Evaluating the HCI
of
Educational Software

Professor Plum

Computing BSc (Hons) 2005/06

Advisor: Dr Richard Foley

Submitted for the Degree of BSc in Computing

Factors

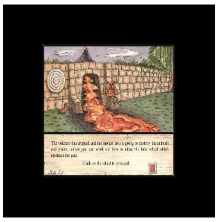
These are whole numbers which divide exactly into other numbers. For example factors of 12 are

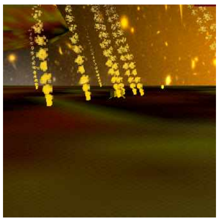
1, 2, 3, 4, 6, 12

Prime Numbers

These are numbers which only have two factors, 1 and itself. Prime numbers up to 20 are

2, 3, 5, 7, 11, 13, 17, 19.





Except where explicitly stated all work in this report, including the appendices, is my own:

Name: ________________________________ Date: _____________

ABSTRACT

This report details a study of the Usability of Children's Educational Software aimed at the primary seven sector of the Scottish Curriculum. This was created to reflect on the research already performed on the usability of educational software for children. Guidelines for the creation of educational software were gathered and used to produce a rating technique that aided in the assessment of the two mathematical edutainment packages. A set of evaluations that incorporated this technique were performed by both an expert and a group of students that had studied Human Computer Interaction. The varying personnel taking part in the evaluations was aimed to find more usability issues and show the quality in assessment technique between students and an expert. The evaluations discovered many usability problem areas in both pieces of software and gave a rating based on the percentage score of each area of children's usability and Nielsen's Heuristics. The comments and ratings from the evaluations were then compared to the views given by children of the targeted academic level after they had immersed with the software. The study proved the previous research correct that the introduction of interactive characters and situation problem solving makes an edutainment package more immersive for children. The research also discovered a considerable number of usability issues in the evaluated software. Ass an added feature this document also suggests possible scenarios and themes for future educational software games as suggested by children.

ACKNOWLEDGEMENTS

I am appreciative of Mark Armstrong's advising words before the project began which allowed me to hit the ground running when it did. I would like to thank Judy Robertson for performing the expert evaluation. I would also like to give my many thanks to my fellow students Michael Blair, Timothy McGraw, Greg Routledge, James Smith, David Bisset and Craig Cameron for taking part in my HCI evaluations. I am also grateful for the opportunity given to me by Mrs Luciana Verrechia that allowed me to have children evaluate my chosen software. I would lastly like to thank Dr Richard Foley for advising me and helping me to improve my writing skills and structuring throughout my study.

Contents

Table of Figures

Evaluating the HCI of Educational Software

Supervisor
Dr Richard Foley
(Glasgow Caledonian University)

Second Marker
Jim Paterson
(Glasgow Caledonian University)

1. Introduction

1.1 Problem Introduction

Norma Said remarked that “Multimedia has enormous potential but nobody is quite sure what works and what does not for children” [2]. This quote states the large problem in designing quality multimedia for children today.

A very large number of factors have to be considered including a child’s cognitive process and the aesthetics of the software when creating software for children. The defined area of this problem stated by Said that this project has researched into will be the usability of commercial educational software, the project specifically looks at educational software aimed at “Key Stage 3” [14] (a national educational level in the English curriculum or “Level D-E” in the Scottish Curriculum). A problem that educational software incorporates that other multimedia does not is that more research has been done on evaluating learning outcomes and not the usability [15]. Erlbaum states “If children can’t use the technology we’ve designed, it is our failure as designers”. Erlbaum also discusses the flaws [15] that designers have in assuming that children can use the software tools they are given, even if the tools aren’t apparent. Becker [3] believes that unless new tools are created to allow exploration at multiple levels of difficulty and accommodate diverse learning styles educational software will be just as limited as textbooks. These issues help justify researching into HCI as a problem area of educational software because the interface of this genre of software appears to be neglected whilst other design areas flourish.

Usability at this level of children’s education focuses on three main factors. The ‘Instructions’, the ‘Activities’ and the ‘Screen Layout’. Though there is research that has produced design information for the creation of each of these areas there has been no properly established evaluation method.

The issues stated above about children’s usability give good reason to research this topic and look into the HCI of educational software for children of any age.

1.2 Project Description

For the purpose of evaluation two pieces of mathematical software have been purchased that are aimed at children studying at the level “Key Stage 3”. Both pieces of software are similar to other available software on the market and are mathematical teaching and assessment packages. To judge the quality of the Human Computer Interface (HCI) it is necessary to perform HCI evaluations. A sequential set of objectives had been created so that the evaluations and the full project itself flowed properly.

Objectives

1. Research into Usability and Related Topics
2. Preparation of Evaluations
3. Evaluator Gathering
4. Disclosure Scotland Attainment
5. Performing Expert Evaluation
6. Performing Student Evaluations
7. Study of Results
8. Performing Child Evaluations
9. Study of Results
10. Conclusions

Research was conducted to source usability guidelines that were aimed at the software that was being assessed. Other related topics such as the child’s cognitive process and other usability studies were also researched so that a good understanding was reached. It was also important to research many forms of evaluation techniques so that the correct methods could be used to get the required results. Once the evaluation methods were decided upon it was necessary to generate documentation that would guide the adult evaluators through the usability areas to be assessed and what should be expected.

After the research had been done it had been concluded that there were three types of evaluation to be undertaken. For the expert evaluation Judy Robertson was gathered and 6 students were recruited from Glasgow Caledonian University who had basic experience with usability. The adult evaluations were performed with the technique called Guideline Analysis. This technique involved evaluating the software against the usability guidelines for it. The purpose of this was to identify the usability issues and then confirm them with the use of child evaluators with a Think-Aloud testing method. The Think-Aloud testing method aloud the children to give verbal feedback at certain points during the exploration of the mathematical software. Since there was the need for a group of children to take part in an evaluation, a source was used from St Aloysius junior school to attain children of the desired educational level. Due to the fact there was going to be children involved in an evaluation a Disclosure Scotland and Ethics Form was required to be processed before entering a school.

The expert evaluation involved reading the evaluation guide and rating system then evaluating each piece of software and rating each usability section. To justify that Judy Robertson was an expert in this field a short interview took place. The same type of evaluation was performed using 6 students studying Computing at Glasgow

Caledonian University. Each student had to satisfy his basic knowledge of human-computer interaction before each evaluation began.

After all the results had been amassed and analysed a simple evaluation technique for children was modified, this was based on the expert and group-based evaluations and was intended to attempt and confirm the original results found by guideline analysis. The group of children (19 in total) evaluated both pieces of software on there own whilst being asked a set of questions in specific sections based upon the previous results and design areas specified in other research [1]. This was not only to confirm the validity of the adult evaluations, but to also confirm the validity of the research that was the basis of the evaluations. After all the results were collected and studied the conclusions were then reached.

1.3 Hypothesis

A testable and accurate hypothesis must be used for a successful project. Due to the large amount of educational software on the market and the obvious flaws suggested by the initial research in children's commercial usability, a fair hypothesis to assume would be:

The Human Computer Interface for children's commercial educational software aimed at "Key Stage 3" (Level D-E in the Scottish Curriculum) has many usability problem issues.

Due to other research that was discovered during the literature review other sub-hypotheses that can be assumed are:

Character Intervention in educational software is desired by children at the educational level of "Key Stage 3".

The use of sound is a desirable attribute of children's mathematical software aimed at the educational level "Key Stage 3"

2. Literature Review

2.1 Introduction to Literature

The purpose of this Literature Review is to provide a foundation and an understanding before performing the evaluations and other associated activities. The Literature Review will look at past research in children's design areas and standard usability guidelines.

This chapter will also look at other areas that should be taken into consideration when an adult is evaluating a piece of children's software. This information is important when performing the methods and is intended to show why the interface of children's commercial educational software is not as good as it could be for Key Stage 3. Evaluation methods are discussed so that the correct techniques were used during the evaluations.

2.2 Usability and Children

2.2.1 HCI Design Areas for Children's Software

This section will discuss the usability guidelines for children's software and how they can be used to create an evaluation technique. Erlbaum states "Existing guidelines for children tend to be general and not age specific" [15]. The guidelines below are for children's software but are not accurately age specific (12 and under) nor has any been sourced, thus none are believed to exist.

There are HCI guidelines for the creation of software for children. These principles are relevant to the HCI of the level being evaluated as it encompasses the age group the software is aimed at (9-11). The Microsoft Corporation published research on Product Design Areas for children's educational software [1]. It stated 3 main design areas:

- **Activities**
- **Instructions**
- **Screen Layout**

These design areas are referred to in other research [15] [19] looked at which leads to the conclusion that it is credible. There has been no other research that has been discovered that offers an alternative to these design areas. There has also been no research discovered to suggest that these design areas are flawed. Further more the child's software expert Allison Druin refers to these design areas in books that she has published [1]. The three main design areas will now be discussed and why they are relevant to the research.

The "Activities" section must be interesting and challenging so that the children can be engaged in the software. If the difficulty keeps expanding it will give the children targets to aim for [1]. Every time a target or objective is met then the child should be rewarded in some shape or form. A good reward structure might give a child a piece of a jigsaw when a task is complete which they can build a picture with. A child will engage with the software if he/she likes [2] it so it would be logical to base software on real life interests or past times.

The "Instructions" section of the document deals with other areas of the software. The software should be created with the age group being considered, as it would be poor design if it was far too advanced or not advanced enough for a child [1]. The instructions must be easy to understand so that it is clear for a child in that age group to understand [1]. This may require an instruction to be repeated if audio or obvious if to the eye if pictorial. Good character interaction is desired so that the child can relate to it. If it is a piece of software which educates the child about science it may be relevant to have some sort of professor as a character. Controlled access to information is an example of good product designs as a child can only process a certain amount of information at the one time [15]. This could be done as bullet points or maybe a simplified version of a paragraph of information.

The Screen Layout refers to the icons and cursors on the interface. Icons should be meaningful and make sense. An example of a good meaningful icon would be a traffic sign that lets the user know when they can continue or a stopwatch that shows how long a task has taken. A good HCI should also have good cursor design, an example of that may be when an object on the screen can be dragged and dropped somewhere, the icon the user controls could change to a hand.

This research describes what the HCI of the software should have and will be an integral part of all the evaluations. It is concluded that this research is credible since it has been used by experts in this field such as Druin [1] and Erlbaum [15].

Each design area has a set of guidelines that should be adhered to for the creation of a good HCI for children. These points have been studied and have been used in the creation of an evaluation form (Appendix A). This document was used by the adult evaluators (Expert and Students) to assess the interface of the software provided.

2.2.3 Usability Differences Between Children and Adults

This section looks at the cognitive difference between adults and childrena and how it could effect usability evaluations. In a bulletin for acm.org [11] Allison Druin a leading expert in children's software mentions many key points that are relevant to the project. She indicates that by just looking at guidelines for the creation of a child's HCI is not enough. There is far more thought that has to be put in to it. This is because it is difficult for an adult to simulate a child's thinking process [11] [15]. It is also very challenging for an adult to understand a child's cognitive, social or physical capabilities. With all these thoughts on HCI design for children it is clear that age is a very important factor. This is why looking at the design areas when conducting evaluations is so important.

Miller and Vernon [16] believe that as a person develops from an infant to an adult their physical and cognitive abilities increase over time. This helps us to understand that children may not have the ability to understand how interfaces work. The Swiss psychologist Jean Piaget [17] analysed how children's cognition evolves and found out that children do not just lack knowledge and experience, but also fundamentally experience and understand the world differently from adults (this could be a problem if a child does not know that red means 'stop' and green means 'go' at a traffic light). This issue is echoed by Erickson [3] who explains that real-world metaphors from the adult world could be a concern if not properly understood.

When talking about the age group that the study is concerned with (primary 7's, 9-11) Schneider [18] explains, "we see children maturing on the brink of adult cognitive abilities. Though they cannot formulate hypothesis, and though abstract concepts such as ranges of numbers are often still difficult, they are able to group items and categorise." Erlbaum [15] puts the point across that this age group is playful enough to use relatively sophisticated software, but young enough to still enjoy using a playful piece of edutainment (educational but entertaining software).

This information will be used to create a document that the student evaluators of the software must read before performing the evaluation, as it is important that they have an outline of a child's cognitive ability and not judge it upon there own. If they judged the results with respect to adult usability the chances of the evaluation being accurate are unlikely. This is especially true when considering Nielsen's Heuristics as the heuristics are not specifically designed for children's software. It is also important that manufacturers of children's software in general that they are mindful of this information.

2.3 Evaluation Techniques

2.3.1 Children as Evaluators

This section will discuss the role as children as evaluators and why they are a very useful tool in this process. Research has shown that [1] the current generation is growing up fast and is far more knowledgeable about what to expect from software. If this is true according to her testing children that are aware of software would be credible evaluators.

Studies by the Microsoft Corporation [1] show that the best way to test children is by basing the testing on their age group. This information will be very useful when evaluating with children. The age group that was looked into is referred to as "Middle School". Children of this age are regarded as being easier to test than the younger children, as they tend to be more comfortable with unfamiliar adults and computers [1]. If this is true in the year 1997 then it must be even more true in the year 2006 with the large amount of GUI's that have been created for the likes of television and games systems, though no firm evidence has been found it could be construed that at present children should have more experience with interfaces than they did 9 years ago. This document also goes into good detail about how to approach the children and speak to them whilst using credible evaluation techniques. An exploratory study is described as studying children's ability to provide verbal comments in usability evaluation sessions [1]. The problem with testing children is that it is hard to know to what extent a child possesses the abilities that are required to do a good HCI evaluation. They concluded that children in general were very good in the task of evaluating usability in general but they had to have a grasp of computer basics. This was discussed with the teacher before the evaluations began to establish what children were suitable for the assessments. B.Kafai [13] argues that the current generation is is far more knowledgeable about what to expect from software, this backs up the theory that the children's familiarity with modern technology will help with my method.

Microsoft indicates that the usability of a product is closely related to children's enjoyment of it [1]. If this is true then it would be logical to test the product with children so it is possible to see how much they enjoy it. A point made by Jakob Nielsen implies that user testing can find many problems that a usability evaluator can not. This would suggest that there is a place in HCI evaluation of educational software from children if it is targeted at there age group. This research was not done on children however other sources [13] have indicated that the use of children as evaluators is very productive. This research gives justification for the use of children as evaluators of commercial edutainment in this age group.

2.3.2 Nielsen's Heuristics Evaluation

This section will look at the 10 points that make up Nielsen's heuristics and discuss how they are relevant within children's software.
Heuristic Evaluation is done by looking at an interface and judging what is good and bad about it [8] whilst looking at the current set of usability guidelines for all types of software. Molich and Nielsen conducted four experiments that had over 30 computer science students evaluate "online" and "offline" versions of the same system. It was found that the more evaluators that took part in the experiment the more faults were found. It was also found that more faults were found in a live system than evaluating screen dumps. This shows that by having a heuristic evaluation that there will be more problems found than using an offline system such as screen dumps. This concludes that having a heuristic evaluation is worthwhile. According to this research it is likely that the quality of the results could further improve by increasing the amount of people that perform the evaluation. This is why there has been the inclusion of an HCI evaluation to be done by computing students as well as an expert.

The further that HCI and design areas were studied the better the inspection should be if Nielsen's research is to be believed [8]. When conducting the Heuristic Evaluation the evaluator will have to have a background in the principles of the evaluation (Heuristics):

1. **Visibility of System Status**
2. **Match Between System and the Real World**
3. **User Control and Freedom**
4. **Consistency and Standards**
5. **Error Prevention**
6. **Recognition Rather than Recall**
7. **Flexibility and Efficiency of Use**
8. **Aesthetic and Minimalist Design**
9. **Help Users Recognize, Diagnose, and Recover from Errors**
10. **Help and Documentation**

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time. With children's software there should be a clear indication of what the computer is doing, an example of which might be informing the child that it might have to wait for the computer to complete a task.
The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order. This would appear to be one of the most important factors in children's HCI, as the child's cognitive process will have to be thoroughly considered.
Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Exit signs will have to be clearer for children which aren't familiar with certain icons.
Users should not have to wonder whether different words, situations, or actions mean the same thing. This is a relevant heuristic in every type of software. If a child clicks on a icon and it doesn't do the same thing every time it is very likely to get confused. An example of this would be an 'up' arrow that sometimes directs a character to go 'right'.
Even better than good error messages is a careful design that prevents a problem from occurring in the first place. Either eliminates error-prone conditions or checks for them and presents users with a confirmation option before they commit to the action. An example of this in children's software that would be good would be if a child was constantly making the same mathematic error. If this was spotted by the software it could explain or prevent the problem.
Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate. Since children have a different cognitive ability (section 2.3) to adults this is another area that must be explored.
Accelerators unseen by the novice user may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. This is very relevant for levels of difficulty in educational software as there is no point for a child who knows the 2 times table to have to prove to the software it can do it before moving to a higher times table.
Dialogues should not contain information that is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility. If a child is looking for an icon on the screen it would be bad practice to 'clutter' the screen with other unnecessary icons and shortcuts that would be never needed.
Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution. A child's understanding of language will be completely different from adults as stated earlier. This is an issue when explaining possible technical faults with the machine.
Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large. A help menu in educational software could be available when a child is having problems with a calculation. This would be a relevant thought with the software provided as it is mathematical software.

The Ten Usability Heuristics were used by students and an expert as part of the evaluating of the software. They evaluated all criteria set out in the heuristics. They are held in high regard worldwide and are used for Usability evaluating [4]. Jakob Nielsen's [8] research indicated that 42% of **major** usability problems were found by a single evaluator of the HCI whilst 32% of **minor** problems were found. This was not done on children's educational software however which might prove not to be the case in this genre.

The heuristics test the general functionality of software and are not aimed at children's software. The design areas are specifically aimed at children's software but it is evident that they give no functional design characteristics. This has concluded the need for a method which can incorporate both types of evaluation so that it covers the necessary areas of usability.

2.3.3 Usability Inspection Methods

This section will look at the options available and the previous research performed with regards to Usability Evaluation Methods. It will also justify the reasons for the choice of inspection methods.

Usability Evaluation Methods (UEM) are used to find usability faults in a human computer interface. Jakob Nielsen [8] created methods that are used today by established companies worldwide [4][27][28][29]:

Cognitive Walkthroughs
Formal Usability Inspections
Pluralistic Walkthroughs
Feature Inspection
Consistency Inspection
Standards Inspection

Cognitive Walkthroughs (Think-Aloud Testing) fully utilizes task scenarios to stress the user's cognitive process. This is advantageous as it puts the focus on the user and recognises user goals. The large disadvantage with this method is that it attempts to make the designer the user which can be difficult [7]. When the user is going to be children a serious consideration of the cognitive process of a child must be employed.

The rest of the UEM's are beneficial for different reasons but not of one single UEM listed above forms a good overall HCI evaluation for the genre that is being assessed, this is because of the complicated cognitive abilities of a child and the lack of guidelines for children's educational HCI. Due to the fact that children and adults will be used in assessments, there will have to use more than one method of evaluation (Formal, Empirical and Heuristic).

Nielsen and Molich [7] have identified four types of methods for evaluating a user interface:

- Formally
- Automatically
- Empirically
- Heuristically

The "Automatic" evaluation technique has been discarded due to its complicated nature and the lack of time to pursue such a highly technical and difficult topic. It is also completely infeasible apart from performing primitive checks on the interface due to complicated cognitive processes of the human brain [15]. A "Formal" analysis is done on the comparison of the guidelines of a product and the actual software. A "Heuristic" [7] analysis takes place when a piece of software is compared to the standards set by Nielsen's Heuristics. The "Empirical" way of evaluating software is highly regarded by Nielsen and Molich [7] because performing this test shows a great deal of usability errors. It involves receiving feedback from test subjects who the software is aimed at.

Companies that specialise in usability in all software processes have looked at other forms of evaluation that could be considered [4][27][28][29]. It is necessary to discuss these evaluation methods to look at how necessary they are for the inspection and what faults they should bring up if there is any in the software obtained. Constructive Interaction is a form of user testing used by usability specialists [4] where two users look at the product but only one interacts. This is very similar to performance testing but with the benefit of having another person's non-interactive view. This is useful as it produces an increased amount of verbal feedback. The problem with this is the more users testing the product the more money it costs (this does not apply to this project). If this project was too incorporate this method with the children for example, it may cause them to be distracted. This method is time consuming in comparison to the Heuristic Evaluation.

Als, Jensen and Skov when inspecting children's software with children have compared Think-Aloud and Constructive Interaction evaluation methods [15]. Constructive Interaction with another child produced the better results however the children used in this research were at a far higher educational level than the children being evaluated in this project. Markopolous and Bekker performed a similar experiment [21] but found little difference in the performance of each method. It could be suggested that the maturity and cognitive abilities of the children evaluated could have been a serious issue if they were allowed to interact and communicate with each other. This is why the Think-Aloud method was used in the child group-based evaluations.

Guidelines Analysis is an evaluation of software against the software domain guidelines. This is a very time intensive and possibly tedious evaluation [4]. The guidelines that have been sourced for children's software are not intensive in comparison to others (web-design for example [22]). It is unclear whether that is due to a lack of research in this topic or its complexity. Guidelines analysis will form part of the expert and student evaluations, as each guideline will be numerically evaluated against the software's ability to follow that guideline. The better the software follows that guideline the higher the rating. This is not a checklist evaluation, it is an evaluation based on the guidelines that should have been followed during the analysis, design and implementation of the software's life cycle. Since there is no specific inspection method for adults evaluating children's software, it would be effective for the adults to assess the software based on the software guidelines.

2.4 Overview of Literature

The objectives of the literature review were to source the necessary methods and information to assess the two mathematical applications by looking at previous research. Since the inspection methods above are generic and are not specifically attributed to children's software it makes the difficulty of performing an evaluation greater. This is not to say that Nielsen and the usability companies' methods are of no use but it does increase the requirement for an evaluation by children. This is due to the fact they are the users and also the age group the software is designed for. The literature review also discusses the cognitive differences between adults and children; this further verifying the need for child evaluators. The design areas aimed at children's usability are fundamentally important to both adult evaluation types and formed an integral part of the guideline analysis. The heuristics have also formed and integral part of the guideline analysis as they deal with important functionality that can be attributed to all types of software.

Whilst performing the initial research of the children's design areas certain key areas became evident. The need for character intervention and a productive use of sound in software packages would appear to be an integral part of good usability for this age group. This was found to be testable and not specifically designed for educational packages. With this being the an easily testable property of the software further answered hypotheses concluded were:

Character Intervention in educational software is a desired by children at the educational level of "Key Stage 3".

The use of sound is a desirable attribute of children's mathematical software aimed at the educational level "Key Stage 3".

The main hypothesis to be tested that is the main issue to be answered in this study came from various research publications by Said, Becker, Druin and Erlbaum. All these publications have stated that there is usability problems within children's software. Thus a testable hypothesis was:

The Human Computer Interface for children's commercial educational software aimed at "Key Stage 3" (Level D-E in the Scottish Curriculum) has many usability problem issues.

The methods used to clarify the hypotheses were very different with regards to age group. The adult evaluations (Expert and Student) were based upon Guidelines Analysis techniques whilst the Child Evaluations will incorporate Think-Aloud Testing.

3. Methods of Evaluation

3.1 Introduction to Methods

This section will describe the various methods that will be used to evaluate the software provided. The description of the methods will include the description of use, the ways in which each method will be carried out and what results were expected. It will also show the reasons the methods and the software were chosen.

Two pieces of software were selected based on the curricular activities of the school taking part and the level of which the pupils are studying. The software catalogue [23] aimed at the Scottish Curriculum was attained and a comparison was made on the price and likeness each piece of software had with each other. A comparison of screen shots and software descriptions were made to ascertain whether the software could be purchased at a low price but not be incomparable to other commercial educational software. Two pieces of software were purchased ('Success' and 'Letts') that have a usability likeness to resemble the HCI and style of more expensive pieces of educational software.

Due to the fact both pieces of software have similar names the company names (Letts and Success) will be used so not to cause confusion. A judgement was made based on the screen dumps and information that was given about them that the user interfaces were quite different but based on mathematics in Scotland at the same educational level (key stage 3). Both pieces of software were purchased at the same personal computer store in Glasgow but were created by different companies. This obviously proves that the software is commercial. After an initial review of both applications it has been ascertained that Success has a storyline along with the maths problems whilst Letts does not. This point proves that Success should be more engaging for a child than Letts according to research originally done by Norma Said [2] that looks at children engaging multimedia.

Once the information was researched to evaluate the HCI of this software the methods were decided upon to use:

- Think Aloud User Evaluations (Children's Group Based Evaluation)
- Expert Based Evaluation (Expert Guideline Analysis)
- Student Group Based Evaluation (Students Guideline Analysis)

The adult evaluations with the students and the expert were used to define what areas of usability would be tested with the children. This would confirm the results or possibly bring up new issues not already discovered. The further sections of this report details what research has been done on the techniques and what methods have been used to evaluate the usability of the software presented.

3.2 Software Description

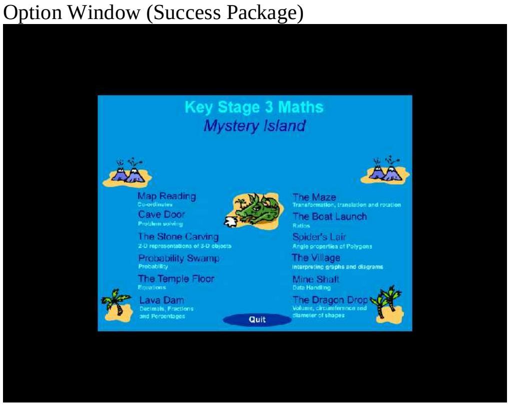
Taylor [24] has divided educational software into 3 categories:

- The computer as a tutor
- The computer as a tool
- The computer as a tutee

Both pieces of software being evaluated work as a "tutor" as they describe the problems and ask the child to answer questions. This is not an uncommon practice, as it would appear to be trying to simulate a teacher. Skinner [25] describes this practice as "Computer Aided Instruction" (CAI).

3.2.1 Success

Option Window (Success Package)



Key Stage 3 Maths

Mystery Island

Map Reading
Co-ordinates

Cave Door
Problem solving

The Stone Carving
2-D representations of 3-D objects

Probability Swamp
Probability

The Temple Floor
Equations

Lava Dam
Decimals, Fractions and Percentages

The Maze
Transformation, translation and rotation

The Boat Launch
Ratios

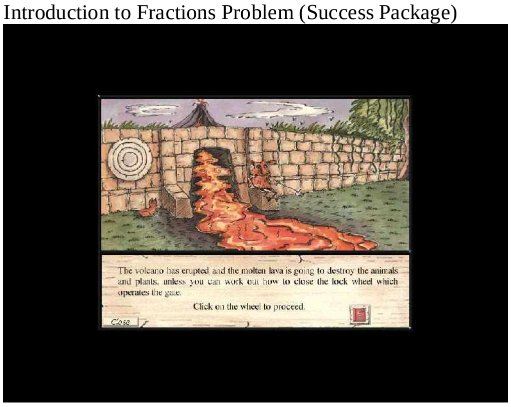
Spider's Lair
Angle properties of Polygons

The Village
Interpreting graphs and diagrams

Mine Shaft
Data Handling

The Dragon Drop
Volume, circumference and diameter of shapes

Introduction to Fractions Problem (Success Package)



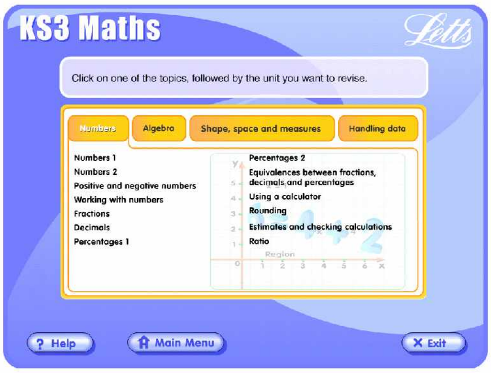
The volcano has erupted and the molten lava is going to destroy the animals and plants, unless you can work out how to close the lock wheel which operates the gate.

Click on the wheel to proceed.

The 'Success' package that has been used in the evaluations has scenario-based problems within it. The child is not just solving a fraction or a decimal problem it is turning a wheel or pulling a lever. An example of this would be when the user has to turn a wheel to move the fractions from highest to lowest in the correct order. The user is then allowed to move on to another problem but there is no reward system. This problem situation keeps the user more engaged in the task according to Said [4].

3.2.2 Letts

Option Window (Letts Package)

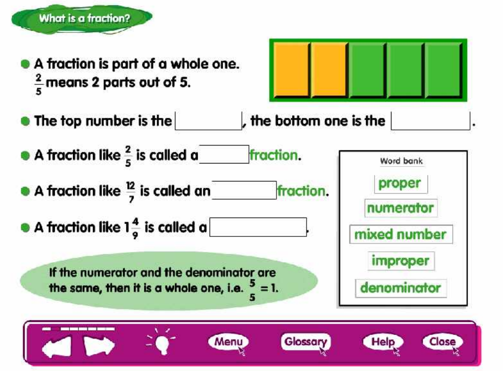


KS3 Maths

Click on one of the topics, followed by the unit you want to revise.

Numbers 1
Numbers 2
Positive and negative numbers
Working with numbers
Fractions
Decimals
Percentages 1
Percentages 2
Equivalences between fractions, decimals and percentages
Using a calculator
Rounding
Estimates and checking calculations
Ratio

Introduction to Fractions Problem (Letts Package)



What is a fraction?

- A fraction is part of a whole one. $\frac{2}{5}$ means 2 parts out of 5.
- The top number is the ______, the bottom one is the ______.
- A fraction like $\frac{2}{5}$ is called a ______ fraction.
- A fraction like $\frac{12}{7}$ is called an ______ fraction.
- A fraction like $1\frac{4}{9}$ is called a ______.

If the numerator and the denominator are the same, then it is a whole one, i.e. $\frac{5}{5} = 1$.

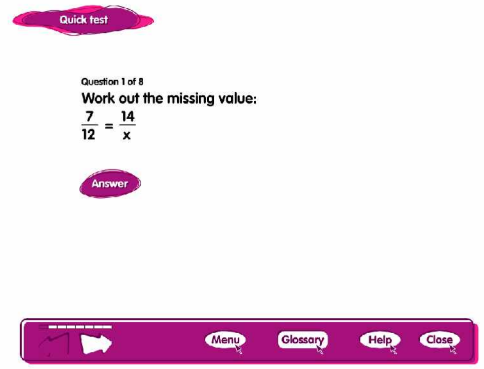
Word bank
proper
numerator
mixed number
improper
denominator

The ‘Letts’ package that has been evaluated does not have any scenario-based problems and produces mathematical problems on selection. An example of this would be when the user has to look at questions and write down answers. There is no reward system in place.

3.3 Evaluated Scenarios

All assessments performed involved the ‘Fractions’ problems from both pieces of software. Though both problems were very different, the educational levels were the same. It was important to maintain the type of problems and educational levels the same so that there could be no argument over the results. The fact that the scenarios were different meant that the design areas and the heuristics would be different (this includes the evaluation of instructions and activities). Both scenarios could be performed iteratively until the evaluator was satisfied with their opinion. This was not the case with the child assessments as they were only given two attempts due to the amount of evaluators that were using the software and the time limit.

Letts Fractions Problems

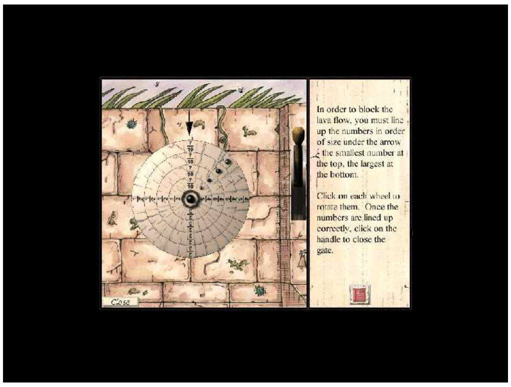

Quick test

Question 1 of 8

Work out the missing value:

$\frac{7}{12} = \frac{14}{x}$

Success Fraction Problems


In order to block the lava flow, you must line up the numbers in order of size under the arrow - the smallest number at the top, the largest at the bottom.

Click on each wheel to rotate them. Once the numbers are lined up correctly, click on the handle to close the gate.

3.4 Guideline Analysis Form

A Human Computer Interface Test Guide was constructed to advise and aid with all adult evaluations (Both Expert and Student). The test guide allowed the evaluator to enter ratings for each piece of the usability area in question whether this was Activities, Instructions, etc. The test guide not only allows input for the rating of each usability area but also detailed what exactly is too be evaluated. The knowledge each adult evaluator has of usability should have been enough for them to make a quality judgement.

The guide in this project was created based on Nielsen’s heuristics and the usability guidelines that were researched by the Microsoft Corporation. The guide (Appendix A) was devised based on a similar rating system used by Baekdal [26] for the evaluation of adult usability. The system was devised by creating a table that included all areas that were to be assessed and the sub areas that were being rated to give an overall rating for that area. Further information was added that informed the evaluator how each area was judged. A comments area was added for each area so that the reason for each rating could be explained. This system was chosen because it was good for achieving a numerical judgement of the interfaces whilst others only allowed comments.

The rating system had each guideline and heuristic evaluated out of 10. The higher the rating the better the software follows that guideline or heuristic. The overall rating will be used to give the interface a percentage as a score. Obviously the higher the percentage the more highly regarded the interface is. There was a need for a comments section so that the usability problems for that section could be identified which would help when performing the evaluation with children. This would assist in the documentation used in the children’s group-based evaluation.

3.5 Expert-Based Evaluation

An expert review is one of the primary evaluation strategies used in both formative (How can this multimedia program be improved?) and summative (What is the effectiveness and worth of this multimedia program?) evaluations [15]. It is often a good idea to provide experts with some sort of instrument as a guide [15].

An expert from Glasgow Caledonian University was willing to help with this project. Judy Robertson has extensive knowledge of both HCI and children. Her own research of evaluating with children will be a massive advantage when coming to perform the Expert Based Evaluation.

This evaluation involved test scenarios of multiple tasks from both pieces of software. These tasks will be consistent over all of the evaluations so as not to give an unfair and inaccurate judgement of the interfaces. This included the fraction problems that are consistent over all problems. The expert looked through the evaluation form provided before the evaluation began so that she was mindful of what factors to look for. The evaluation was iterative so that ratings of the interface and comments could be added constantly. Further help was provided during the evaluation to answer any points that were not made clear. There was no constraint on what type of screen or computer that was used as long as the resolution is 1024 by 768; this is for reasons of consistency. There was no time limit but initial tests with a student had taken nearly one hour for both pieces of software. The evaluation was timed.

It was explained to the expert that she would receive no benefits from the evaluation.

3.6 Group-Based Evaluation (Student)

To perform the Group-Based Evaluation using students the computing students first had to be acquired. Material has been read that states that 5-7 evaluators is quite sufficient [15]. 6 evaluators were acquired from Computing at Glasgow Caledonian University.

This evaluation involved test scenarios of certain tasks from both pieces of software. These tasks were consistent over all of the evaluations so as not to give an unfair and inaccurate judgement of the interfaces. The students looked through the evaluation form provided before the evaluation began so that they were mindful of what factors to look for. It was also necessary to have them look at a document describing the difference in cognitive abilities between children and adults; this made them aware of more possible usability errors (Appendix I). The evaluation was iterative so it was easier to change a rating of any interface and the comments. Further help was provided during the evaluation to answer any points that were not made clear. There were no constraints on what type of screen or computer that was to be used as long as the resolution was 1024 by 768; this is for reasons of consistency. There was no time limit but previous tests had shown that the full evaluation would take around and hour.

It was explained that the student may receive no benefits from the evaluation.

3.7 Group-Based Evaluation (Children)

As discussed in the Literature Review, it is concluded that it was necessary to have an HCI evaluation that is performed by children due to the fact they have a different cognitive thinking process from adults. The information below describes similar evaluations using children.

Think-Aloud User Testing (Cognitive Walkthrough) involves a user interacting with a piece of software through scenarios whilst giving verbal feedback [6] [7]. This was the form that was adopted when evaluating the HCI of the software with children. M.Bekker (2003) uses a technique he calls “Active Intervention” [3] when having children evaluate the usability of a toy. With this method the child evaluates a toy and answers questions at certain points of the evaluation. This has similarities to Think-Aloud evaluations as they both provide verbal feedback. Thus the quality of verbal feedback by children is clearly appropriate as a form of evaluation.

A primary school in Glasgow was identified that was willing to perform the evaluation. The full primary 7 class from St Aloysius junior school was used in the evaluations. The evaluations took place in the school’s IT room during the month of May 2006. The questions and the stages of them being asked were decided after the expert and student evaluations were analysed, this was so that it was possible to test to see if the children find usability problems in the same areas as the adults. This evaluation would also help confirm that the adult evaluations are a worthy form of usability inspection. The amounts of usability problems were noted for each application. It also gave the ability to prompt them on certain points made by the adults that would not been possible without the adult evaluations. The children also performed the same scenarios as the adults for consistency reasons.

During the evaluation of each child’s experience with the software the same technique was used with appropriate questions about the HCI. These questions were at a mixture of stages when problems were found with the other usability methods and when no errors are found. An example of which would be when the instructions appeared. This method is good due to the fact it can elicit verbal comments from children at various stages, this assesses whether the children discovered problem issues in different stages that adults did not.

A possible option was to have a user's interaction with the product videotaped and analyzed to improve the product. This was a problem because permission and transport would have had to been organised to take the children out of school. The ability of being able to iteratively evaluate the same observations many times has its obvious advantages however there is a disagreement from sources about the use of videos. Druin does not prefer to use video cameras when evaluating children, as she believes that the children perform for the camera [19]. This was a factor to consider when looking at what evaluation method to use. Not only would transport of the children or the video equipment be difficult but the children may perform out of character when not in a school environment in front of a camera. Other children on the other hand can freeze when they know they are being filmed, again this effects the results. This is argued by Goldman-Segall [20] due to the fact that he believes it is necessary as an evaluator might miss a key element of the evaluation. It also relieves stress from the evaluator so that person does not have to worry about missing any necessary issues. This information has given reason to go ahead with the 'Think Aloud Testing Strategy' as there would be no need to look out for emotion from a child and worry about these possible variables during an evaluation that involves children.

To be allowed to enter a schools premise it has been necessary to obtain a form called a disclosure. To apply for this it is a requirement to register with an organisation called "Disclosure Scotland". This document (Appendix G) discloses criminal convictions and up and coming trials. An Ethics Form was also completed so that the project conforms to regulations at Glasgow Caledonian University.

3.8 Overview of Methods

The Guideline Analysis evaluations performed by the Expert and Students concluded results that were used to aid in the inventing of the evaluation with the primary 7 class. Each child from the class will perform a small Think-Aloud assessment that produced conclusions that can confirm or argue the results of the adult assessments.

The adult ratings produced numerical ratings for the interfaces. The Think-Aloud Testing performed with the children produced no numerical values. The type of results that were discovered revealed the quality of the interfaces from each piece of software. The results were collected and conclusions were produced for each interface.

4. Results of Evaluations

4.1 Introduction to Results

In this chapter of the report it will show all aspects of the results that were founded from the evaluation of the educational software packages. The first evaluation that was performed was by the HCI expert Judy Robertson. After the expert evaluation had been performed the student evaluations then took place and the results gathered. Both the results of the expert and student evaluations were studied and the areas of concern were noted. This was so that the results could be confirmed or otherwise by the Children's group-based evaluation.

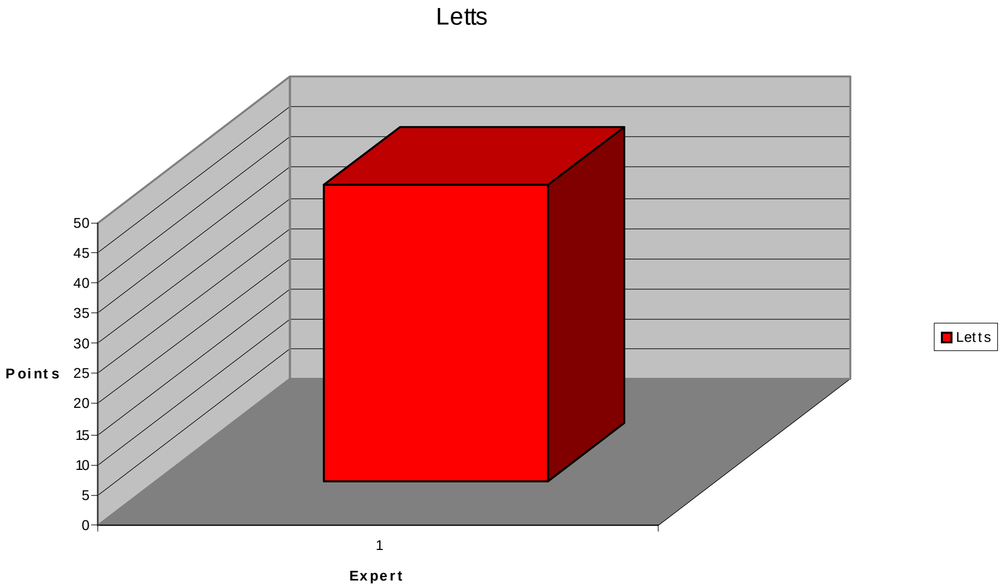
To regard one expert having a worse evaluation technique than a group of students would be a poor use of her skills and knowledge. Since there has been a considerable difference in the results between the expert and the students it is necessary to look at the results in different categories.

4.2 Results of Expert Evaluation

This section will look at the expert's evaluations of both pieces of software. It will delve further into the points total for the evaluations and the areas that make up the evaluation. There will also be a brief summary of the comments from the expert.

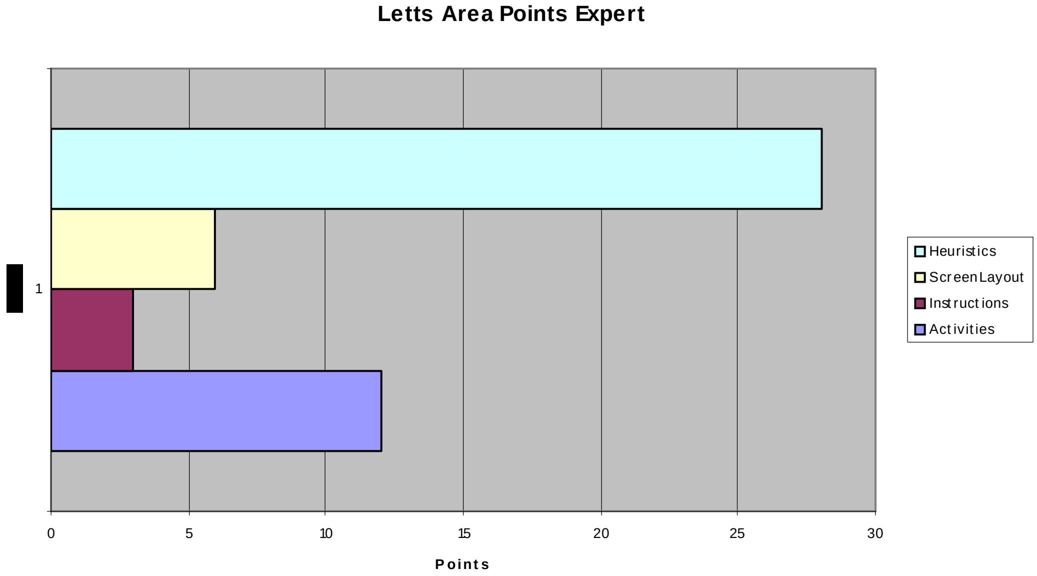
4.2.1 Letts Evaluation

Expert Evaluation Total (Letts)



The Letts software package received the lowest points total over all of the evaluations by the expert (49 points). The expert was very unimpressed by this package as it performed very poorly in all areas.

Expert Evaluation Areas (Letts)



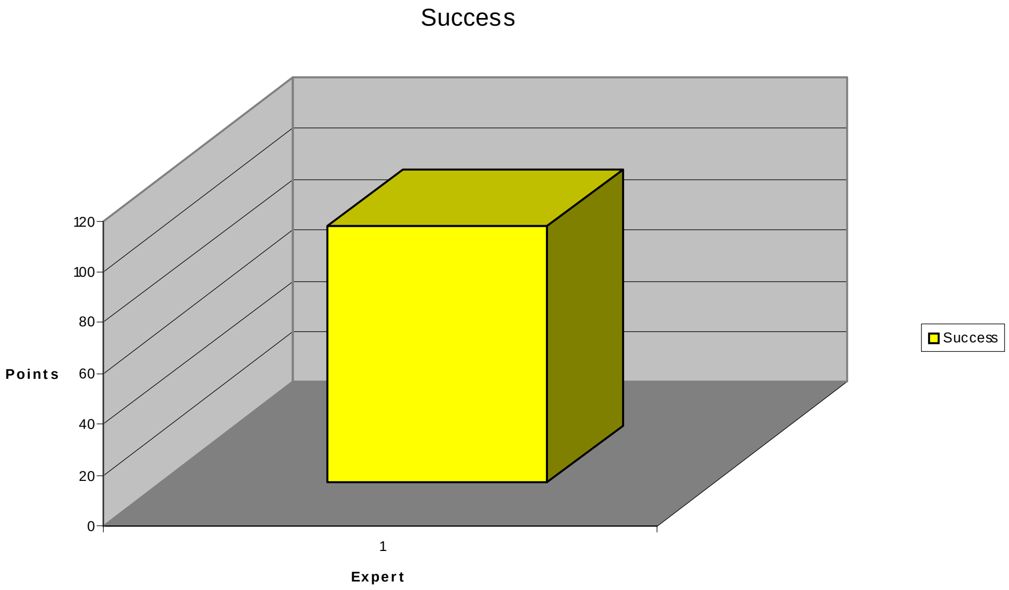
The Heuristic evaluation is a general form of evaluation that assesses the functionality and basics of usability. The heuristic area of the evaluation did not score highly with the Letts software package. A mere 28 points from 100 is not a score that was expected. It is a worrying score for a piece of "off the shelf software". The system status of the software was described as 'confusing'. If it were confusing for an adult then this confusion would only be multiplied if given to a child. The confusion is also found in other areas, as there is poor user control and freedom. It is also stated that the software "almost forces the user into errors", again another problem associated with the confusion element. Other areas it performed badly in with regards to Nielsen's Heuristics are Flexibility, Aesthetics and Help Functions. The software does not cater for familiar users and does not explain errors that the user has made. The screen is very busy which makes it difficult to find help functions. The only positive that has been taken from the heuristic overview is that it is easy to recognise real world objects.

The evaluation areas based on the fundamentals of usability design areas for children were worst than the results for the Heuristics. The Activities area of the evaluation scored 12 out of 50 and was described as 'confusing' and 'not very interesting'. The Literature Review chapter stated that keeping the child interested in the software was extremely important. This software failed badly in this sector.

The Instructions area of the evaluation scored 3 out of 40. This was the worst score out of all the point evaluations. The confusion that has been echoed all over the expert evaluation of the Letts product and is at pinnacle in the instructions area. The word "appalling" was the choice of my expert when attempting to comprehend the instructions.

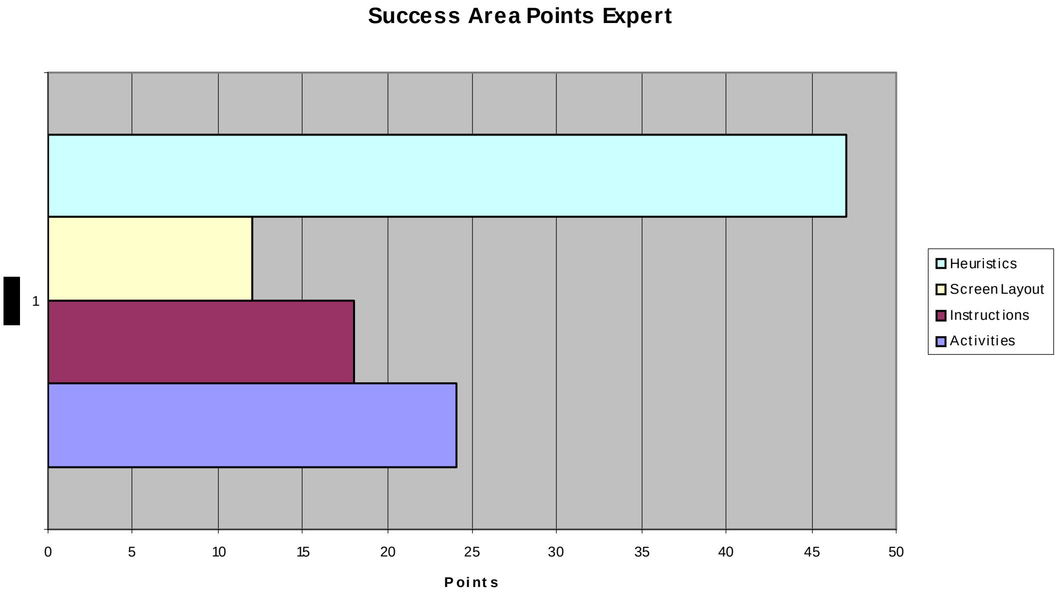
The Screen Layout area of the evaluation scored 6 out of 30. The screen icons were age specific to an average level but a cluttered screen gave the product a very poor rating in this area as well. If there is a cluttered screen it makes it difficult to pick the important icons and labels out and also makes it more time consuming in comparison to few icons. Overall the usability of the Letts package is very poor and lacks quality in nearly all areas of usability that it has been assessed on.

4.2.2 Success Evaluation



The Success software package received 101 points in total by the expert. The expert Judy Robertson liked the tasks and the activities that were to be completed. The reasoning behind this is due to the implementation of problem based scenarios that have to be solved with the use of mathematics. This is different from the Letts package which uses simple mathematical problems.

Expert Evaluation Areas (Success)



The Heuristic evaluation is a general form of evaluation that evaluates the functionality and basics of usability. The Success software package scored midway in the heuristics section. The system status of the software did not give progress reports on tasks and scores. The users control of the software meant that by moving from one screen to another would delete any previous work already done. Drag and drop functions were not consistent and errors could still be easily made. Another issue that was brought up was the lack of flexibility of level difficulty. The help function was easily obtained but only offered educational help and did not offer help with usability understanding. There were positives to be taken from this evaluation though. It is easy to recognise real world objects and the help icon is consistent throughout. The design was minimalist and not confusing with good real world matches in the software.

The evaluation areas based on the fundamentals of usability design areas for children were similar to the results for the Heuristics. The Activities area of the evaluation scored 24 out of 50 and was described as ‘fun’. The Literature Review chapter stated that keeping the child interested in the software was extremely important. This software was liked which can only be a positive.

The Instructions area of the evaluation scored 18 out of 40. The instructions were very mixed in quality. Some were very clear whilst others were confusing, however they were in an age appropriate format. This would have scored far higher had the implementation of a character been present.

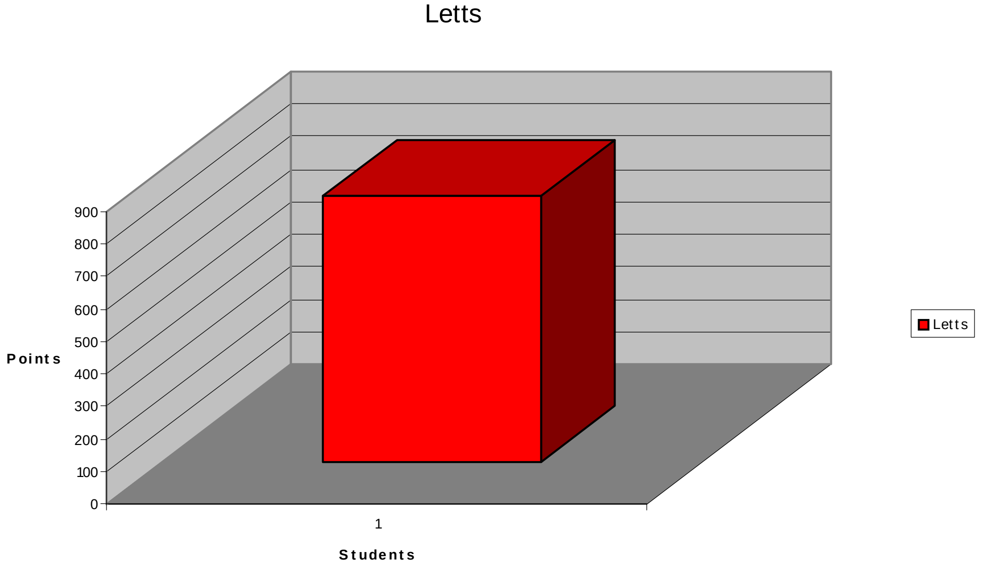
The Screen Layout area of the evaluation scored 12 out of 30. The screen icons were age specific to an high level but cursor design was poor.

An overall difference of 52 points in the expert evaluation between the two packages would seem to suggest that the Success package has a far better HCI.

4.3 Results of Student Evaluations

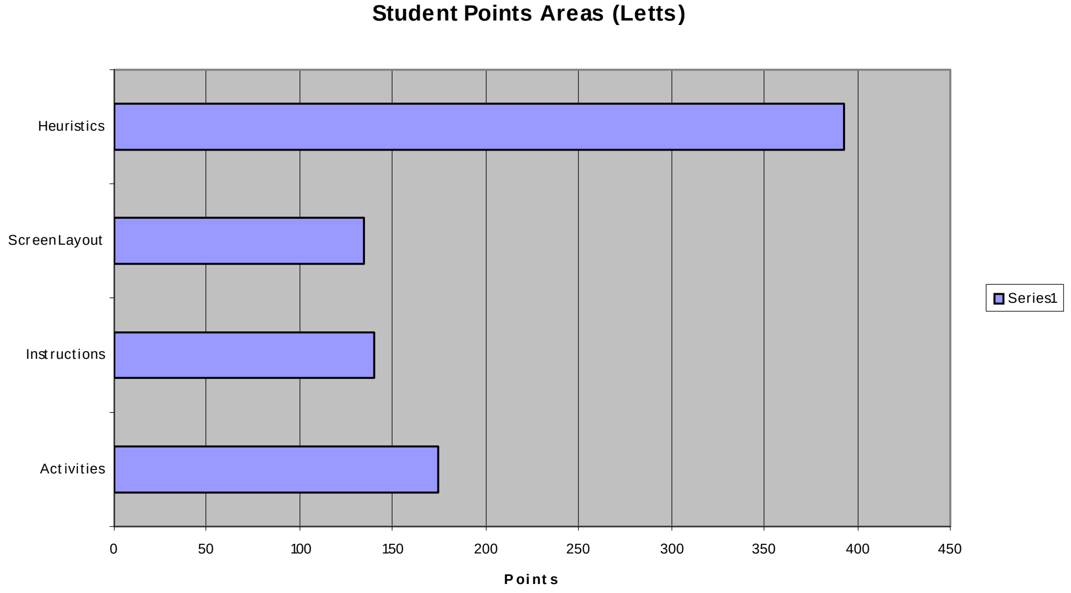
This section will look at the students’ evaluations of both pieces of software. It will delve further into the points total for the evaluation and the areas that make up the evaluation. There will also be a brief summary of the comments from the students. Only the clear and relevant comments are shown.

4.3.1 Letts Evaluation



The Letts software package received 842 points in total by the 6 students. This gives a mean value of 140.34 points per student.

Student Evaluation Areas (Letts)



The Heuristic evaluation is a general form of evaluation that evaluates the functionality and basics of usability. The heuristic area of the evaluation scored midway with the Letts software package. 393 points from 600 gave a mean value of 65 out of 100 per student.

The Activities area of the evaluation scored 174 out of 300 which gave a mean value of 29 out of 50 per student.

The Instructions area of the evaluation scored 140 out of 240 which gave a mean value of 23.34 out of 40 per student.

The Screen Layout area of the evaluation scored 135 out of 180, which gave a mean value of 22.5 out of 30 per student.

The students produced a lot of written feedback during the evaluations. There were mixed views on nearly every area though the need for character intervention was stated clearly in all areas. This was evident due to the lack of points in this area. The project will mention only the valid or constant comments that have been produced. Thus eliminating the confusing and inconsistent comments.

The comments associated with the Activities area of the Letts package were consistent. "The activities do increase in complexity but are not interesting for the child user." All evaluations done by the student described the activities as "boring" or "uninteresting".

The area with the most variation in comments was the Instructions section. Some evaluators described the instructions as "short" whilst others described them as "long". The evaluators generally poorly described this area though the ratings would suggest that the instructions were good. One evaluator commented that "there was no unnecessary information". It is hard to understand why the points tally is so 'high' in the instructions section as there is no "explanation of mathematical errors."

The Screen Layout area was regarded as being consistent by all evaluators but was also said to be "too busy".

With regards to Nielsen's Heuristics, the Letts package is given many comments that do not reflect good usability.

"The system's alert functions are poor."

"There is no confirmation request when exiting."

"No accelerators."

"Couldn't exit."

"Help didn't work."

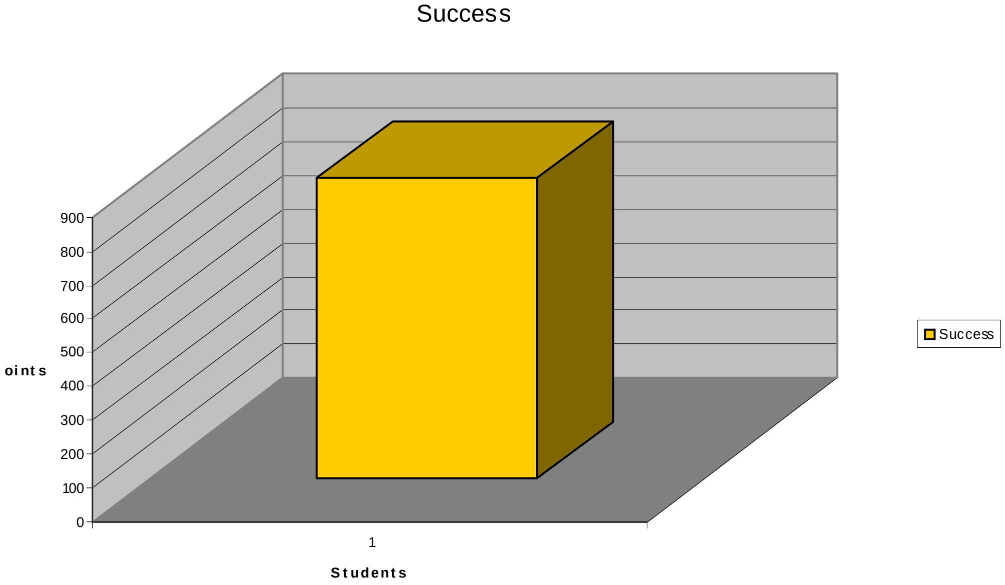
"Navigation is poor."

"Help is poor little information."

"Exit button did not work."

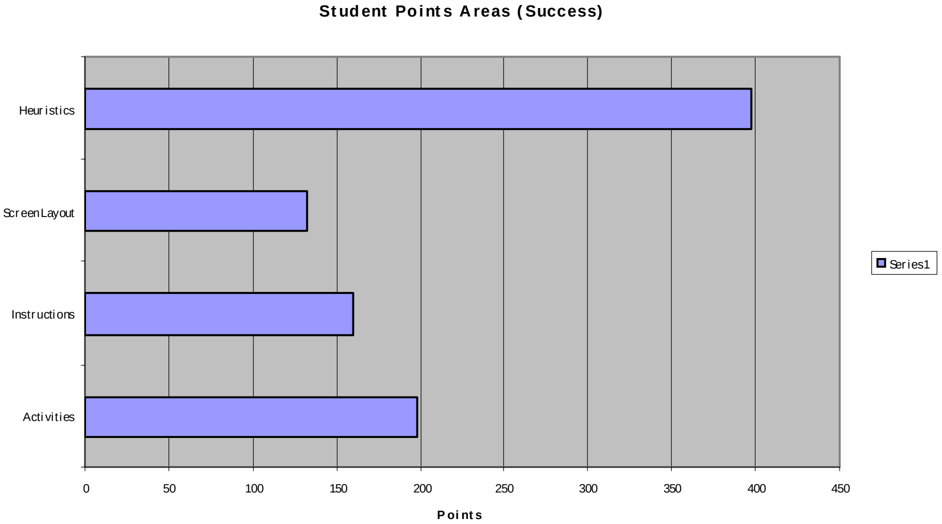
With so many unwanted comments for the interface designer of this product, it is very difficult to comprehend where the high scores have arrived in the student evaluations.

4.3.2 Success Evaluation



The Success software package received 887 points in total by the 6 students. This gives a mean value of 147.84 points per student.

Student Evaluation Areas (Success)



The students' views of this software are consistent in most areas of this evaluation. The storyline and problem based scenarios within the game were a popular interest of the evaluators.

The Instructions area of the Success package was generally consistent when being evaluated. "Instructions always displayed" was a phrase found several times in the HCI test guides with no real negative comments being elicited.

"Instructions are clear in most cases but are poor when explaining icon dragging."

Success performed well in the area of Activities in comparison to the Letts package due to the storyline and different real world problems.

“Activities are good for a child but do not increase in difficulty or offer the option to.”

The main problem that is remarked regularly in the assessment forms about the activities is the missing difficulty level; this appears to reduce the rating of this area by a large quantity.

The Screen Layout was said to be “clearly labelled” but certain icons were “not recognisable” with no rollover function to explain their use. The screen is also said to be “busy” but “neat”.

Heuristic comments once again do not reflect the rating system. The comments appear to be mixed yet the rating is still high. It has been a very puzzling area of the student evaluations as there appears to be no explanation for this other than the evaluator error.

The Heuristic evaluation is a general form of evaluation that evaluates the functionality and basics of usability. The heuristic area of the evaluation scored midway with the Success software package. 397 points from 600 gave a mean value of 66 out of 100 per student.

The Activities area of the evaluation scored 198 out of 300, which gave a mean value of 33 out of 50 per student.

The Instructions area of the evaluation scored 160 out of 240, which gave a mean value of 26.6 out of 40 per student.

The Screen Layout area of the evaluation scored 132 out of 180, which gave a mean value of 22 out of 30 per student.

4.4 Results of Child Evaluations

The evaluation at St Aloysius junior school had a total of 19 primary sevens assess the software using a Think Aloud testing technique. The complicated nature of a questionnaire like the one performed in the other evaluations meant that an interview technique involving general HCI questions which would give the child a chance to create there own opinion. The questions were based upon the sections of children’s design areas and Nielsen’s Heuristics. The question was also asked what they thought about the creation of an interactive character and what type of game they would like to see.

4.4.1 Letts Evaluation

- 2 children remarked that they preferred the simple problems. This was because they did not like mathematical software and shorter problems meant finishing problems quicker.
- 15 children did not like the problems because they were “boring”.

- 16 found the instructions hard to comprehend.
- 15 found the navigation to be poor.
- 5 were confused by errors messages.
- 11 ‘crashed’ the software.
- 7 found the buttons simple to understand and find.

4.4.2 Success Evaluation

- 1 child commented that they didn’t like the problems because they were “too childish”. The child believed that the software would have been better if a younger age group used the application. The child had no complaints about the educational level.
- 17 children liked the problem solving situations. This was because they had actual scenario problems to solve in comparison to paper problems. This has been found in previous research by the Microsoft Corporation.
- 2 interposed that they found certain parts hard to understand within the instructions in the fractions problems. This was due to the fact that the instructions were too long and they claimed to get bored by reading them.
- 4 found the buttons easy to find.
- 6 found the navigation poor. Some children were finding themselves in the wrong assessment area due to the fact they could not understand the navigation.
- 7 found the instructions simple. The children that made this comment were observed taking time to read the instructions and appeared to show a lot of patience.
- 6 didn’t like the instructions. A very mixed view was given about the instructions; there was no great difference in the amount of children who understood the instructions clearly and those who didn’t. The main difference appeared to be there patience.
- 1 believed the program should have given examples. Examples are a common practice in every type of education whether that is conversational or lectured. It would be obvious that the use of examples would be beneficial in this software.
- 1 commented that considering it was maths the problems were good. This was not specifically aimed at the HCI but did show that the problems were interesting. This is a guideline that should be followed in the children’s design areas.
- 2 found Success challenging. This was another guideline that should be followed in the children’s design areas, this should be educationally challenging and not usability challenging.

More children preferred the Success package due to its story scenarios that involved the use of mathematics to solve problems.

4.4.3 Children's Ideas for Edutainment Software

2 children openly commented without being provoked that they felt that the software needed characters. All of the children believed that more sound should have been in both pieces of software and that other games should have been between the mathematical problems so they give the child a break from concentrating on them. Another point that was recurrent was the use of short instructions in software.

The children were asked what type of educational software package they would like to see created so in the future that would help them learn. Some scenarios are listed below:

- 3 boys wished to see the crafting of a game which aloud the user to use mathematics to structure the wages and finances of a football club. This was due to their personal interests and they believed that it was a good way of learning and using knowledge.
- 3 children wanted to use a detective game in which mathematical problems were solved to find a murderer or stolen possession.
- 1 child wanted to use coordinates in a mathematical package to travel around space.
- 1 child wanted a mathematics package that could be used for clay pigeon shooting; this would incorporate trajectories and using a mixture of maths problems.

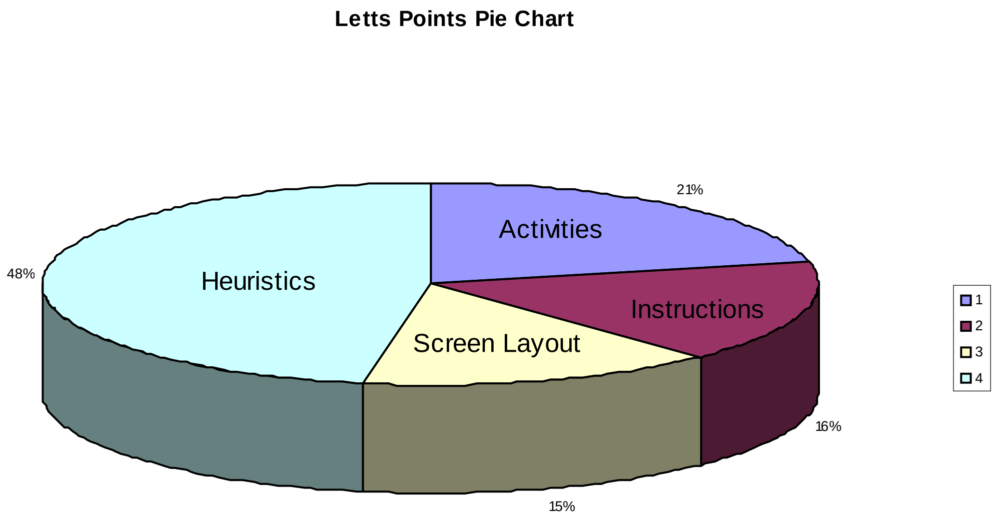
Other suggestions included golf, baseball and the fun fair. All of this information may be useful for the creation of a future edutainment package for children.

Due to the fact that the children had mentioned character intervention it was decided to ask them what character they would like to see. The results were very mixed from trees to aliens. The main point that was taken from the answers was that there were no characters suggested by the children based on real life people or television programs. This information may be useful to a future edutainment developer.

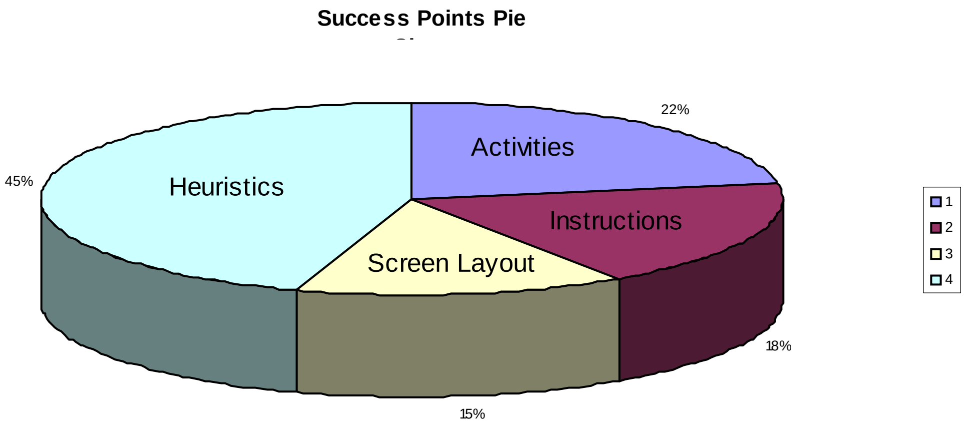
4.5 Summary of Results

It is hard to predict how much better an expert's results would be in comparison to the students' (if at all). Thus making extremely difficult to give an accurate points award as a total including both types of evaluation. To further complicate the points system there is different categories to be considered. Further more there is no evidence that has been presented in literature that states the level of importance of different design areas. This subsection will show the contribution of points each area designates to the final points tally. It will also show the percentage of scores each area achieved and give a conclusive verdict on the evaluations.

Letts Points Percentage



Success Points Percentage



The largest part of the points for both packages is the heuristic section. This is because there is more points available in the heuristic evaluation, with this being the case it is necessary to give each section a tally based on the points available i.e. a percentage. This would give the most accurate result for each area and make it simpler to draw conclusions.

Adult Evaluations Percentage Table

EVALUATOR	PACKAGE	AREA	%
Expert	Letts	Activities	24
		Instructions	7.5
		Screen Layout	20
		Heuristics	28
Expert	Success	Activities	48
		Instructions	47
		Screen Layout	40
		Heuristics	47
Students	Letts	Activities	58
		Instructions	58.35
		Screen Layout	73.34
		Heuristics	65
Students	Success	Activities	66
		Instructions	66.5
		Screen Layout	75
		Heuristics	66

Expert Variation

Activities	+ 24% Success
Instructions	+ 39.5% Success
Screen Layout	+ 20 % Success
Heuristics	+ 19 % Success

The expert evaluation found that the Success software was considerably better in all areas of evaluation.

Students Variation

Activities	+ 12% Success
Instructions	+ 8.15% Success
Screen Layout	+ 1.66 % Success
Heuristics	+ 66 % Success

The student’s evaluation found that the Success software was better in all areas of evaluation.

The final result from both evaluations would show that the Success package had the better Human Computer Interface. After the results had been assembled it was important to discuss the conclusions that they produced.

5. Discussion and Conclusions

5.1 Introduction

This section discusses the quality of the results that have been found and the conclusions that can be drawn from them. It will also show the reasons in which ‘Success’ was regarded as being a better package than ‘Letts’.

5.2 Discussion

Norma Said believed that the usability of children’s software was not at a high enough standard. Erlbaum believed that designers often neglected key areas of usability design. There have been no findings that suggest the usability of children’s software is of a high enough quality to believe otherwise. There has been no reason or research to prove that educational software aimed at children is any different. The aim of this project was to prove that the HCI for children’s commercial educational software aimed at “Key Stage 3” had many usability problem issues.

It would appear that the research studied during this project has been confirmed in the evaluations.

5.5.1 Expert Versus Students

This subsection will look at the differences in results between both adult evaluations. This will cover both the comments and the ratings.

<u>Letts</u>

With regards to the Letts application there is a very large difference in the percentage of points for each area. In all cases the student evaluations averaged over double the amount in points in every area in comparison to the Expert evaluation. A possible reason for this could very well be the lack of experience each student has with usability. Other explanations could also be that the students were unsure did not understand what was really required of educational software for children. Whilst it was not the first usability inspection for the expert it was very likely the first for every student. If all of these factors were to be considered it could explain the drastic difference in ratings.

The difference of the commentary for the Letts software was not great. The major difference was the quality and depth that the expert evaluation had in comparison to the student evaluations. However there were some good points that were made by some of the student evaluations. One student commented that the problem situations were “boring”. This turned out to be a comment that was not used by the expert but was used by children in the their assessments. However the expert evaluation was in depth with good comments such as explaining the ‘cluttered’ screen layout.

Success

With regards to the Success application there is not as great a difference as was found in the Letts package between the different forms of evaluation (Student Group and Expert). In all cases the student evaluations were once again averaging higher than the expert evaluations. The student assessment was averaged between 18/35% higher in evaluation areas. This is still a large difference but is still smaller than the Letts evaluation differences. A possible reason for this could very well be the lack of experience each student has with usability. Other explanations could also be that the students were unsure did not understand what was really required of educational software for children. Whilst it was not the first usability inspection for the expert it was very likely the first for every student. If all of these factors were to be considered it could explain the drastic difference in ratings. One possible reason for the decrease in the difference between both adult types of inspection is the quality of the software. This research may suggest that the better the usability the lesser the difference between an expert and a group-based student evaluation. This study is not conclusive however.

The difference of the commentary for the Success software was not great. The major difference again was the quality and depth that the expert evaluation had in comparison to the student evaluations. However there were some good points that were made by some of the student evaluations. One student commented that the icon dragging functions were poorly explained and it was difficult to implement. This turned out to be a comment that was used by the expert. This would show that there was areas in which the students were still producing expert quality analysis.

5.5.2 Adults Versus Children

This subsection will look at the differences in results between the expert and child evaluations. This will cover both the comments and the ratings.

Nearly every result found in this software by the children had already been discovered by the expert. This may be due to the fact that the questions that were created for the children were based on the expert evaluations. The only major factor that was brought up by the children that was not touched upon in any other evaluation was the lack of sound. This appeared to be a major factor that the children would have liked to see in both pieces of software. The integration of a character was communicated by the children and confirmed the findings by both adult evaluations. Success was again the preferred choice of software for the mass of the children due to the problem solving nature of the mathematical dilemmas. This was echoed in the adult assessments.

The mixed results over the quality of the instructions were very similar between the student and child evaluations. No conclusive result or opinion was obtained from these judgements.

No results or comments on a significant scale were discovered in any of the evaluations that would conclude that any of the screen layouts were of a high quality.

This may have been due to the fact that the evaluation technique was designed to find usability issues and not usability qualities.

5.3 Project Critique

As this is the first project I have performed on this scale that has involved research of this depth, I have learned a considerable amount about what is required both in work rate and educationally. I have learned that taking a piece of research at face value is not adequate enough and looking at other research to confirm the quality of results is also needed. I have learned a sizeable part of usability evaluating and children's usability. I have also learned structuring documentation and a methodical way of thinking.

During this study there were many areas in which I could have performed better. The structuring of my documentation and my writing skills hindered the reading and understanding of my documentation. My referencing was poor in the beginning and my understanding of research was substandard. However I am of the opinion that my effort and self-reliance was of a very high standard. I had thought of topics to research into before the Honours studies had begun. I had also met every target my supervisor thought I should be at, sometimes before that. Unlike other students I had to perform 3 evaluations, 2 of which involved using multiple people. In total I have spent time with over 27 people in evaluation sessions alone. This amount of people is more associated with questionnaires with regards to an Honours project. The added work rate required with preparing and researching different types of evaluation for different age groups caused concerns at times, though I am satisfied with the results.

The original aims of the project were to prove that there were many usability problems with commercial software aimed at the Key Stage 3 level in the Scottish Curriculum. The results of this project have certainly done that and also made suggested ways in which they can be improved upon. The results have also suggested other genres that could be used for the creation of an educational piece of mathematics software.

No issues at all arose which caused any concern in the project. The risk analysis was performed early on but no contingencies were used. There were 3 evaluations designed so that if one of the evaluations was unable to be performed then there would still be a strong project. All evaluations were completely successful and worked as they were designed. Since no problems arose, all deadlines were met and I am more than satisfied with the results and the organisation of the project I would not change how it has transpired.

5.4 Further Work

One of the problems faced by performing this study was the lack of tried and tested evaluation techniques for software at this educational level. No evaluation documentation was found that could have been used as an aid. This could be a final aim of an initial study like this. The conclusions and information from this study would also be very useful for the creation of a piece of 'edutainment' software. Not only does it give some do and don’ts of software creation in this area but it also gives suggested genres in which to base an application on.

5.5 Conclusions

The design areas aimed at children that have been produced by the Microsoft Corporation are effective. The description given for each area (Instructions, Screen Layout and Activities) appears to be integral in constructing a positive and efficient Human Computer Interface. Jakob Nielsen’s Heuristics also appear to aid in the evaluation of all types of HCI. They would not be a quality evaluation tool alone for children’s usability but do give a far better assessment of functional usability such as navigation. Without efficient functionality no software performs properly.

Character intervention was discussed with the children that would be useful research for a future edutainment developer. The expert evaluation appears to follow the Guideline Analysis well with no contradictive results between comments and ratings. This was not the case with the student evaluations. This information concludes that experience is key for a productive human computer interface evaluation of children’s software. This study has also confirmed previous research with comments made by the child evaluators. The children’s feedback presented the suggestion that the use of sound was found to be a stimulating media in software. The children’s feedback also made possible game suggestions that could contribute to the educational software industry.

The assessments have demonstrated that commercial educational software at the Key Stage 3 level of the Scottish Curriculum is flawed. The amount of usability issues that have been found in the software evaluated is quite disturbing. Based on the evaluation techniques performed by adults it was shown that there were usability issues in the educational packages. The evaluations done by the children of St Aloysius junior school have proven beyond doubt that there is poor usability at present in both the ‘Letts’ and the ‘Success’ applications.

References

[1] Druin, Alison (editor) (1999), A Design of Children's Technology, Morgan Kauffmann , California, USA

[2] Said, Norma S. (2004), Interaction Design And Children,
Proceeding of the 2004 conference on Interaction Design and children: building a community, ACM, Maryland, USA

[3] Bekker, M. (2003), Assessing Usability Evaluation Methods On their Effectiveness To Elicit Verbal Comments From Children Subject, Proceedings of the 2003 Conference on Interaction Design and Children, ACM, Preston, UK

[4] The Usability Company(2005) Usability Evaluation Methods [online] 2003
www.theusabilitycompany.com/methods
[Accessed 19 October 2005]

[5] Nielsen, Jakob (1990), How to perform a Heuristic evaluation [online] 2002

http://www.useit.com/papers/heuristic/heuristic_evaluation.html

[Accessed 19 October 2005]

[6] Nielsen, J. & Mack, R., Eds. (1994) Usability Inspection Methods. John Wiley and Sons, New York.Nielsen, J. & Mack, R., Eds. (1994) Usability Inspection Methods. John Wiley and Sons, New York.

[7] 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

[8] Nielsen, Jakob, Finding Usability Problems through Heuristic Evaluation, Proceedings of the SIGCHI conference on Human factors in computing systems, p.373-380, May 03-07, 1992, Monterey, California, United States

[9] Nielsen, Jakob, Usability Engineering, Morgan Kaufmann Publishers Inc., San Francisco, CA, 1993

[10] Ambron, Susanne, Kristina Hooper, Learning with interactive multimedia: developing and using multimedia tools in education, Microsoft Press, Redmond, WA, 1990

[11] Druin, Allison, ACM SIGCHI Bulletin - a supplement to *interactions*
Volume 2002 [online]

[Accessed 19 October 2005]

[12] Hourcade, Juan Pablo, Benjamin B. Bederson , Allison Druin , François Guimbretière, Differences in pointing task performance between preschool children and adults using mice, ACM Transactions on Computer-Human Interaction (TOCHI), v.11 n.4, p.357-386, December 2004

[13] B.Kafai, Communications of the ACM, Volume 39 , Issue 4 (April 1996) Pages: 38 – 39, ACM Press, New York, NY, USA

[14] Learning Teaching Scotland (2006) Software Education Levels [online] 2005

http://www.ltscotland.org.uk/5to14/resources/mathsgames/index.asp
[Accessed 9 February 2006]

[15] Lawrence Erlbaum (editor) The Human-Computer Interaction Handbook: Fundamentals, Evolving
Technologies, and Emerging Applications, 2002, Erlbaum and Associattes, 2005

[16] Miller, L. T., & Vernon, P. A. (1997). Developmental Changes in speed of information processing in young children. *Developmental Psychology*
www.uwo.ca/fhs/deansoffice/ FHS_annual_scholarship_report_2003-04.pdf
[Accessed 9 February 2006]

[17] Piaget, J. (1970). *Science of Education and the Psychology of the Child.* New York: Orion Press.

[18] Schneider, K. G. (1996). Children and Information Visualization Technologies. *Interactions, 3*(5), 68-73.

[19] Druin, A. (1999). *Cooperative Inquiry: Developing New Technologies for Children with Children.* Paper presented at the Proceedings of the ACM Conference on Human Factors in Computings Systems: the CHI is the limit, Pittsburgh, PA

[20] Goldman-Segall, R. (1996). Looking Through Layers: Reflecting upon Digital Video Ethnography. *JCT: An Interdisciplinary Journal For Curriculum Studies, 13*(1).

[21] Markopoulos, P., & Bekker, M. (2003). *Interacting with Computers*, Proceedings of the 2003 Conference on Interaction Design and Children, ACM, Preston, UK

[22] Jakob Nielsen, Usability and the Web (2005),

http://www.useit.com/

[Accessed 14 February 2006]

[23] Pre-school and Primary Software Yearbook 2006 (Catalogue)

Published by Curriculum Online

[24] Taylor, R. P. (Ed.). (1980). *The Computer In the School, Tutor, Tool, Tutee*: Teachers College Press.

[25] Skinner, B. F. (1968). *The Technology of Teaching*. New York: Appleton-Century-Crofts.

[26] Baekdal, (2005) http://www.baekdal.com/articles/Usability/usability-severity-rating/
[Accessed 20 February 2006]

[27] ERCIM(2006) Usability Evaluation Methods [online] 2004
www.ercim.org
[Accessed 30 April 2006]

[28] FrontEnd (2006) Usability Evaluations [online] 2004
www.frontend.com
[Accessed 30 April 2006]

[29] User Centric (2006) Usability Evaluations [online] 2004
www.UserCentric.com
[Accessed 29 April 2006]