses the overall conclusions which arose as an outcome from carrying out the literature research and conducting an experiment. The discussion then leads onto overall "Guidelines for Key Stage 2 Mathematics Software" which have resulted directly from the findings within this report. A consideration of further research areas and further development that a project like this could follow is then discussed.

5.1 General Overall Conclusions

This report uncovered a lot of useful information which contributed towards the research question which sought to identify how suitable HCI within this type of software is, with regards to its use of multimedia and underlying learning models. It was noticeable from the lack of direct research in this exact area, that it would only be beneficial to refine and personalise an evaluation method which would fully benefit development in this area. Researchers such as Nielsen and Powal re-iterated the importance of adapting and creating specific usability guidelines within the subject area of the software being evaluated (Nielsen et. Al, 1990) and thus adapting these popular guidelines in the context of Mathematics software for children was done with close academic rigour in order to try and validate the experiment as much as possible. Further to this, it was evident that from research such as (Dix et al, 2004**)** and (Roschelle, 2001) that there are many design and usability problems which have discouraged the use of this type of software on a much larger scale, and certainly, a lot of these issues surfaced from the experiment.

One of the major findings of this report concluded that a game-based environment, was, by far, the most optimal method of learning for Key Stage 2 Mathematics students. Looking back on these results, the success of this software was predominantly motivated from "learning by doing". This was succeeded by presenting information in a natural manner for this age group, eliminating the users need to externally use resources or remember unnecessary amounts of information, being rich in feedback, rewarding the user for their progress and generally adopting a colourful, friendly atmosphere and design. On the other end of the spectrum, appearing to be largely unsuccessful, was a text-based environment, however this was not surprising when referring back to the research of (Green et al, 1996) who discussed how such an environment was ultimately un-motivating and lacking in interactivity for such an age group. This was certainly touched upon throughout the whole experiment where the textual environment did not succeed beyond practicality, failing to offer resources, control of outcomes, track of progress and accomplishments, lacking in self-evident and positive design and difficulty of navigation.

Therefore, taking into consideration all of the research within the Literature Review which contributed to constructing 29 in-depth guidelines which were all discussed and analysed within the Results section against four different environments, it will be beneficial to consolidate all these findings, both positive and negative and convert them into a set of guidelines which can contribute to selecting or designing Key Stage 2 Mathematics educational software. These final guidelines are sectioned under the same headings which were used within the evaluation. This same technique was performed by Nielsen within his research (Nielsen et. Al, 1990), which contributed towards his refined set of Heuristics, extensively used today.

5.2 Guidelines and Selection Practices for Key Stage 2 Mathematics Educational Software

5.2.1 Practicality of the Software

Always remain within context of Mathematics

Whatever method is used to deploy the learning material it must always be centred around the subject of Mathematics. This software is not aimed at other subjects such as English or Science and should not stray too far into a discursive manner – it's all about the numbers and thus language should be kept simple. The multimedia delivering this material should use its resources to represent the subject first and foremost and showcasing its ability and skills should be secondary to this. For example, a Game teaching mathematics should be more about the subject at hand than the features offered by the game itself.

Consistency is Key!

The functions, terms and headings should always remain within the same style and position from screen to screen. Common elements such as main menu controls must always be available within every single screen and options within one exercise must be similarly located within another exercise.

Present information in a way which is optimal to children

The software is aimed at children and thus should be presented in a structure which is most optimal to their age group. The child should be able to understand the "flow" of information presented in front of them from the very first use of the software. In addition to this, something should never be explained in complex terms, all examples should fully relate to the child's lifestyle.

5.2.2 Simplicity of the Software

Isolate the subject; there should be no clutter!

If a control is not central to the task at hand, then it shouldn't be there. If calculations or times table charts are not necessary for a particular task, then these elements should not be visible during that task. Educational software which is accessed online better succeeds if it avoids a standard webpage layout, and particular effort should be made to remove any links which are not relevant to the subject of mathematics. All information needed should fit on one screen with no scrolling necessary and if the exercise is lengthy it should be split up into several pages. Elements which are only partially relevant or optional to a task should be condensed and expandable upon request – all of this will avoid distracting the child from the task.

The layout should always be "self-evident"

All features within the educational environment must stand out within seconds of the child engaging with the interface. Topics, controls and options must be placed in such a way that the child will know where to go from the main menu and how to get back to the main menu.

Everything should be achievable on-screen

The purpose of educational software is to speed up the process of mundane tasks which would usually be done on paper. The child should not have to use any external resources such as pencil and paper to take notes or complete the exercises. Note taking and entry of answers should be achievable on-screen and thus the software must promote a high level of interaction. Further to this, cognitive resources should be available and not just externally as “stand-alone” within the software but should also be usable within any tasks which would normally require this resource, such as an on-screen calculator being selectable on-screen during the task, where required. Finally, printing tasks or worksheets before they are completed should be discouraged and these should instead only be printable afterwards when all the information has been entered on screen as this will further prevent the need to manually note-take.

Always keep “straight to the point”

Little instructions should be needed to use a function as too much information may over-complicate the task and could detract from the child’s attention span. Information which is required to complete a task should not be deployed all at once, where avoidable, and instead should be presented, when most relevant throughout the task. The exercises themselves should also not be lengthy and must also promote the option to complete later and continue from the same point.

Relate the exercises to the user

A child best learns best by examples which relate to their everyday life. This includes objects which are most common to them and examples which they can relate to personally. Therefore the material should always adopt these techniques, especially when introducing a new topic.

5.2.3 Communicating the Subject

Reassure the user and promote feedback

The child will be more encouraged and pleased if constant feedback is presented back to them, informing them of their actions. This feedback should be timely and intuitive and is best used as popup messages on screen throughout a task. Educational software rich in feedback should also display the progress of their current task; the user should always know where they are within a task, and never be “left in the dark” or unaware when a task will finish. The user should also be well informed where to enter their answer and immediately receive feedback upon entering their answers.

On-screen help should cover everything

All functions and examples should provide additional explanation, where the user may not be fully informed about the element at first glance. Generally, a separate help section is not an effective method for a young age group who are not likely to read through large amounts of help topics. Instead, each screen should provide access to help for each existing function, and this help information should not re-direct the user away from the task. This is best demonstrated using “hover over” elements, where the information will only appear for an element once the mouse pointer is rested over it for a set period of time.

Statistics, achievements and progress tracking encourages the user

Features such as statistics, achievements and information on progress are only beneficial within educational software. This will also promote the "come-back" factor, where the child is likely to revisit exercises and tasks if there are rewards for doing so and excelling. These features will also keep the user informed of their progress and can help isolate areas in which they need to improve using simple graphs and charts.

Keep metaphors, conventions and terminology simple!

Mathematical symbols and terminology should not be out with the bounds of Key Stage 2 level. That is, no scientific notification or unnecessary terms which would not be used by this age group. Terminology should be referred to universally (e.g. "x" for multiplication, "+" for addition) and on-screen resources should remain simple and not contain any unused elements by this age group; such as on-screen calculators, which should not be scientific.

5.2.4 Efficiency of the Software

Advance with the user's skill

The software should be able to detect when a user is ready to advance in skill. This is done by collecting statistical information and offering harder exercises for users who are excelling in the easier ones. Lengthy exercises should naturally increase in difficulty, as answering the same answer in different ways several times could become tedious, mundane and boring for a child. Shortcuts should also be available for the users who are more familiar with the system after constant re-use, thus able to skip screens they no longer need to access.

Always allow the user to take part

Interaction is the single most important element within good educational software and should be promoted throughout. Generally, the more interaction software promotes the better. Therefore, game-based environments have the most potential of offering a high amount of interaction, as they allow the child to directly manipulate the environment and feel responsible and in control of their actions.

The overall environment must remain bright, positive and promote fun

Avoid environments which are dull and colourless as these are simply not appealing to children – children are far less likely to explore the environment (or even know how to navigate it) if the appearance itself does not communicate the concepts of fun and positivity. Therefore avoiding environments littered in text and links in favour of colourful environments with pictures, animations and games is a better choice.

5.3 Further Work

The scale of this project could be altered or extended to suite several distinct purposes.

The 29 guidelines which were adapted from Nielsen's Heuristics and Powal's cognitive principles could be reviewed and adapted, encompassing a larger field of heuristics or usability guidelines from other various experts such as those created by Schneiderman and Holden and Butler (Kohls et. Al, 2009)**.** It would be interesting if these guidelines were adapted and extended to see if they produced further refined results within this subject area and level.

Researchers with more time and scope within a project may also want to add further evaluative approaches to the project such as Empirical experiments with sample of children studying at Key Stage 2 level, where this was not achievable within the timeframe and scope of this project along with the analytical methods. With extra time, children could take part in the experiment by sampling Key stage 2 Mathematics software over a period of time, where the evaluator would extract the outcome of these results using techniques such as "think aloud", "questionnaires" or interviews. This could then be coupled along with the analytical methods used within this project to extract further outcomes about the HCI promoted within these software packages – thus increasing the validity of the results and extracting further, more detailed research information within this field.

Another limitation within the scope of this project was the constraints of funding, which meant the most "Commercial, Off the Shelf" software could not be included within the evaluation (Although the software did still have some commercial appeal). Due to the nature and level of this project, spending large amounts on several software packages was not desirable and too costly. It may be useful if this project were to be undertaken similarly with an unconstrained budget, where the expensive, commercial Key Stage 2 Mathematics software, which is likely to have more money and time invested into its development, was purchased. Each of these categories could then be evaluated similarly to this project and it would be interesting to see how varied the results would turn out due to the "optimal commercial" variable being thrown in.

A limitation which was discussed within the Methods section of this report was the use of a single expert evaluator. If multiple expert evaluators were included within the analytical evaluation, these results could be triangulated and would be more accurate as each evaluation would be likely to discover some points about the software which the other evaluator may have overlooked. Additionally, this may allow the evaluator to include more software packages within each category, having more evaluators to reduce the workload.

Lastly, the scope of the project could be altered using any of these techniques, to specifically research a single category of Key Stage 2 Mathematics software, for example, evaluating several game-based software packages in order to analyse the variability of effective Human Computer Interaction between these.

Overall, it is desired that the research and results of this project will contribute towards the development of Key Stage 2 Mathematics educational software. With the increasing dependence and resourcefulness of technology in today's digital age, guidelines for developing or selecting educational software would be expected to increase in demand. Perhaps these guidelines may even be used a stepping stone to creating an evaluation process for other educational software.

6. References

Allison, S. (2007). E-learning in the age of multimedia. *E.Learning Age,* , 20.

Anon. *Overview of learning styles.* Retrieved 10/23/2009, 2009, from http://www.learning-styles-online.com/overview/

Arthurs, J. B. (2007). A juggling act in the classroom: Managing different learning styles. *Teaching and Learning in Nursing, 2*(1), 2-7.

Bongers, B., & van der Veer, G. (2009). HCI and design research education A creative approach. *Creativity and Hci: From Experience to Design in Education; INTERNATIONAL FEDERATION FOR INFORMATION PROCESSING; Conference on Creativity and HCI - from Experience to Design in Eduction,* Aveiro, PORTUGAL. *, 289* 90-105.

Clark, R., & Mayer, R. (2008). Learning by viewing versus learning by doing: Evidence-based guidelines for principled learning environments. *Performance Improvement, 47*(9), 5.

Cole, G. (2007, Nov 27). Magic in maths: Technology: Teaching taken to another dimension: New technology can help maths teachers bring their lessons to life, making learning more interactive and engaging. *The Guardian,* pp. 5.

David a. kolb on experiential learning Retrieved 11/13/2009, 2009, from http://www.infed.org/biblio/b-explrn.htm

Dix,A. Finlay,J. Abowd,G. & Beale,R.(2004).*Human-Computer Interaction.3rd Ed.* Essex: Pearson Education Ltd.

Fleming, N. *VARK -- A guide to learning styles.* Retrieved 10/24/2009, 2009, from http://www.vark-learn.com/english/index.asp

Gerhardt-Powals, Jill (1996). Cognitive Engineering Principles for Enhancing Human-Computer Performance. "International Journal of Human-Computer Interaction", 8(2), 189-21

Green, T. R. G., Davies, S. P., & Gilmore, D. J. (1996). Delivering cognitive psychology to HCI: The problems of common language and of knowledge transfer. *Interacting with Computers, 8*(1), 89-111.

Gupta, S., & Bostrom, R. (2009). Technology-mediated learning: A comprehensive theoretical model *. *Journal of the Association for Information Systems, 10*(9), 686.

Hartley, R. (2010). The evolution and redefining of 'CAL': A reflection on the interplay of theory and practice. *Journal of Computer Assisted Learning, 26*(1), 4-17.

Increasing awareness on using multimedia in education Retrieved 1/16/2010, 2010, from http://www.agocg.ac.uk/wshop/32/watson.htm

Inkpen, K. (1997). Three important research agendas for educational multimedia: Learning, children, and gender. *AACE World Conference on Educational Multi-Omedia and Hypermedia'97,* 521.

INTERACT 2009 home Retrieved 2/01/2010, 2010, from http://www.interact2009.org/

John, M. (2007, Jan 9). Education: Link: International: Everyone on board: Teachers in britain could well learn from the experiences of mexican schools in developing multimedia learning platforms. *The Guardian,* pp. 25.

Jones, A., Scanlon, E., Tosunoglu, C., Morris, E., Ross, S., Butcher, P., et al. (1999). Contexts for evaluating educational software. *Interacting with Computers, 11*(5), 499-516.

Kopf, D. (2007, e-learning market to hit $52.6B by 2010. *THE Journal,*

Kohls, C., & Uttecht, J. (2009). Lessons learnt in mining and writing design patterns for educational interactive graphics. *Computers in Human Behavior, 25*(5), 1040-1055.

Lapin, K. M., & Ragaisis, S. (2007). Teaching HCI in SE curriculum. *Proceedings of the 4th IASME/WSEAS International Conference on Engineering Education; 4th IASME/WSEAS International Conference on Engineering Education,* Agios Nikolaos, GREECE. 236-241.

Learnng styles take your test. Retrieved 10/23/2009, 2009, from http://www.ldpride.net/learningstyles.MI.htm#Learning Styles Explained

Leite, W. L., Svinicki, M., & Shi, Y. (2009). Attempted validation of the scores of the VARK: Learning styles inventory with multitrait-multimethod confirmatory factor analysis models. *Educational and Psychological Measurement,*

Lilley, M., Barker, T., & Britton, C. (2004). The development and evaluation of a software prototype for computer-adaptive testing. *Computers & Education, 43*(1-2), 109-123.

McNamara,L., RN, MSN,CCNS, CCRN. (2009). Translating traditional into creative through e - learning. *Nursing,* , 17.

Meira, L., & Peres, F. (2004). A dialogue-based approach for evaluating educational software. *Interacting with Computers, 16*(4), 615-633.

Moroz-Lapin, K. (2008). Enhancing creativity in HCI education. *New Aspects of Engineering Education; Mathematics and Computers in Science and Engineering; 5th IASME/WSEAS International Conference on Engineering Education,* Heraklion, GREECE. 232-237.

Mueller, L. (2007). SwissCHI's HCI education: A successful joint effort of practitioners and academia. *Interactions, 14*(5), 20.

Multimedia in education: The transition from primary school to secondary school - european parliament STOA 1997 Retrieved 2/03/2010, 2010, from http://www.pjb.co.uk/mmeduc.htm

Nielsen, Jakob, Rolf Molich, Heuristic Evaluation of User Interfaces, Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people, p.249-256, April 01-05, 1990, Seattle, Washington, United States

NRC: Backgrounder on the three mile island accident Retrieved 2/19/2010, 2010, from http://www.nrc.gov/reading-rm/doc-collections/fact-sheets/3mile-isle.html

Plimmer, B. (2006). A computer science HCI course. *People and Computers XIX - the Bigger Picture; B C S CONFERENCE SERIES; 19th Annual Conference of the British-Human-Computer-Interaction-Group,* Edinburgh, SCOTLAND. 185-199.

Rashty, D. (1999). Traditional learning vs. eLearning.

Robinson, C. (2009). Lessons on learning. *The Journal for Quality and Participation, 32*(1), 25.

Roschelle, J., Pea, R., Hoadley, C., Gordin, D., & Means, B. (2001). Changing how and what children learn in school with computer-based technologies.*10*(2), 76.

Sedighian, K. (1997). Challenge-driven learning: A model for Children's multimedia mathematics learning environments. *In ED-MEDIA 97: World Conference on Educational Multimedia and Hypermedia,*

Shen, J., & Eder, L. (2009). Intentions to use virtual worlds for education. *Journal of Information Systems Education, 20*(2), 225.

Squires, D. (1999). Proceedings of the 32nd hawaii international conference on system sciences - 1999. *Educational Software and Learning: Subversive use and Volatile Design,* 1.

Squires, D., & Preece, J. (1999). Predicting quality in educational software:: Evaluating for learning, usability and the synergy between them. *Interacting with Computers, 11*(5), 467-483.

Tscheligi, M., & Bernhaupt, R. (2005). HCI education at the ICT&S human-computer interaction as a cornerstone between technology and society. *Interactions, 12*(5), 34.

Ubisoft my stop smoking coach - DS - nintendo DS lite games Retrieved 2/19/2010, 2010, from http://www.retrevo.com/s/Ubisoft-My-Stop-Smoking-Coach---DS-Nintendo-DS-Lite-Games/id/23361bh926/t/1-2/

Vinogradov, S., Fisher, M., Warm, H., BA, Holland, C., BA, Kirshner, M., BA, & Pollock, B. (2009). The cognitive cost of anticholinergic burden: Decreased response to cognitive training in schizophrenia. *The American Journal of Psychiatry, 166*(9), 1055.

Williams, P. (2009). Bridging the gap. *E.Learning Age,* , 5.

Zaphiris, P., & Ang, C. S. (2007). HCI issues in computer games. *Interacting with Computers, 19*(2), 135-139.

5. Appendices

Appendix A: Maths Whizz Experiment Results

Game-Based Software Category/Criteria	Maths Whizz Max Score	Actual Score	Comments
Practicality of the Software			
1. Does the software provide adequate facilities for the user to address the core concepts of Mathematics?	4	2	Some of the harder games could provide additional tasks before starting the exercise
2. Is the information provided to the user relevant to the task and context?	4	4	Always in context and directly related to task
3. Are the functions provided to the user relevamt to the task and context?	4	4	Always in context and directly related to task
4. Are commands, actions and terminology followed consistently throughout the software?	4	3	Commands, actions and terminology mostly stick to a convention, however, this is not followed in some of the more advanced games
5. Is information presented in natural and logical order for this particular age group?	4	4	All relevant information is attention-grabbing and always followed by animations and sound
Category Total **Category Percentage**	**20**	**17** **85.00%**	
Simplicity of the Software			
6. Is each screen free of clutter? (not too much information displayed on one screen)	4	3	The screens don't appear to be over-cluttered, in some screens there could be less information/controls
7. Does each screen condense elements which are not central to the task at hand? (and free of unnecessary elements)	4	2	"credits" which are earned at the end of each exercise could be omitted in-game where they take up some valuable space, which can also add to clutter (above)
8. Are the most commonly used elements and options for a given task visible on each screen (with other options easily accessible)?	4	4	Always. Options and elements will only appear when necessary

9. Is the general design of the software "self-evident" (e.g. Would you expect the target age group to be able to navigate and locate functions without prior experience and detailed instructions with the software)?	4	3	The in-game exercises are very well directed and self-evident - the main menu a bit tricky at first
10. Is all "unwanted workload" automated? (e.g. Is it made faster, simpler, easier and more fun where possible?)	4	4	Everything is broken down to its simplest form and the games always seperate unnecessary workload whilst making the exercises fun
11. Are the instructions and data "straight to the point" (consolidated and summarised?) where expected, with this particular age group in mind?	4	3	Most instructions are concise, some are a little lengthy and pop up regularly
12. Is new information/topics introduced with aids to interpretation which would be meaningful to such a young age group?	4	4	New topics are always introduced at the start of a game, with an animated example
13. Does the software reduce the need to remember information, where possible? (e.g progress on calculations, displaying times tables where appropriate, etc)	4	4	Player's progess on calculations is updated synchronously throughout all exercises
14. With this particular age group in mind, are high-level tasks such as calculations and estimations supported by cognitive resources?	4	2	Cognitive resources such as calculators, symbols and note taking are available, however, these are available outside the challenges in a "whiteboard" module
Category Total **Category Percentage**	**36**	**29** **80.56%**	
Communicating the Subject			
15. Does the software provide appropriate, clear and timely feedback for the user so that they can see what is going on and know the result of their actions?	4	4	Always. User is kept well informed with feedback through pop-ups and several different progress bars
16. Is a structure within the software adopted? Are related items put together and unrelated elements set apart?	4	2	Many related items are set apart within the exercises, such as the two different progress elements, which are at either end of the screen. However, other related elements are geneally grouped together

17. Is the user aware of the sequence of the task in hand? For example, are they clearly informed of their progress at all times?(beginning, middle, end)	4	4	The user is well informed of this. This is done with the two progress bars, the top progess bar tells the user how many questions they have completed and have left to go, the other progress bar at the bottom informs them of their current percentage of correctness and time taken to complete
18. Help and documentation at hand? Are these instructions concise and do they focus properly on the task at hand?	4	4	Always. Each element or control within the interfaces have a small "i" (for info) at the top corner, and when the mouse pointer hovers over this, a sufficient explanation is displayed in a speech bubble
19. Does the environment keep track of the user's accomplishments? (scoreboards, graphs and charts depicting history of progress etc.)	4	4	There are scoreboards, charts, graphs, statistics and a medal system for each exercise. These scores can also be compared with the user's "buddies" scores if they have completed the same exercise
20. Does the interface properly communicate between pre and post submission of answers?	4	4	Yes. This is achieved by synchonous updates after the user performs an action. On-screen prompts are used intuitively, advising the user to submit once they have entered their answers
21. Where appropriate, are common metaphors, concepts and real-world conventions followed and understandable for the targed age group?	4	4	Always. These conventions are always depicted by examples and animations which fully and cleverly demonstrate the task at hand at a level which this target age group will understand
22. Are all mathematical symbols and terminology used limited to this target age group?	4	4	No mathematical symbols which are foreign to this age group are used
Category Total **Category Percentage**	**32**	**30** **93.75%**	
Efficiency of the Software			

23. Does the software accomodate the user's continuous advancement in knowledge and skill? (e.g. Less help popups once the user begins to reuse common elements within the software)	4	2	The software does promote this by eventually calculating a "brain age" (between the age range in key stage 2), however, the exercises themselves become repetitive and perhaps lengthy e.g spending 12 questions asking the same question in different ways and not increasing in difficulty
24. Are alternative shortcuts available to allow the user to access different areas of the software, once they become more experienced with frequent use?	4	3	Topics can be accessed through shortcuts if the user no longer desires to follow the dialy exercises in sequence. However, getting to back to the main menu, in some cases can only be achieved by clicking the same action several times as it passes previously explored screens (as opposed to directly going back to the menu)
25. To which extent does the environment allow the user to feel "in control" of their actions? (e.g initiating an action as opposed to responding to one)	4	3	Mostly
26. The environment should employ "forgiveness" where possible. Is the user prevented from making serious errors, wherever possible? Is there confirmation in place before the user performs a "destructive action"?	4	4	Always Supported
27. Error Recovery? Error messages displayed when there's disconnects or internal errors? Does it prompt to restart? (only deduct in the event of an error, where it's apparent there's no error recovery options)	4	4	Successfully recovered from an unexpected error
28. Does the environment promote "undo" and "redo" actions, where beneficial? The ability to reverse actions encourages the user to explore unfamiliar options.	4	3	Mainly, apart from a few tasks which didn't have confirmation
29. How well does the interface design relate to the target audience? (e.g attractive, bright, positive etc)	4	4	
Category Total **Category Percentage**	**28**	**23** **82.14%**	
Total Score	116	**99**	
Percentage Score		**85.34%**	

Appendix B: Topmarks Experiment Results

Picture/Animation Software Category/Criteria	Topmarks.co.uk Max Score	Actual Score	Comments
Practicality of the Software			
1. Does the software provide adequate facilities for the user to address the core concepts of Mathematics?	4	2	Yes, but in many cases inconstently
2. Is the information provided to the user relevant to the task and context?	4	3	The information is relevant to the task at hand, although sometimes the information is little unclear
3. Are the functions provided to the user relevamt to the task and context?	4	2	This varies greatly, in some areas the functions are very relative to the task, and in others the functions are barely identifiable, using foreign images/symbols which are not an obvious representation for such functions at this age group - this is perhaps because this learning resource pulls many different exercised for different resources together
4. Are commands, actions and terminology followed consistently throughout the software?	4	0	There is no consistence for either of these, again, due to many exercises belonging to different resources/developers
5. Is information presented in natural and logical order for this particular age group?	4	2	In some cases, the information is presented as such for the target age group. However, other exercises presented information in an order more appropriate for an older audience
Category Total **Category Percentage**	**20**	**9** **45.00%**	
Simplicity of the Software			
6. Is each screen free of clutter? (not too much information displayed on one screen)	4	2	Most screens aren't over-cluttered, although the screens do provide links and access to other subjects instead of isolating the current subjuect
7. Does each screen condense elements which are not central to the task at hand? (and free of unnecessary elements)	4	4	This seems to be achieved in all resources provided by this environment

8. Are the most commonly used elements and options for a given task visible on each screen (with other options easily accessible)?	4	3	Mostly. Some exercises don't provide these elements, however
9. Is the general design of the software "self-evident" (e.g. Would you expect the target age group to be able to navigate and locate functions without prior experience and detailed instructions with the software)?	4	3	Whilst the interface isn't as intuitive and simple for the target age group as others have achieved, it would be expected that the targed audience would have little or no trouble accessing all areas of the software with no prior experience
10. Is all "unwanted workload" automated? (e.g. Is it made faster, simpler, easier and more fun where possible?)	4	2	As these exercises are pictorial/animated in the most, not all workload is automated and does require some "manual" note taking in some of the exercises. Additionally - the design of the environment is a little standard and "webpage-esque", thus doesn't really promote a fun and inviting environment as strongly as it could
11. Are the instructions and data "straight to the point" (consolidated and summarised?) where expected, with this particular age group in mind?	4	3	Most of the instructions are straight to the point, but many of them fail to communicate effectively (ref. Communication section)
12. Is new information/topics introduced with aids to interpretation which would be meaningful to such a young age group?	4	4	All topics come with a visual example which is viewed before the user begins an exercise
13. Does the software reduce the need to remember information, where possible? (e.g progress on calculations, displaying times tables where appropriate, etc)	4	2	Some tasks do detail progress, other tasks either do not display progress or don't mention it's an ongoing task until the user decides to quit
14. With this particular age group in mind, are high-level tasks such as calculations and estimations supported by cognitive resources?	4	3	Most of the tasks offer calculators, notepads and fact sheets, where appropriate
Category Total **Category Percentage**	**36**	**26** **72.22%**	
Communicating the Subject			
15. Does the software provide appropriate, clear and timely feedback for the user so that they can see what is going on and know the result of their actions?	4	1	In most cases this was not achieved due to inconsistency and exercises belonging to different resources. However, the odd exercise did fully achieve this

16. Is a structure within the software adopted? Are related items put together and unrelated elements set apart?	4	1	The only structure adopted is within the main menu, where elements are grouped intuitively. Most of the exercises had it's own personal structures, thus there was no convention adopted
17. Is the user aware of the sequence of the task in hand? For example, are they clearly informed of their progress at all times?(beginning, middle, end)	4	2	This is achieved in some cases and not in others - the tasks where this was not achieved did not promote progress at all
18. Help and documentation at hand? Are these instructions concise and do they focus properly on the task at hand?	4	3	Help and documentation was available for navigating through the main menus and were clear and concise. As the exercises themselves were stand-alone, help and documentation was sometimes vague
19. Does the environment keep track of the user's accomplishments? (scoreboards, graphs and charts depicting history of progress etc.)	4	1	Some of the external exercises promoted by topmarks would retain some limited information about the user's progress. Most often, however, this was not the case
20. Does the interface properly communicate between pre and post submission of answers?	4	4	Surprisingly, although each exercise was developed externally, they had all adopted this well
21. Where appropriate, are common metaphors, concepts and real-world conventions followed and understandable for the targed age group?	4	2	This seemed to vary greatly between good and bad. Some extercises did commincate effectively, using common metaphors and conventions which would be understood by the target age group. Contrary to this, others seemed to explain things in a way which would only be interpreted by an older age group
22. Are all mathematical symbols and terminology used limited to this target age group?	4	3	Mainly. There were a few notable cases where cognitive resources, such as scientific calculators were not necessary to the age group and could have been provided in it's simplest form
Category Total **Category Percentage**	**32**	**17** **53.13%**	
Efficiency of the Software 23. Does the software accomodate the user's continuous advancement in knowledge and skill? (e.g. Less help popups once the user begins to reuse common elements within the software)	4	2	As topmark's learning modules link to several different external resources, no internal skill tracking or artificial intellegence could adopt this strategy - however, many of the exercises themselves used this method

24. Are alternative shortcuts available to allow the user to access different areas of the software, once they become more experienced with frequent use?	4	0	This was not achieved. Navigation would be constrained for this age group. Getting to exercises and back to the main menu is only achieved from pressing the "back" button on the browser after completing an exercise
25. To which extent does the environment allow the user to feel "in control" of their actions? (e.g initiating an action as opposed to responding to one)	4	2	Users often don't control or personally control the outcome of an action. In some cases this is achieved
26. The environment should employ "forgiveness" where possible. Is the user prevented from making serious errors, wherever possible? Is there confirmation in place before the user performs a "destructive action"?	4	2	In some cases, prevention of making mistakes (entereing invalid values was often prevented), but in other exercises, this was not prevented
27. Error Recovery? Error messages displayed when there's disconnects or internal errors? Does it prompt to restart? (only deduct in the event of an error, where it's apparent there's no error recovery options)	4	4	
28. Does the environment promote "undo" and "redo" actions, where beneficial? The ability to reverse actions encourages the user to explore unfamiliar options.	4	2	Again, this was "hit and miss" due to the variety in design and behavour of each external resource
29. How well does the interface design relate to the target audience? (e.g attractive, bright, positive etc)	4	2	The interface design was not entirely appealing to this age group, it lacked bright, colours and a fun, visual, simple design. However, the interface design was not completely redundant of these properties and the targed age group would still somewhat relate to the target audicence
Category Total **Category Percentage**	**28**	**14** **50.00%**	
Total Score	116	**66**	
Percentage Score		**56.90%**	

Appendix C: BBC Learning Zone Experiment Results

Video/Audio Software Category/Criteria	BBC Learning Zone Max Score	 Actual Score	Comments
Practicality of the Software			
1. Does the software provide adequate facilities for the user to address the core concepts of Mathematics?	4	4	This is delivered visually throughout the videos
2. Is the information provided to the user relevant to the task and context?	4	4	This is achieved consistently in every video
3. Are the functions provided to the user relevamt to the task and context?	4	1	There seems to be minimal functions, as all information is delivered through video
4. Are commands, actions and terminology followed consistently throughout the software?	4	4	These are all followed consistently
5. Is information presented in natural and logical order for this particular age group?	4	4	Written information is minimal, but all information delivered by the videos achieves this
Category Total **Category Percentage**	**20**	**17** **85.00%**	
Simplicity of the Software			
6. Is each screen free of clutter? (not too much information displayed on one screen)	4	3	The screens are generally clutter free apart from the links to other subjects appearing in the menus
7. Does each screen condense elements which are not central to the task at hand? (and free of unnecessary elements)	4	3	In the most. There are a few elements which could be condensed during viewing the exercise
8. Are the most commonly used elements and options for a given task visible on each screen (with other options easily accessible)?	4	2	This is inconsistently achieved

9. Is the general design of the software "self-evident" (e.g. Would you expect the target age group to be able to navigate and locate functions without prior experience and detailed instructions with the software)?	4	3	This environment adopts a "you-tube" style which in todays day and age appeals to all age ranges. Furthermore, the design is much simpler than the likes of youtube making navigation easier for this age group. Nonetheless, this is not completely self-evident, compared to some other packages in this experiment
10. Is all "unwanted workload" automated? (e.g. Is it made faster, simpler, easier and more fun where possible?)	4	2	The video element certainly makes learning more fun and simplifies many concepts. However, the child is expected to take their own notes, as internal resources do not provide this
11. Are the instructions and data "straight to the point" (consolidated and summarised?) where expected, with this particular age group in mind?	4	4	The nature of this enviroment being video/audio, this is acieved well
12. Is new information/topics introduced with aids to interpretation which would be meaningful to such a young age group?	4	4	This is always the case in every video. Every day items and objects are used to cleverly help this age group apply meaning to the exercise at hand
13. Does the software reduce the need to remember information, where possible? (e.g progress on calculations, displaying times tables where appropriate, etc)	4	1	This is only achieved through repetition and summary in the video. The user is otherwise left to make use of their own resources to permanently record this information
14. With this particular age group in mind, are high-level tasks such as calculations and estimations supported by cognitive resources?	4	0	No cognitive resources are provided
Category Total **Category Percentage**	**36**	**22** **61.11%**	
Communicating the Subject			

15. Does the software provide appropriate, clear and timely feedback for the user so that they can see what is going on and know the result of their actions?	4	2	This is moderately achieved through pauses or breaks within the video itself, however, other software within this experiment has demonstrated this better
16. Is a structure within the software adopted? Are related items put together and unrelated elements set apart?	4	3	This is mianly the case. Due to the simplicity of the software design, these elements are intuitively grouped. However, there does seem to be a slight lack of items and functions to warrant full marks
17. Is the user aware of the sequence of the task in hand? For example, are they clearly informed of their progress at all times?(beginning, middle, end)	4	4	This is achieved well through the clear and unmistakable progress of the videos
18. Help and documentation at hand? Are these instructions concise and do they focus properly on the task at hand?	4	1	There is no documentation or help at hand. There is some help within the videos itself but lack of help for navigating round this resource
19. Does the environment keep track of the user's accomplishments? (scoreboards, graphs and charts depicting history of progress etc.)	4	0	This feature does not exsist in this software
20. Does the interface properly communicate between pre and post submission of answers?	4	2	This is done through vocal cues within the video itself, however, a better strategy could have been adopted
21. Where appropriate, are common metaphors, concepts and real-world conventions followed and understandable for the targed age group?	4	4	This is always achieved well throughout all the video exercises
22. Are all mathematical symbols and terminology used limited to this target age group?	4	4	This is always achieved
Category Total **Category Percentage**	**32**	**20** **62.50%**	
Efficiency of the Software			

23. Does the software accomodate the user's continuous advancement in knowledge and skill? (e.g. Less help popups once the user begins to reuse common elements within the software)	4	3	The videos achieve this rather well and supports continuous advancement in knowledge. However, there is a lot of repetition which cannot be avoided in these videos
24. Are alternative shortcuts available to allow the user to access different areas of the software, once they become more experienced with frequent use?	4	2	There are different ways to access these video resources (list view, grid view etc), although there is otherwise not a lot in the way of shortcuts and variety of accessing and navigating these resources
25. To which extent does the environment allow the user to feel "in control" of their actions? (e.g initiating an action as opposed to responding to one)	4	2	The video exercises do encourage the user to write things down and join in, however, there are no resources within the environment itself to synchronously control the outcome of these exercises
26. The environment should employ "forgiveness" where possible. Is the user prevented from making serious errors, wherever possible? Is there confirmation in place before the user performs a "destructive action"?	4	2	There were a few videos which did not load and should have otherwise been removed from the catalogue - however there were error mesages on the video panel itself informing the user of this
27. Error Recovery? Error messages displayed when there's disconnects or internal errors? Does it prompt to restart? (only deduct in the event of an error, where it's apparent there's no error recovery options)	4	3	
28. Does the environment promote "undo" and "redo" actions, where beneficial? The ability to reverse actions encourages the user to explore unfamiliar options.	4	4	no need, due to the nature of the environment

29. How well does the interface design relate to the target audience? (e.g attractive, bright, positive etc)	4	3	Again, the simple "youtube style" works well and there would be no doubts that the target age group would be able to relate to this. Could be more bright, positive and attractive for this age group though
Category Total **Category Percentage**	**28**	**19** **67.86%**	
Total Score	116	**78**	
Percentage Score		**67.24%**	

Appendix D: Happychild Experiment Results

Text-Based Software Category/Criteria	happychild.org Max Score	 Actual Score	Comments
Practicality of the Software			
1. Does the software provide adequate facilities for the user to address the core concepts of Mathematics?	4	3	In the most this is the case, although the software offers printable worksheets which seem very heavily leaned to only a few topics
2. Is the information provided to the user relevant to the task and context?	4	4	Always, uses a text-based narrative style which cleverly relates to this age group
3. Are the functions provided to the user relevamt to the task and context?	4	2	There is a lack of functions available due to the maximum simplicity of this environment - however the functions which are available are relevant to the task and context
4. Are commands, actions and terminology followed consistently throughout the software?	4	3	This is mostly the case, although there are two different structures adopted within the software where this slightly varies between the two (e.g excercises vs worksheet interface)
5. Is information presented in natural and logical order for this particular age group?	4	4	This is achieved cleverly, again through the relevant narrative style
Category Total **Category Percentage**	**20**	**16** **80.00%**	
Simplicity of the Software			
6. Is each screen free of clutter? (not too much information displayed on one screen)	4	2	The screens arent cluttered by too many functions or elements, however, the scenarios within the exercises are very long and require a lot of scrolling. It would have been better if these were split into several pages, especially if the user wants to return to the exercise from a checkpoint
7. Does each screen condense elements which are not central to the task at hand? (and free of unnecessary elements)	4	2	The screens lack elements which could have been included, although there are no unneccessary elements present

8. Are the most commonly used elements and options for a given task visible on each screen (with other options easily accessible)?	4	2	Again, lacking in options and elements, however, the ones which are present are always present consistently through the available exercises
9. Is the general design of the software "self-evident" (e.g. Would you expect the target age group to be able to navigate and locate functions without prior experience and detailed instructions with the software)?	4	1	This type of interface comes accross rather dull and slightly difficult to navigate for such age group
10. Is all "unwanted workload" automated? (e.g. Is it made faster, simpler, easier and more fun where possible?)	4	2	Made simpler and easier, although not made faster and minimal in fun (although interesting, like reading a children's book)
11. Are the instructions and data "straight to the point" (consolidated and summarised?) where expected, with this particular age group in mind?	4	2	IT does take the "long route" explaining things although this still works quite well
12. Is new information/topics introduced with aids to interpretation which would be meaningful to such a young age group?	4	4	This is achieved consistently and in detail throughout all exercises
13. Does the software reduce the need to remember information, where possible? (e.g progress on calculations, displaying times tables where appropriate, etc)	4	2	This is achieved through the narrative quite well, although the user would have to use their own resources to fully benefit and learn from this software
14. With this particular age group in mind, are high-level tasks such as calculations and estimations supported by cognitive resources?	4	0	None available
Category Total **Category Percentage**	**36**	**17** **47.22%**	
Communicating the Subject			
15. Does the software provide appropriate, clear and timely feedback for the user so that they can see what is going on and know the result of their actions?	4	2	Again, this is achieved quite well through the narrative, although there is no direct feedback to the user, as they do not control the outcome
16. Is a structure within the software adopted? Are related items put together and unrelated elements set apart?	4	2	Lacks in items and functions, although this is acieved with that is there
17. Is the user aware of the sequence of the task in hand? For example, are they clearly informed of their progress at all times?(beginning, middle, end)	4	2	This is only avilable from looking at the scrollbar position within the browser

18. Help and documentation at hand? Are these instructions concise and do they focus properly on the task at hand?	4	1	There is a little help and advice for the user on the main page, however, that is all that was evident during evaluating this software
19. Does the environment keep track of the user's accomplishments? (scoreboards, graphs and charts depicting history of progress etc.)	4	0	This feature doesn't exist within this environment
20. Does the interface properly communicate between pre and post submission of answers?	4	0	The nature of the environment does not support this
21. Where appropriate, are common metaphors, concepts and real-world conventions followed and understandable for the targed age group?	4	4	This is achived very well through the textual examples
22. Are all mathematical symbols and terminology used limited to this target age group?	4	4	This is always achieved
Category Total **Category Percentage**	**32**	**15** **46.88%**	
Efficiency of the Software			
23. Does the software accomodate the user's continuous advancement in knowledge and skill? (e.g. Less help popups once the user begins to reuse common elements within the software)	4	3	Difficulty naturally increases throughout the process of the activity - however, the worksheets seem to maintain a single level of difficulty
24. Are alternative shortcuts available to allow the user to access different areas of the software, once they become more experienced with frequent use?	4	2	These are avilable although they are not too accessable as the user would have to do a lot of scrolling to reach these "shortcuts", which are at the very bottom of the screen
25. To which extent does the environment allow the user to feel "in control" of their actions? (e.g initiating an action as opposed to responding to one)	4	1	only worksheets allow control
26. The environment should employ "forgiveness" where possible. Is the user prevented from making serious errors, wherever possible? Is there confirmation in place before the user performs a "destructive action"?	4	0	The nature of the environment means that this is not supported
27. Error Recovery? Error messages displayed when there's disconnects or internal errors? Does it prompt to restart? (only deduct in the event of an error, where it's apparent there's no error recovery options)	4	4	too simple for this
28. Does the environment promote "undo" and "redo" actions, where beneficial? The ability to reverse actions encourages the user to explore unfamiliar options.	4	2	too limited for this so half marks

29. How well does the interface design relate to the target audience? (e.g attractive, bright, positive etc)	4	1	
Category Total **Category Percentage**	**28**	**13** **46.43%**	
Total Score	116	**61**	
Percentage Score		**52.59%**	

Appendix E: Creating the Evaluation Guidelines

Below are Nielsen's Heuristics and Powal's Cognitive Principles and outlines how all of the 29 guidelines were constructed. Some of these heuristics/principles were not used as were not relevant to this particular evaluation and one guideline can fall into many of these categories. For reference, the full description of each guideline can be found within sections 3.2.3.1 – 3.2.3.4 of this report. It should be noted that Guidelines 11, 19 and 23 were independently created from research and neither of the methods below.

Nielsen's Heuristics:

1. Visibility of system status
    - Guideline 15
    - Guideline 17
    - Guideline 20
2. Match between system and the real world
    - Guideline 1
    - Guideline 2
    - Guideline 3
    - Guideline 21
    - Guideline 22
3. User control and freedom
    - Guideline 9
4. Consistency and standards
    - Guideline 4
5. Error prevention
    - Guideline 26
    - Guideline 27
    - Guideline 28
6. Recognition rather than recall
    - Guideline 25
7. Flexibility and efficiency of use
    - Guideline 8
    - Guideline 14
    - Guideline 24
8. Aesthetic and minimalist design
    - Guideline 6
    - Guideline 9
    - Guideline 29
9. Help users recognize, diagnose, and recover from errors
    - Guideline 27
    - Guideline 28
10. Help and documentation
    - Guideline 18

Gerhardt-Powals' cognitive engineering principles:

1. Automate unwanted workload
    - Guideline 10
    - Guideline 13

2. Reduce uncertainty
    - Guideline 3
    - Guideline 9
3. Fuse data
4. Present new information with meaningful aids to interpretation
    - Guideline 5
    - Guideline 12
    - Guideline 21
    - Guideline 22
5. Use names that are conceptually related to function
    - Guideline 16
6. Limit data-driven tasks
7. Include in the displays only that information needed by the user at a given time
    - Guideline 6
    - Guideline 7
8. Provide multiple coding of data when appropriate
9. Practice judicious redundancy