



Citation for published version:

Shipp, VE 2006, Multimedia educational software to aid learning in pupils with reading difficulties. Computer Science Technical Reports, no. CSBU-2006-23, Department of Computer Science, University of Bath.

Publication date:
2006

[Link to publication](#)

©The Author December 2006

University of Bath

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

**Department of
Computer Science**



Technical Report

Undergraduate Dissertation: Multimedia Educational Software to Aid Learning in Pupils with Reading Difficulties

Victoria Elizabeth Shipp

Copyright ©December 2006 by the authors.

Contact Address:

Department of Computer Science
University of Bath
Bath, BA2 7AY
United Kingdom
URL: <http://www.cs.bath.ac.uk>

ISSN 1740-9497

**Multimedia Educational Software to Aid
Learning in Pupils with Reading Difficulties**

Victoria Elizabeth Shipp
BSc (Hons) Computer Information Systems
2006

Multimedia Educational Software to Aid Learning in Pupils with Reading Difficulties

Submitted by: Victoria Elizabeth Shipp

COPYRIGHT

Attention is drawn to the fact that copyright of this dissertation rests with its author. The Intellectual Property Rights of the products produced as part of the project belong to the University of Bath (see <http://www.bath.ac.uk/ordinances/#intelprop>).

This copy of the dissertation has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the dissertation and no information derived from it may be published without the prior written consent of the author.

DECLARATION

This dissertation is submitted to the University of Bath in accordance with the requirements of the degree of Bachelor of Science in the Department of Computer Science. No portion of the work in this dissertation has been submitted in support of an application for any other degree or qualification of this or any other university or institution of learning. Except where specifically acknowledged, it is the work of the author.

Signed:

This dissertation may be made available for consultation within the University Library and may be photocopied or lent to other libraries for the purposes of consultation.

Signed:

Abstract

A common trend in current teaching methods and educational software is to focus on providing reading intervention to children at a young age. However, reading difficulties still persist in older children reaching secondary school and thus need to be dealt with. A product is needed that can provide an appropriate level of learning content, within an environment that can motivate and stimulate older users. During this project the theoretical background of this domain is considered, followed by an in-depth stakeholder based empirical investigation to establish the requirements of the proposed system. Through a user-centred design process a high fidelity prototype is developed and evaluated. Preliminary indications from these evaluations suggest that the system provides high levels of motivation and enjoyment to the target users whilst also developing literacy skills. It is concluded that whilst these results show promise, further evaluations are needed to demonstrate the long term benefits of the system.

Acknowledgements

I would like to thank my supervisor Dr. Hilary Johnson who has provided me with invaluable support, direction and advice throughout the development of this project. I would also like to extend my thanks to all pupils and staff in the Special Needs Department at Oakmeeds Community College for providing such a fantastic amount of enthusiasm and support.

Contents

1	Introduction.....	1
1.1	<i>Aim.....</i>	<i>1</i>
1.2	<i>Objectives.....</i>	<i>1</i>
2	Literature Review	3
2.1	<i>Introduction.....</i>	<i>3</i>
2.2	<i>Cognitive Development.....</i>	<i>3</i>
2.3	<i>Reading</i>	<i>6</i>
2.4	<i>Reading Difficulties</i>	<i>8</i>
2.5	<i>The National Curriculum.....</i>	<i>12</i>
2.6	<i>ICT in Schools.....</i>	<i>15</i>
2.7	<i>Human-Computer Interaction.....</i>	<i>20</i>
2.8	<i>Conclusion</i>	<i>24</i>
3	Requirements.....	26
3.1	<i>Introduction.....</i>	<i>26</i>
3.2	<i>Stakeholder-Centred Requirements Analysis.....</i>	<i>27</i>
3.3	<i>Existing Software Evaluation.....</i>	<i>33</i>
3.4	<i>Requirements Specification.....</i>	<i>40</i>
3.5	<i>Conclusion</i>	<i>48</i>
4	Design.....	49
4.1	<i>Introduction.....</i>	<i>49</i>
4.2	<i>Early Design Decisions.....</i>	<i>49</i>
4.3	<i>System Architecture</i>	<i>50</i>
4.4	<i>Detailed Design</i>	<i>52</i>
4.5	<i>Low-Fidelity Prototype 1</i>	<i>54</i>
4.6	<i>Re-Design.....</i>	<i>56</i>
4.7	<i>Low-Fidelity Prototype 2</i>	<i>59</i>
4.8	<i>Further Design Decisions</i>	<i>61</i>
4.9	<i>Conclusion</i>	<i>63</i>
5	Implementation	64
5.1	<i>Implementation Language.....</i>	<i>64</i>

5.2	<i>System Architecture</i>	64
5.3	<i>Graphics</i>	66
5.4	<i>Database Design and Implementation</i>	67
5.5	<i>Integrating Flash, PHP and MySQL</i>	68
5.6	<i>Classes</i>	70
5.7	<i>Game Implementation</i>	71
5.8	<i>Reward Scheme Implementation</i>	76
5.9	<i>Learning Content Adaptability Implementation</i>	79
5.10	<i>Display Settings Implementation</i>	80
5.11	<i>Testing</i>	82
5.12	<i>Publication</i>	82
5.13	<i>Conclusion</i>	82
6	Evaluation	83
6.1	<i>Introduction</i>	83
6.2	<i>Planning the Evaluation</i>	83
6.3	<i>Usability Evaluation</i>	86
6.4	<i>Learning Content Evaluation</i>	90
6.5	<i>Conclusion</i>	95
7	Conclusion	96
7.1	<i>The Future</i>	98
8	Bibliography	101
8.1	<i>Books</i>	101
8.2	<i>Journals</i>	101
8.3	<i>Reports</i>	102
8.4	<i>Online Resources</i>	102
8.5	<i>Additional Reading</i>	103
	Appendices	A1

1 Introduction

A reading difficulty has many interpretations. One of the more familiar forms is Dyslexia, literally meaning a 'difficulty with words'. However, reading difficulties can be varied, and if left un-tackled can impede a pupil's academic achievements as well as opportunities outside of the school environment.

The teaching of literacy, due to its nature, can often benefit from the multi-sensory environment that a multi-media software tool can provide. For example a word on the screen (the visual representation) can be heard (an auditory representation) as well as manipulated. For this reason many tools have been created to assist in this area of learning.

Initial investigations into the software currently available in this domain has shown that the target age range for these tools can often be very broad, in some cases targeting users as wide apart as 5 to 15 years. This raises questions about the effectiveness of the tool for such a large audience. Can a tool that is used by a 5 year old also provide the same benefits to a 15 year old user, and will the environment and content level be appropriate for such a broad range of user?

The tools that target specific age groups tend to focus on the younger school pupils. The reason for this may be the fact that intervention techniques have been shown to be more effective in younger children (Sylva & Hurry, 1995). However up to 20% of 11 year olds entering secondary education are still failing to read at an age-appropriate level (House of Commons Educational & Skills Committee, 2005), and evidence found by Lynch, Fawcett and Nicholson (2000) shows that computer-assisted reading intervention can still be successful at this age.

1.1 Aim

Bringing these factors into consideration, the primary aim of this project is to create a tool specifically designed for the use of secondary school pupils (aged 11 – 16) with reading difficulties. The tool will need to include content and tasks at an appropriate level in an environment that captures and holds the attention of older users. This could be referred to as high interest/low level software.

1.2 Objectives

The key objectives of the project are as follows:

- To carry out reading into learning theory and teaching techniques.
- To look into contemporary issues in the area of Special Educational Needs (SEN) teaching. For example the area of Synthetic Phonics vs Analytic phonics.
- To carry out research in the area of HCI and how an optimum environment can be created for older users with reading disabilities.
- To observe lessons in a secondary school to analyse current teaching methods and areas that the tool could assist with.

- To carry out evaluations of current software that is available on the market including stakeholder evaluation and HCI analysis.
- To talk to relevant SEN staff and pupils to elicit further stakeholder requirements.
- To carry out a detailed iterative design process resulting in the creation of a high fidelity design prototype that can be evaluated by SEN staff and pupils.
- To carry out extensive and varied testing of the tool in a school environment with SEN pupils and staff.
- To consider the future use of the tool and areas which could be developed further.

The following chapter considers the extensive literature within this domain.

2 Literature Review

2.1 Introduction

Preece (2002) states that when creating an interactive system you need to firstly consider who is going to be using the system, where they are going to be using it, and what activities they will be doing when interacting with the system. For this reason, prior to designing the system, it is important to consider literature that may provide information in these areas.

The users of the proposed system will be secondary school children with reading difficulties as well as secondary school teachers. The environment will be schools and the activities involved will be both literacy and learning.

It is, therefore necessary to look into how a child develops and the differences between a child who reads normally and the special needs of a child with reading difficulties. Additionally, to understand the possible activities being carried out, a review of reading development and literacy will be undertaken. This will be followed by a consideration of teaching methods and the school curriculum as well as the use of ICT in schools.

2.2 Cognitive Development

A prominent and much studied theorist in the area of cognitive development is Jean Piaget (originally a zoologist) who was interested in human intellectual development.

Piaget's underlying belief was that humans are driven by a desire to make sense of the world and to achieve equilibrium. Equilibrium is a level of organisation and harmony with the current environment. If a state of disequilibrium is to occur a human will be driven to make changes in their behaviour and understanding to rectify the situation through a process of adaptation.

When a child is growing up they gradually develop a series of organised behaviour patterns for dealing with the world around them. These are known as schemas and are developed and expanded through assimilation and accommodation.

For example when a child is young their store of schemas will be relatively small. When encountering a new situation they will try and fit the situation to one of their existing schemas, such as when a child sucks a new toy, as they are used to sucking something to get food (this is the process of assimilation). However, if these schemas do not fit the situation (i.e. sucking the toy is not achieving anything) they will be in a state of disequilibrium and will be driven to adapt their schemas or even create new ones to accommodate this. In the example of a new toy they may discover that shaking it produces a stimulating noise.

This is a key idea in teaching as it indicates a child's willingness to learn and develop, especially when faced with a state of disequilibrium. The implications of this are that, if you can provide a child with a situation where the disequilibrium is just right, it can encourage growth. But as Hunt (1961, cited in Wolfolk, 2004) states, you need to match this to the child's current level of understanding. The child should not be bored by work that is too simple nor left behind by teaching that is too difficult for them.

Another theory of Piagets that can help in analysing the developmental stage of a child is his 4 Stages of Cognitive Development. These, he proposed, are the stages that every human will go through in the same order.

Sensorimotor Stage (approx. 0 – 2 years)

At the beginning of this stage a child is limited and can only make a small number of reflex responses to the environment (such as the sucking example stated above). As discussed the child can already use the processes of assimilation and accommodation to develop these responses into organised, more flexible patterns of behaviour.

Additionally at this stage a child will also develop the notion of object permanency - that an object exists even when it can no longer be seen. The ability to reverse simple actions is also gained.

Pre-operational Stage (approx. 2 – 7 years)

By this stage a child will be able to use schemas; however these do not help a child with concepts such as recalling the past and planning into the future. This requires the child to develop the ability to plan and reverse actions mentally, a skill which develops during this stage.

At this stage the semiotic function will develop. This is the ability of the child to use symbols such as words to represent things. For example, using a picture or the word 'bike' to represent a real bike. This development enables a child to develop a vocabulary that rises steeply during the ages of 2 – 4 from around 200 to 2,000 words.

An egocentric view of the world is taken and children will often perform a collective monologue by talking about what they are doing even if nobody is listening.

Concrete Operational (approx. 7 – 11 years)

This can also be described as the stage of 'hands on' thinking. The main step in this stage is the ability to recognise the logical stability of the world, such as the concept of conservation and reversible thinking that will have been difficult for a child prior to reaching this stage.

The ability to classify objects according to certain characteristics and recognising the concept of sub-classes is also a key development at this stage which leads to the development of seriation. This is the ability to order things from large to small and vice versa which in turn allows for more logical thinking such as if $a < b < c$, then b can be larger than c but smaller than a.

However thinking is still based on a physical reality and hypothetical and abstract reasoning is not yet accessible.

Formal Operations (approx. 11 + years)

In this stage thinking begins to shift from the 'what is' to the 'what might be'. This leads to the ability to form hypothetico-deductive reasoning – the ability to consider a situation and reason deductively about it. This leads to an individual being able to form hypotheses and test them as well as systematically considering all possible alternatives in a situation.

Adolescent egocentrism is also likely to occur, causing the teenager to become very focused on their own ideas, beliefs and attitudes. This can lead to the feeling that "everyone is watching them", described by Elkind (19781, cited in Woolfolk, 2004) as an imaginary audience. In this mindset any imperfections or social blunders can seem devastating to the teenager.

However, not everyone may reach this final stage. The first three stages according to Niemen (1985, cited in Woolfolk, 2004) will be forced onto children by physical realities and eventually even Piaget himself agreed that the Formal Operational stage may be achieved only in some areas. Another problem with this theory is that the timing of the onset of each stage can vary greatly between individuals.

The implications of these stages of development is that students in one year group may not all have reached the same level of thinking and it may even be difficult to tell which stage a child is at. This will make it difficult to set the difficulty of class material to match the levels of each student in the class. For example a student with reading difficulties may well be advanced in many respects even though their reading skills are behind that of their peers

An adaptable multimedia tutor may be able to assist in this area as the nature of computers enables the level of the work to be adapted to the needs of each student.

The tutor being developed is to be aimed at 11 – 14 year olds (Key Stage 3) and subsequently the children using it are likely to be at a transitional stage between concrete operational and formal operations. It may be especially important to consider the effects that adolescent egocentrism may have. For example, a child with reading difficulties may be becoming increasingly aware of being behind their peers in reading and this could in turn lead to much anguish at this stage of cognitive development.

2.2.1 Sociocultural Theory

A key problem with Piaget's theory is that it takes a very westernised view and represents the expectations of the western culture. Russian psychologist Lev Semenovitch Vygotsky supported an alternative view to Piaget's called Sociocultural theory. This is the view that culture and social interactions shape cognitive development.

He proposed that every function in a child's development will first appear in a social context before being internalised and used independently. These functions will develop through interactions with people who are more advanced in thinking such as a teacher or parent.

He also stated that tools such as computers can play a very important role in cognitive development. There are 3 ways that this type of interaction can pass on knowledge (Tomasello, Kruger & Ratner, 1993, cited in Woolfolk).

- Imitative learning, where the learner imitates actions.
- Instructed learning, where learners internalise instructions from a teacher (or tutor tool)
- Collaborative learning, where learning occurs through a group of peers trying to understand each other

When passing on knowledge Vygotsky believed that it was important to do this in a structured way through guidance and assistance.

Scaffolding

Vygotsky proposed that this structuring could be provided through scaffolding. Scaffolding is the process of giving information, prompts, reminders and encouragement to the learner at the right time and in the right amounts. This is then reduced, allowing the student to do more and more of the learning on their own. This can be done in the form of walking the student

through the steps of a problem, doing part of this problem for them and then giving detailed feedback (Rosenshine and Meister, 1992, cited in Woolfolk).

Zone of Proximal Development

One key theory of Vygotsky's was that at a point in time a child will be close to solving a problem. Yet, at this point in time they will not be able to solve the problem on their own. However, under guidance from a teacher or through collaboration with a more advanced class member they will be able to. This is called the zone of proximal development.

It is important in teaching as it emphasises the need to work with someone, or something else (such as a tutor) to work through problems that are nearly in the reach of the child.

The overall implications of these theories for a computer based tutor is that it needs to help a child develop by providing tasks of an appropriate level for the child - tasks that will stretch the child to the right extent in order to promote growth. It also needs to provide structured assistance to the learner through the use of scaffolding. Finally, the tool should be used to vary the task difficulty to match that of the user's current developmental level as this can be vastly different, especially in a special needs classroom.

2.3 Reading

Reading is a crucial skill for any child to develop as it can have a fundamental effect on their development in other areas, particularly at school where reading plays an important role in the classroom. It can effect imagination, emotion, morals and verbal intelligence and subsequently the type of person that a child may become (Harrison, 2004). It can also open the door to information and knowledge as well as to the simple enjoyment of reading a book.

For these reasons it is important that every child in school is given the chance to develop reading skills.

2.3.1 Reading Development

Reading is the ability to decode words in text into a meaningful mental representation. In order to do this the reader needs to know what sounds the letters represent and how to put these sounds together to make a word. For this reason reading involves both the processing and representation of visual information and the processing of auditory information.

Prior to reading development can begin, it is generally agreed that a child needs to develop phonological awareness. This is the ability to break a word up into its component sounds (but not yet linking them to written text). For example cat would become c-a-t. This skill can in fact be used as a measure of a child's future reading ability (Goodwin & Thomson).

There are three stages in the development of phonological awareness.

1. Awareness of syllables. This is the simplest way to break up a word. An example of this would be splitting the word footballer in foot-ball-er.
2. Awareness of onset and rime. This is the ability to distinguish between the beginning and ending sounds in a word, for example the ability to split the word bounce into b-ounce.

3. Phoneme awareness. This is awareness of the different sounds in a word such as sh-ee-p. As syllables this would be sheep and in onset and rime it would be sh-ee-p. (Goodwin & Thomson, 2004)

If this phonological awareness is not developed then a child will struggle to benefit from the teaching of phonics (Harrison, 2004). Phonics is the relationship between words and print. It is important for a child to learn this skill, as when they are presented with a new word they will not be able to recognise it as a whole. Instead they will need to split the word up into letters or blends of letters that represent sounds. They will then be able to put these sounds together to pronounce the word.

Once the key concepts of phonics have been grasped, a child will also become able to use analogies to decode unfamiliar words. This is the ability to use similarities between the new word and already familiar words to aid word recognition. For example if a child sees a word beginning with 'ca' and they are familiar with the word 'cat' they may be able to work out that the new word will begin with the same sound as 'cat'.

Through using analogies and phonological awareness a reader will continue to increase their vocabulary until they have a huge wealth of quickly recognisable words available to them.

The recognition of whole words is the key difference between good readers and poor readers. However, even fluent readers still process a large number of words through letter-by-letter analysis, although this process is achieved more quickly by this stage. (Harrison, 2004)

Stonovich (1980, cited in Harrison, 2004) has proposed an interactive compensatory model of reading. This is the view that good readers recognise whole words rapidly, as decoding is now an unconscious process, however poorer readers compensate for this by devoting extra resources to the process. Therefore, whilst word recognition is important to being a fluent reader a learner reader must be able to use as many tools as possible for word recognition (i.e. phonological awareness, analogies and context).

It is now, therefore, generally recognised that children need to develop both an automatic and phonological approach to reading words.

2.3.2 Teaching Reading

Over the last century there have been many approaches used in the teaching of reading and to this day much debate exists on the best techniques.

In the 1970s and 1980s a whole word approach to teaching was generally used. This involved using flash cards based on whole words to encourage children to develop a large vocabulary of instantly recognisable words. Children were also encouraged to use picture and context cues to decode words.

The problem with this is that rapidly recognisable word vocabularies is often the result of being a good reader rather than the cause, as the whole word approach does not allow for decoding new and unfamiliar words.

Research done by Bradley, Goswami and Bryant (Goswami and Bryant, 1990) argues the importance of phonemic awareness in learning to read. They claim that:

1. There is a strong correlational link between knowing nursery rhymes and acquiring phonemic awareness.
2. There is a strong correlation between acquiring phonemic awareness and learning to read.
3. The connections are causal.

Long term research undertaken in over 20 schools found a link between a child's knowledge of nursery rhymes at the age of 3 and their reading and spelling at age 6. Any differences in intelligence, social background and initial phonological knowledge were controlled so that they could not impact the results.

This research emphasises the importance to teachers in spending time looking at nursery rhymes, action rhymes and using word games to support the development of phonics and also the importance of a child acquiring phonological awareness before reading.

The implication of this is that the teaching of phonics is the key to reading development and that teaching of reading should focus on this area. However, such implications are the basis for an on-going debate. For example some teachers believe that reading is not just about decoding, it is about teaching children to become readers (Harrison, 2004) i.e. to read real books and enjoy the reading process rather than seeing it as a task. This is known as 'real reading'.

It is argued by Harrison that phonics cannot develop the interest that is required to get a child to read real books and without reading books a child will not develop the depth of vocabulary that is needed to become a fluent reader.

The implication of this is that in addition to teaching a child the basic skills in reading it is necessary to try and also encourage an enjoyment of reading. This is perhaps less easily achievable when carrying out repeated learning of sub-skills rather than reading real books.

2.4 Reading Difficulties

The tool that is being developed for this project is aimed at trying to teach reading to children with reading difficulties. Subsequently it is important to understand the nature of reading difficulties.

2.4.1 Dyslexia

With 3 million people in the UK suffering from dyslexia, it is probably the most well known form of reading difficulty and when translated directly literally means a "difficulty with words". However, there is currently much debate in the area of dyslexia as to the specific causes of the condition and even whether it exists at all.

The Dyslexia Myth

In September 2005 a Channel 4 Dispatches program titled 'The Myth of Dyslexia' was aired. In this program Julian Elliot put forward his theory that dyslexia is a myth and that poor readers are poor readers, regardless of whether they are dyslexic. The recommendation that Elliot made based on this was that it was no use classifying a child as dyslexic as the teaching methods needed would be the same as for any other child.

However, Rod Nicholson debates this in his article in the *Psychologist* (2005) titled 'Dyslexia Beyond the Myth'. In this article he states that the condition cannot be a myth due to the fact that 50% of dyslexia is genetic and therefore must have a 'clear and distinct basis'. He also points out further problems, such as Elliot's argument that 'children with low IQ can be helped just as much as children with reading problems with a high IQ'. Nicholson states that nobody has ever suggested that children with a low IQ can't be helped in learning to read and that the fact that children with a low IQ can learn to read just as well as a dyslexic child is one of the key enigmas of dyslexia.

In conclusion Nicholson states that a one-size-fits all approach to literacy is not realistic considering such strong psychological evidence to state otherwise. However he does state that perhaps the furor caused by the program could prompt researchers in the field to collaborate more in their work in this area.

The issues arising from this debate are somewhat out of the scope of this project, and the project will be based on the assumption that Dyslexia is a 'real' condition. However, any further issues arising as a result of this research will be taken into consideration if appropriate.

2.4.2 The Effects of Dyslexia

Before looking further into the causes of dyslexia it is important to look at the least disputed area – what the effects and symptoms are.

A definition of dyslexia produced by the government can be found on teachernet - an online resource for teachers. This definition is based on the symptoms rather than the cause of dyslexia and provides a simple overview of the condition:

“Pupils with dyslexia have a marked and persistent difficulty in learning to read, write and spell, despite progress in other areas. Pupils may have poor comprehension, handwriting and punctuation. They may also have difficulties in concentration and organisation and in remembering sequences of words. They may mispronounce common words or reverse letter sounds in words”

Dyslexia Definition (www.teachernet.gov.uk)

From this definition it can be seen that there are many difficulties that dyslexic children face. These can differ between individuals but can consist of many, or all, of the following:

- Problems with reading
- Fluctuating concentration
- Unorthodox spelling and grammar
- Problems with physical coordination and handwriting
- Difficulty remembering information
- Difficulties working to time limits
- Difficulties thinking and working in sentences
- Visual difficulties (e.g. blurring and distortion of print)

(Goodwin and Thomson, 2004)

A tutoring tool for dyslexic pupils will need to take these difficulties into account as they could all affect the way in which the child learns and also how they interact with the system. However, the primary focus of the tool is to develop and enhance reading ability.

The main types of difficulties that arise when a child is learning to read are as follows:

- Remembering the different sounds of vowels
- Telling the difference between certain letters e.g. p – q, and b – d.
- Pronouncing th and f
- Remembering the sound for ‘ough’ in words such as ‘through’, ‘though’, ‘cough’, ‘enough’ and ‘bough’.

(Goodwin and Thomson, 2004)

- Breaking down words into syllables
- Remembering rhythmic patterns

(Griffiths, 2002)

2.4.3 The Causes of Dyslexia

Above is a summary of the types of difficulties caused by dyslexia, but it is now necessary to look into the causes of dyslexia as this may impact on the ways in which dyslexic children can be taught effectively.

Extensive conflict exists in theories about the exact causes of dyslexia. Fawcett suggests that this may be due to the overlapping yet conflicting needs of researchers who, in order to secure funding, are emphasising the differences in their theories rather than the commonalities.

In the 1990s the dominant theory was that dyslexia is caused by a phonological deficit (Bradley and Bryant, 1983, Snowling, 1986, Stanovich, 1988, cited by Fawcett, 2003).

The deficits were said to be caused by abnormalities in the language processing areas of the brain, therefore causing dyslexic children to use the alternative and slower, less reliable, visual spatial centre of the brain when reading.

This is supported by Rack et al. (1992, cited by Fawcett, 2003) who state that the most clear and common sign of dyslexia in a child is difficulties with the reading of non-words. Reading of non-words can only be achieved through the use of phonics as the word cannot be recognised by sight.

The proposed solution for this deficit was to carry out intensive phonics training with children.

In more recent years new techniques in neuroscience, brain imaging and genetics have led to much progress in dyslexia research. Evidence from the US suggested that the cerebellum in human brains may be involved in language dexterity (in addition to speed, learning and motor skills as was already known).

This led Fawcett et al. (1996, cited by Fawcett, 2003) to conduct research into whether this area may be a possible cause of dyslexia, and in doing so, found strong evidence of abnormalities in cerebellum function in dyslexic individuals. This was supported by Nicholson et al. (1999, cited by Fawcett, 2003) who found reduced activation in the cerebellum in dyslexia.

These findings suggest that there are more widespread problems in dyslexia and may explain some of the dyslexic difficulties (such as poor handwriting and concentration) that cannot simply be linked to a phonological deficit. However, it is important to note that these brain abnormalities could be a result of dyslexia rather than the cause.

2.4.4 Teaching Pupils with Reading Difficulties

The methods traditionally recommended for dyslexic intervention usually involve overlearning. This is a process where newly acquired skills are practiced beyond the point of mastery to automaticity (wikipedia, 2005).

This concept is supported by Nicholson and Fawcett (2000, cited by Fawcett, 2003) who suggest that performance can become automatic, but that with dyslexic children this process can take much longer. Their “square root rule” suggests that this can in fact take longer by a factor of the square root of the time normally taken to acquire a skill. So, a skill that would take a normal learner 400 sessions to acquire could take a dyslexic child 8000 sessions.

As a result of this it is understandable that a dyslexic child may find it difficult to maintain motivation levels when repeatedly carrying out a task.

In addition to this, once a child reaches secondary school they may well have become frustrated and discouraged by slow progress to the point that they no longer bother trying. This is due to them holding the belief that they can no longer control or improve their learning – a mindset known as learned helplessness (Woolfolk, 2004).

The implications of this are that, whilst an interactive tutor may be able to provide a useful opportunity for over learning, it is important to ensure variety in these activities in order to avoid motivational issues. The tutor could also be used to provide additional motivational techniques and reward schemes to encourage an already de-motivated child.

If instruction is effective it can be highly successful and can even cause a dyslexic child’s brain to alter in appearance to resemble more closely that of a typical reader (Knight and Hynd, 2004).

However, the efficacy of intervention strategies can lower as a child’s age increases. Research evidence seems to indicate that early intervention in children with reading difficulties is more effective than remediation at a later stage (Sylva and Hurry, 1995).

This may be due to reasons such as the ‘Mathew Effect’ (Stanovich, 1986, cited by Lynch, Fawcett and Nicholson, 2000). This is where an initial failure to read can lead to a vicious circle of reduced exposure to print, reduced motivation and reduced opportunities to participate in literacy based activities. This can operate above the initial difficulties of the child and make remediation very difficult.

This poses the question of whether it is cost effective to carry out intervention schemes with secondary school children?

The fact that up to 20% of 11 year olds entering secondary education are still failing to read at an age appropriate level (House of Commons Education and Skills Committee, 2005) suggests that it is at least necessary in some from.

In addition to this a study by Lynch, Fawcett and Nicholson (2000) found that computer-assisted reading intervention in a secondary school can be both successful and cost effective.

This finding is positive, as, with the need for intervention demonstrated, it is encouraging to find that a tool can be effective at this late stage in reading development.

When developing a tutor it is important to consider these key areas such as the teaching of reading and issues faced when teaching children with reading difficulties. However, within schools, teaching is also governed by the National Curriculum and other guidance documents. Subsequently, it is necessary to look into these areas to understand how teaching in the classroom is currently carried out and the requirements that the curriculum places on this.

2.5 The National Curriculum

The national curriculum outlines the learning experiences that should be provided by schools to all their pupils. It secures an entitlement for each child to certain areas of learning as well as establishing national standards for pupils in the subjects that it covers.

The curriculum covers many subject areas, but highlights two areas that must be taught across the curriculum. These are:

- Use of language (reading, writing, speaking, listening)
- The use of information and communication technology (except non-core foundation subjects at Key Stage 1 and in PE).

This highlights the importance of a computer based literacy tutor, because, as well as developing the use of language for the students it also will provide additional exposure to ICT.

2.5.1 Special Educational Needs (SEN)

Within the national curriculum SEN is not explicitly covered as a subject. However, the curriculum is enclosed by an inclusion statement that outlines how teachers should modify programs of study as necessary to provide all pupils with an appropriate level of challenging work throughout each Key Stage. It sets out three principles to provide an inclusive environment:

- Setting suitable learning challenges
- Responding to pupils' diverse learning needs
- Overcoming potential barriers to learning and assessment for individuals and groups of pupils

This means that special needs teachers will need to alter the curriculum to meet the differing needs of the pupils in their classes, which includes students with reading difficulties.

In addition to the inclusion statement a SEN Code of Practised also exists. The purpose of this document is to provide practical guidance to LEAs, school governing bodies and providers of early education. The document sets out policies and procedures that should

enable pupils with SEN to reach their full potential and make a successful transition from childhood.

Within the document a child with SEN is described as follows:

“A child has special educational needs if they have a learning difficulty which calls for special educational provision to be made for them.

A child has a learning difficulty if they:

- (a) have a significantly greater difficulty in learning than the majority of children of the same age; or
- (b) have a disability which prevents or hinders the child from making use of educational facilities of a kind generally provided for children of the same age in schools within the area of the local education authority
- (c) is under five and falls within the definition at (a) or (b) above or would so do if special educational provision was not made for the child.”

The code of practice is an in depth document that sets out many key needs for teaching SEN children. This will, therefore, be consulted regularly when developing the literacy tool.

2.5.2 Literacy

The expected levels of a Key Stage 3 pupil for reading (the targeted age group in this project) at the end of year 9 is to be a shrewd and fluent independent reader:

- orchestrating a range of strategies to get at meaning in text, including inferential and evaluative skills
- sensitive to the way meanings are made
- reading in different ways for different purposes, including skimming to pick up quickly the gist of a text, scanning to locate specific information, close reading to follow complex passages and re-reading to uncover layers of meaning
- reflective, critical and discriminating in response to a wide range of printed and visual texts

However this is an unlikely level of attainment for a student with reading difficulties to reach. Teachers may therefore use the flexibility provided by the inclusion statement to draw on curriculum advice for younger pupils.

A useful source of guidance may be the National Literacy Strategy (NLS) which was introduced by the government in 1997 and is aligned with the national curriculum’s teaching of English. It is now also part of the Primary National Strategy.

The aim of this strategy is to provide a framework for teaching and to set out objectives for reception pupils up to year 6. The key objective is for these pupils to become fully literate.

The current NLS framework promotes reading in a broad sense, including decoding, comprehension and grammatical understanding. No individual aspect such as phonics (as discussed in section 2.3.2) is given a priority.

For example when a child sees a word that is unfamiliar to them they are encouraged to work it out by either:

- Inferring the meaning from the narrative context or syntax.
- Sounding out the word.
- Recognising the shape of the word from a previous encounter.

This model is known as the searchlights model (see figure 5.1).

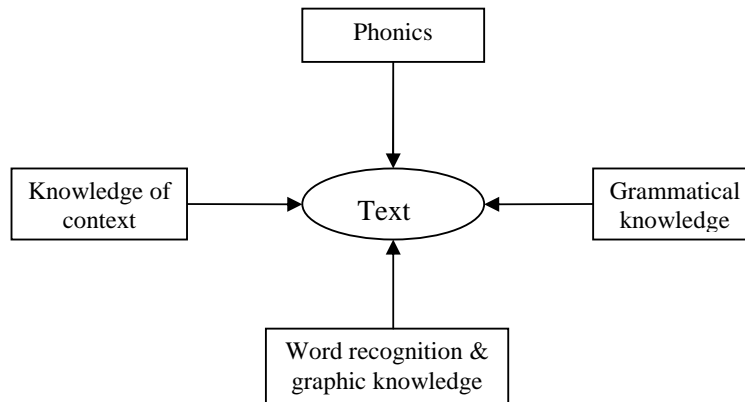


Figure 2.1 Searchlights Model (House of Commons Education and Skills Committee, 2005)

The belief is that the more searchlights that are switched on, the less critical it is if one of them fails. It is, therefore, optimising the range of cues for pupils.

In recent reports by the House of Commons Education and Skills Committee (2005) and former HMI Director of Inspection at Ofsted Jim Rose, this method has been criticised.

In Rose's report (2005) it has been pointed out that there seems to have been a tendency to look at what a fluent reader would do and assume that all strategies used here need to be covered when a child is learning to read. However, this can be a daunting prospect to a child and also reduces the priority on phonics.

It has therefore been recommended in these reports that the searchlight model be replaced. This has led to the proposal of a new phonics based model.

There are two approaches to the teaching of phonics:

- **Analytic phonics** – breaking down the words into onset and rime e.g. sh-eeep, rather than into it's smallest units (known as phonemes).
- **Synthetic phonics** – segmenting words into the smallest units of sound and then teaching children to blend these together to make a word.

Enquiries mentioned in the reports found that teaching in synthetic phonics produced the greatest improvements. A key study by Professor Rhona Johnston and Dr Joyce Watson carried out in a school in Clackmannanshire, Scotland compared synthetic phonics, analytic phonics and an alternative analytic phonics method.

The findings were that after a year those who had been taught using synthetic phonics were 7 months ahead of the other pupils in reading. All children were subsequently given synthetic phonics teaching over the next 7 years. At the end of this period all children were found to be achieving significantly higher levels in reading and spelling than would be expected for their chronological age.

The report therefore recommends that the NLS is changed to support a synthetic phonics approach to teaching literacy.

Key features of a synthetic approach are as follows:

- Letter/sound (grapheme/phoneme correspondences in a clearly defined, incremental sequence.
- To apply the highly important skill of blending phonemes in order to read a word.
- To apply the skills of segmenting words into their constituent phonemes in order to spell.
- That blending and segmenting are reversible processes.

This new strategy will have a large effect when determining the contents of the literacy tutor that is to be created during the course of this project. Whereas previous tutors may have used whole word or searchlight approaches to teaching, the new tutor will need to be able accommodate dividing words in the style of synthetic phonics. This will give the tool longevity, as to produce a tool that uses the old techniques may not fit in with the classroom curriculum.

The National Curriculum places an emphasis on using ICT across the curriculum. However, it is important to look into the specific benefits that ICT can provide to schools and more specifically to children with reading difficulties.

2.6 ICT in Schools

Today ICT is widely used in schools across the whole curriculum. On average secondary schools own 240.6 computers each which is 1 computer for every 4.1 pupils. Of these computers, 218.2 in 2004 were used for teaching and learning which is double that of the 1998 figure of 100.9 (ICT in Schools Survey, 2004). This shows the rapid development that is occurring in the area of ICT in schools.

In the BECTA review of ICT (2005) support for the use of computers in schools is noted by Passey et al (2003) who found that the impact on motivation provided by ICT was positive. Although this can only be the case if underpinned by a clear task orientation.

An additional finding in the report is that whilst ICT has a positive impact on pupil's motivation to learn, this is of a higher level with boys than girls.

It is reported by Cox et al (2003) that high quality interactive learning resources are more likely to result in higher gains in pupil learning than other resources.

However, the report goes on to state that there is no universally agreed framework for content quality. It suggests that the quality of educational resources can be affected by:

- Technical aspects such as interoperability and accessibility.
- System facilities such as tracking learner progress through a package

- Pedagogical design aspects such as whether it suits the needs and learning styles of the users and whether it is appropriate for the subject area it is trying to address.

These are all aspects that will need to be considered when creating a learning tool to ensure that it is of as high a quality as possible.

Cox et al (2003) also stated, however, that the positive effect on attainment is greatest when ICT resources have been integrated into teacher's practices. This means that ICT needs to be accepted and used effectively by teachers to support their work rather than to replace it.

2.6.1 ICT and SEN

In terms of SEN, it is perceived by schools that ICT has a positive impact. 61% of the teachers asked, agreed that it had some impact, whilst 30% said it has a substantial impact. This is a very positive outlook for the use of software aimed at SEN pupils. In fact 47% of secondary schools are equipped with specialist SEN software (ICT in Schools Survey, 2004).

Thomson (1984; cited by Lynch, Fawcett and Nicholson, 2000) noted that there are several potentially crucial advantages of computer based presentation for dyslexic children:

- It provides immediate feedback
- It can go at a child's own pace
- It is non-judgemental and predictable
- It can provide essential overlearning
- It can store and access information about the learner
- It can provide new motivation

A BECTA Overview of SEN and ICT provision (2002) highlights the following advantages that ICT can provide to children with special needs:

- ICT can be a motivating medium as many users are excited by computers and want to use them. Software that uses colours, pictures, animations, sounds and humour can build on this interest and engage the pupils in learning.
- ICT can enable the presentation of data in different ways. This is beneficial as students learn through different channels. Information presented in this way can provide more opportunities for the child to connect with the information being presented. For example information can be presented through text, graphics or sound to suit the individual's strengths and learning style providing them with alternative routes to understanding.
- ICT offers differentiation. Software packages can offer different levels of difficulty to suit the various needs of SEN pupils. Teachers can even create tasks themselves which can then be saved and adapted for the level of support required by each individual learner.
- ICT offers an efficient way of keeping records. Record keeping of pupil progress as well as contacts with outside agencies is an important part of the work of SEN teachers and coordinators. However this can be time consuming. Therefore using specially designed software can save valuable teacher time.

This supports the evidence that computer based learning approaches using ICT can have a very strong impact for students with learning disabilities.

2.6.2 Computer Assisted Learning

Taylor (1980) (cited by Merrill et al., 1996) suggested that all educational computer applications can be placed into one of three categorisations. These are tutor, tool and tutee applications. We will focus on tutor applications as this is the area that the tool being developed will most probably fit into.

Tutors

Here the computer performs a teaching role so that in effect the student is being tutored by the computer. This can be known as computer-based instruction (CBI), computer-assisted instruction (CAI) or computer-assisted learning (CAL).

The general format for these systems is that the computer will present some information, which the student is asked to respond to by answering a question or solving a problem related to it. The computer will then evaluate the student's response against specified criteria and based on this will determine what to do next.

Tutor applications can be split into further sub-types, three of which are as follows.

Drill-and-practice

The aim of a drill-and-practise tutor is to help a student memorise the appropriate response to a stimuli such as drilling a child in the spelling of words.

An example of this would be a student entering their spelling of a word which the computer would then analyse and give appropriate feedback to. For example if incorrect the computer would present the right answer and move on to the next problem. In a way this can be seen as a high-tech form of flashcards but the difference being that the questions can be tailored to the user's responses.

Tutorial

The primary purpose of these tutors is to teach new information. For example presenting the definition of a noun to a student, following this by showing examples of nouns and then asking the student to practise identifying them for themselves.

This method can once again be used to tailor materials to the individual needs of the student.

Game Applications

These application types are a way of bringing interest and motivation to learning. They generally involve competitive play between a student and one or more opponents. Aspects of game applications can also be used in other types of tutor.

The literacy tutor being developed for this project is most likely to be based around a drill-and-practice tutor. Anderson (1980) and Gagne (1982) (both cited by Merrill et al, 1996) point out that the more a skill is practiced the more automatic it becomes. This ties in with the recommended method of overlearning when teaching dyslexic children and can be easily implemented through a drill-and-practice tool.

However, a tutorial approach may be required to provide the initial scaffolding for the children when they first use the tool or if it has to introduce new concepts.

One problem is that drill-and-practise is unlikely on its own to be very interesting for the learner, but if gaming elements were to be incorporated into the system this could bring some interest and motivation to the work.

2.6.3 Studies of Computer-Assisted Learning and Reading Development

This section will look how ICT can be used specifically in the development of reading.

It has been argued that the use of multi-media presentations that utilize the motivational aspects and the ability of software to be used in a variety of ways can contribute to the success of computer assisted learning as a means of providing practice in reading instruction (Reinking and Bridewell-Bowles, 1991, cited in Underwood, 2000).

Computer interventions in the past have proven to be successful. For example Van Daal and Reitsma (1990, cited by Lynch, Fawcett and Nicholson, 2000) trained children with reading difficulties using words flashed rapidly onto the screen whilst the computer sounded either the whole word or the sub-lexical parts of the word.

Additionally Olsen and Wise (1992, cited by Lynch, Fawcett and Nicholson, 2000) found that students using ROSS (Reading Orthographics and Speech Support), a system allowing children to read stories with speech support, gained twice as much as control groups in untrained reading. However this approach can only be beneficial if an adult is present to provide further guidance.

Later studies (Wise et al. 1997, cited by Lynch, Fawcett and Nicholson, 2000) emphasised the importance of phonological decoding instructions due to the rise in research that suggested a causal role between phoneme awareness and reading disabilities. In fact many commercial intervention packages are now formed on this basis.

However, these current techniques offer only part of the solution to the whole problem of literacy as they do not offer much flexibility to the teachers and have built in assumptions about the development of literacy

A study conducted by Lynch, Fawcett and Nicholson (2000) which looked into the cost-effectiveness of a computer assisted reading intervention in secondary schools used a variation on these current tutors. The system used was called RITA.

RITA stands for Readers Interactive Teaching Assistant and was designed and programmed by Nicholson. The aim of the system was to combine the advantages of human and computer support by creating an intelligent teaching assistant (ITA). The role of an ITA is to act as an assistant to the teacher rather than replace the teacher. The idea is to free the teacher to carry out tasks where human instruction is vital (and cannot be replaced by computers) and allow the computer to carry out tasks which it is well suited to such as motivating, engaging, repeating, being non-judgemental, providing immediate feedback, record keeping and replays.

Within RITA teachers can specify activities for their students based on suggestions from the system (or they can choose an alternative). These suggestions are based on the child's current progress, guidelines and other available resources. Once the activities have been selected the sessions are pre-programmed and scheduled by RITA. Subsequently whilst a child carries out the scheduled activities the system will store and analyse the results.

Flexibility is introduced by allowing the computer to carry out the scheduling of the activities on its own and also allowing a child to take some control of their learning activities.

The key principle in RITA is that the teacher has the central role and the computer is an assistant. The teacher can adapt material to match current school activities and also on an individual basis.

The results of this study were that the intervention scheme showed excellent effectiveness and cost-effectiveness. In addition to this it was noted that the computer based presentation caused the children participating to show significantly higher levels of enthusiasm and commitment than with the traditional approach.

The researchers interpreted the findings as follows:

- It is possible to provide intervention in secondary schools where targets are set at an appropriate level for each individual child and that their Individual Education Plan (IEP) is followed.
- The RITA system proved effective in phonics, word attack, fluency and comprehension skills.
- The system was unsatisfactory for children with English as a second language – higher quality speech output is needed to distinguish English phonemes.
- If a child is in need of extensive support in secondary school, computer assisted learning may prove more effective than traditional methods. But continuing support will still need to be provided.
- Significant improvements in standardised scores were found, together with reading rate, accuracy and comprehension. Several children made progress of 12 months or more after 10 week intervention.

A similar study conducted by Underwood (2000) asked whether there is evidence that computer supported learning can be an effective strategy in the reading classroom?

The study compared two different forms of intervention, one based on a sub-skills training and the other on whole word books. The sub skills training used a system that is commonly known as an Integrated Learning System (ILS). An ILS is based on a behavioural model of learning using drill-and-practise to deliver the curriculum content and skills. The tutoring and practise is individualised for each user, introduced through a computer model of the child's current skills, which are then matched to the domain of knowledge that is to be acquired.

Brown (1997; cited by Underwood, 2000) says that an ILS uses three main components to facilitate the management of learning by teachers:

- Curriculum content – This consists of a range of tutorial, practise and assessment modules covering a large part of a pupils curriculum and across a range of subjects and abilities.
- A pupil record system – This stores and maintains information on every pupil as well as their current level of achievement.
- A management system – This links and controls the flow of data within the system and may perform functions such as:
 1. Interpreting the pupils responses in the tasks
 2. Updating students records
 3. Choosing a pathway through the curriculum content

4. Delivering the appropriate sequence of learning modules
5. Providing feedback to pupils and teachers.

An ILS tool can operate independently of the teacher or learner.

The results from this study were that whilst the tutor provided a positive motivational stimulus, the progress made by the ILS groups and control groups were much the same.

Where positive gains were made it was noted that:

- Children used the software in short sessions of 15-20 minutes three or more times a week.
- Children were focused on the task
- Teachers made significant interventions in the management of the learning environment, taking some control back from the computer.

However, in schools where a differentially negative outcome occurred it was observed that:

- The children used the software in 1 hour session once a week.
- Children were selected as needing ILS remediation rather than being involved with their peers.
- Children were distracted by the task as attention levels deteriorated during the long sessions. Additionally in some circumstances the children were un-supervised and attention to the task was limited.

The interpretation of these results is interesting as no benefit was found between the ILS and normal teaching except when a teacher took an active role in setting the agenda of the tool. This would suggest that using a highly automated tutor rather than allowing for teacher intervention can remove the advantages that computer assisted learning can provide.

It is, therefore, important to retain features that allow teachers to take control over the tool and adapt it to meet needs of the curriculum and other class activities.

It also highlights the need for a tool to be designed to be used 'little and often' rather than in longer sessions as it is not realistic to keep attention for such a long period of time.

Computer assisted learning is shown within this chapter to be useful in teaching literacy to students with reading difficulties as long as it is implemented in an appropriate way.

To enable the tool to be effective in motivating the children and maintaining their attention, the user interface will need to be designed carefully to allow cognitive effort to be expended on activities and tasks rather than on interactions with the computer. In order to ensure this, relevant HCI issues will now need to be considered.

2.7 Human-Computer Interaction

As Preece (2002) states, the key to a usable system is one that is easy to learn, effective to use and will provide an enjoyable experience for the user.

The first step in developing a usable system, however, is considering the users, domain and activities that will be involved. This has been undertaken during the literature review and these ideas will be drawn upon during this final section to deal with areas such as the

presentation of the system, how interaction will occur and any scope, flow and dialogue issues.

Some researchers into the area of HCI have developed sets of rules (also known as heuristics) that can be used almost as a checklist when designing an interface. Each rule that can be ticked should mean a more usable interface, however it is up to the designer to interpret and implement these rules effectively. Please see the Literature Review Appendix for a listing of both Schneiderman's and Nielsen's guidelines.

2.7.1 Interaction Design Basics

Dix et al (2004) sets out a number of interaction design basics that should be considered when developing an interactive system. Covered in the following section is an overview of some key areas that should be considered when designing an interface for dyslexic children using points from Dix et al (2004) and the findings of the literature review to date.

Memory

Dyslexic users may well have memory and concentration problems as has been previously stated in section 2.4.2. This means that it is important for the system not to overload their memory with information which is also reinforced by Schneiderman's 8th Golden Rule (see Literature Review Appendix) that emphasises the need for displays to be kept simple and time to be provided for learning sequences.

Sensory Memory

Sensory memory is used to filter stimuli to only those of interest but has a very limited capacity. Therefore, as the tool being created is to be highly sensory it is important that whilst it captures the attention of a child it does not overload them with information that may prove to be overwhelming.

To provide a system that does not contain too many simultaneous stimuli will help with this, allowing the user to take in the key information without distraction.

Short-term Memory

Short term memory is used to store information that is required very briefly. It also has a limited capacity. Miller (cited in Dix et al, 2004) states that the average person can remember 7 chunks of information + or - 2. Therefore any tutorials or instructions used within the tool should not contain more than around 5 key points.

However, providing good semantics are in place behind the information being presented, more information may be remembered as it can be 'chunked' into meaningful points. Subsequently it is important to consider the semantics behind the presentation of the information being displayed in the tutor as if this is done effectively more information is likely to be remembered by the users.

Long-term Memory

Whereas a human's short term memory provides a type of working memory, the long term memory is used for more permanent storage of information. The capacity is huge (if not unlimited) and forgetting takes place slowly over a long period of time. (Dix et al., 2004).

Long term memory can be divided into two categories, episodic memory and semantic memory. Episodic memory is an individual's memory of events in a serial form, whilst

semantic memory is a more structured categorised record of facts, concepts and skills (Dix et al., 2004).

To effectively present information in the tutor it is important to understand how best to get information into long-term memory. For example, Dix et al (2004) state that information from the short-term memory is embedded into long term memory through rehearsals such as repeated exposure to a stimulus. This is the approach taken by overlearning that had been discussed in section 2.4.4.

Through experiments into the recollection of nonsense words, Ebbinghaus (cited in Dix et al., 2004) discovered that the amount learned was directly proportional to the time spent doing this learning. Again, as discussed earlier in section 2.4.4, this will impact Dyslexic students who take a longer amount of time to learn a skill. However, Baddeley amongst others (Dix et al., 2004) have suggested that this learning time will be most effective if spread over a period of time. This is known as the distribution of practise effect.

Implications for this are that the tutor should distribute learning sessions over a prolonged period rather than providing it through more intensive short term learning as this should be more effective for the learners.

However, to make this learning more effective the information needs to be meaningful as this can be related more clearly to the semantic and categorised structure of long term memory (Dix et al., 2004). Therefore, to present information in a meaningful, structured and familiar way within the tutor will promote the transfer of knowledge to long term memory.

Information Retrieval

There are two types of information retrieval. These are recall and recognition. Recall involves producing information directly from memory whilst recognition involves realising that information being presented has been seen before (Dix et al., 2004).

Recall can be improved by providing cues to the individual. Vivid imagery is a common form of cue as people often visualise things when trying to remember them. Categorisation is also an effective way of aiding recall (Dix et al., 2004).

A tutor could aid recollection by providing both an effective categorisation of the information presented as well as visual imagery. Visual stimuli can be provided effectively in a rich multimedia environment and would aid visualisation during future recall.

Presentation & Ergonomics

The presentation of a tutor system will be highly important for any learners, however for dyslexic users this may be even more important.

Colour

Although it is tempting to do so, colour should be used sparingly. This is due to the fact that it can be highly distracting to users. In addition to this users should not be confused by the use of colours in a non-conventional way (such as using red text to display a confirmation as this is usually used to indicate a problem). This is summarised by Schniederman's 1st golden rule of interface design (see Literature Review Appendix).

Colour of text can be especially important to Dyslexic users who often like to choose the colour of the background that is used in a system.

In an interactive system it is possible for a user to be able to choose options for colours. However, a high figure/ground contrast should be used where the colours of text and the background are not too similar. Generally, dark colours on a light background are preferred.

Interaction Styles

As Dix et al (2004) states, interaction is a form of dialog between a user and the computer. Subsequently it is important to design an interface that meets the needs of the dialog that will be taking place as well as the users involved in this dialog.

There are many differing forms of interaction which should be considered in the development of the interface and this will be considered in more detail during the design of the system. However, with dyslexic users it may be important to consider the use of HCI redundancy. This is where interaction takes many channels (such as speech, text and graphics) to enable all routes to learning to be tackled. This will cater for the differing learning preferences of dyslexic users.

Navigation

When designing navigation it is important to make sure that at each point in time a user is able to see where they are and whether they are closer to reaching their goal destination. This is especially relevant with dyslexic users as short term memory problems can make navigation difficult.

A windows style system generally relies on a multilayered model to navigation which is navigated using drop down boxes. These can create barriers for users who may have memory problems, as all menu options cannot be seen directly and, therefore, require the user to remember where options are located (Dickinson, Eisma, Gregor, 2004). Subsequently, when designing a tool for dyslexic users it is important to make the navigation system as simple as possible so that complex menu navigation sequences do not need to be memorised and instead visual cues can be used.

Additional aspects to be considered when designing navigation is that using icons can cause confusion as some are not hugely self explanatory. As a result of this, icons, if used, should always be accompanied by an explanation or tool tip. With children with reading difficulties it may be appropriate to provide a verbal tool tip as a word may be as equally unfamiliar as a symbol.

2.7.2 Dyslexia Style Guide

Due to the nature of Dyslexia, text and the presentation of information is key to successful interaction. The following style guide combines ideas from an article on www.dyslexic.com as well as the British Dyslexia Association's (BDA) Style Guide. It pulls out the key ideas from both these documents in relation to text presentation and screen layout.

Font

- The fonts used, where possible, should reflect ordinary cursive writing. This is due to the fact that in general dyslexic people find natural hand writing easier to read.
- Serif fonts can obscure the shapes of letters, so san-serif fonts are preferred.
- The size of ascenders and descenders of letters (the stems on letters like p and b) is important as having them too short can distort the shape of the word.
- Space should be left between letters so confusion does not occur.
- The font size used on a screen should be a minimum of size 12 or 14.

- Lower case letters should be used where possible (i.e. no whole words in upper case) as this distorts the visual shape of the word.

Recommend fonts include Trebuchet MS and Arial. **Comic Sans** is often liked by dyslexic users but this can sometimes be seen as too childish and informal.

In fact, it is recommended that it should be possible for users to set their own choice of font style and size as well as background and text colours.

Text Layout

- Light text on a dark background should be avoided. The background colour white should also be avoided. Instead pale colours or off-white should be used.
- Using boxes to emphasise important text can be effective (especially as opposed to capitalisation).
- Lines should be left aligned (rather than justified) so that the right hand side has a jagged edge.
- Use bullet points in preference to continuous prose.
- Numbered lists are preferred to simple bullet points as this provides a frame for reference.
- Space between lines is important. It is recommended that this should be 1.5 to 2.

Writing Style

- Write in short simple sentences.
- Avoid starting new sentences at the end of a line as this makes it harder to read.
- Give instructions clearly.
- Keep sentences short at around of 15 – 20 words.
- Where possible refer to the readers as “you”.
- Use active verbs where possible i.e. “we’ll do it” rather than “it will be done by us”.

The following factors were also listed in the BDA Style Guide. However, these will also apply to designing an interactive tutor. Listed is a selection of the factors that have not already been covered:

- Navigation should be easy. A site map should be used.
- Graphics images and pictures are good ways to break up text.
- Large graphics may make pages harder to read.
- Moving text will be difficult to read for people with visual difficulties.
- It should be possible for users to see which pages have previously been accessed.

All these factors in the style guide will be taken into account during the requirements stage of the system development to ensure that the system is as usable as possible for dyslexic users.

2.8 Conclusion

High levels of debate currently exist in many of the areas covered in this literature review making it a difficult area to evaluate and summarise. Subsequently it will be important to remain aware of any future changes in these fields that may have implications for the literacy tutor.

From the findings of the literature so far it can be established that there is a need for tutors to help promote literacy skills in secondary school children with reading difficulties. However, it is important that this tutor fits into the learning requirements of the national curriculum and the recommended teaching methods detailed within this. In addition to this it should also incorporate relevant teaching theory from the areas of reading development and reading difficulties.

It has been shown that a computer aided tutor can provide many benefits to the teaching of children with reading difficulties such as providing a multi sensory learning environment and providing additional motivation where this may be low.

However when producing a tool for users with reading difficulties it is necessary to take into account the usability issues that may arise for these users such as the presentation of data and the need for the interface to be adaptable to each user's specific needs.

These aspects will all need to be taken into account when defining requirements for the tutor. This will help to ensure that the tool created makes full use of the advantages that a multi-media environment can provide whilst remaining both accessible to the users and compatible with the requirements of the National Curriculum and SEN guidelines.

3 Requirements

3.1 Introduction

Whilst the literature review provides a detailed look at the relevant theoretical background to the development of the system it is now necessary to gather empirical data. This, combined with the literature review will form the basis for the requirements of the system. These literature review requirements can be found in section 2.3.4 along with the full requirements specification.

The aim of this requirements analysis is to gain an insight into the potential users of the system, the tasks they will be carrying out with the system and the environment that this will occur in. The purpose of this is to help establish the type of support that the tool will need to provide, the type of activities it should be able to carry out as well as an understanding of any environmental constraints that may affect its use.

To ensure a usable product it is important to take a user-centred approach to this process and involve the users as much as possible during the requirements analysis and subsequent phases.

3.1.1 Stakeholders

In order to take a user centred approach to the systems development it is firstly necessary to identify the potential users of the tutor. These will primarily be the people who will be interacting directly with the tutor. However there are further users who will also have involvement with the system. Eason (1987, cited by Preece et al, 2002) has identified 3 different categories of user as follows:

- Primary users – those people who are likely to be frequent hands on users of the system.
- Secondary users – those people who will be using the system occasionally or through an intermediary.
- Tertiary – those people who will be affected by the introduction of the system or who may influence its purchase.

Therefore there are, in fact, a wide number of people who have a stake in the system's development. These stakeholders will subsequently need to be identified to enable them to participate in the development of the tool.

The stakeholders that will be involved in the development of the tool are as follows:

- Secondary school pupils with reading difficulties. – The primary users of the system. The pupils range from years 7 – 11 (ages 11-16) and all receive special needs education at school. The reading difficulties range from basic reading difficulties to severe dyslexia.
- Special Educational Needs (SEN) teachers – The secondary users of the system. These teachers teach the Special Needs lessons.
- Special Educational Needs Coordinators (SENCOs) – These are additional secondary users and have an influence on the purchase of systems. They are also both SEN teachers.

3.1.2 Requirements Gathering Process

During this requirements gathering process the potential system stakeholders as discussed above will be involved in interviews (sections 3.2.1 and 3.2.3), observations (section 3.2.2), and software evaluations (section 3.3) in order to gain an insight into their needs as well as the needs of their environment. In addition to this, a heuristic evaluation of existing software will be undertaken by an HCI specialist to recognise the usability problems of these packages.

3.2 Stakeholder-Centred Requirements Analysis

This section details the analysis that has been carried out with stakeholders of the system (as listed in section 3.1.1). The analysis has been divided into two sections to represent the differing stakeholders.

This stakeholder analysis involved the following sessions:

Secondary Stakeholders:

- Interviews with SEN teachers and a SENCO to gain their input into what functionality the system should provide for the pupils and also themselves.
- Observation of a current phonics based teaching method known as TOE-BY-TOE that does not use computers. This is in order to see how teaching occurs without the use of technology and also to observe a purely phonics based teaching method.
- Observations of a variety of SEN lessons to observe any environmental constraints and the general dynamics of the classroom. This also provided an opportunity to observe how software fits into an average SEN lesson.

Primary Stakeholders:

- Observation of pupils using the computers within the classroom to gauge their current computer competency.
- A series of semi structured interviews with pupils to gain their opinions of current software being used, their computer competency and usage and to elicit their ideas for the new system.

3.2.1 Teacher Interviews

SEN teachers, as discussed previously, are key stakeholders in the system and subsequently it was necessary to interview them to understand their system needs and also to gain important information on the SEN domain.

These interviews took place over a series of school visits and were semi-structured interviews. Semi-structured interviews use both closed and open questions and are based on a simple script which guides the direction of the interview (Preece et al., 2002). During these sessions a number of specific questions were answered whilst other discussions took a more general approach.

Potential Role of the System

When asked about the potential role of the system, teachers highlighted Ofsted's recommended lesson structure. This recommends that all lessons should be divided into three parts. These are as follows:

- Starter Activity – to introduce the topic
- Main Activity – the main body of the teaching session
- Plenary Activity – where the class go back over what has been covered in the lesson. For example a verbal recap or test. The aim is to focus on the learning and to reinforce and consolidate it as well as provide overlearning.

During the interview the SENCO stated that the systems preferred use within the lesson would be as a plenary activity to go over what has been covered in the lesson with the aim of encouraging the pupils to build on their success and improve their knowledge. It was suggested that it would be useful if the system could cover words that have been used during the main activity stage even if this is not directly related to the skills the game is developing. For example the words from a story that the group has been reading together during the lesson could be used within the system.

This suggested usage supports findings from the literature review that suggest that a computer based system would provide a useful tool for overlearning. It also supports the idea of the system as a support tool for teachers rather than as a replacement for the teachers.

Current Computer Usage

When asked about the current use of systems in the lessons it was stated that the current usage of computer based games in the lessons is not regular. The teachers said that the usage may occasionally be planned as a plenary or starter activity as discussed above. However they may also be used unexpectedly in lessons, such as when a student finishes some work and needs an extra activity (as was observed in year 10 and 11 coursework sessions). In addition to this if a pupil is being disruptive (a lot of the pupils in SEN lessons also have behavioural difficulties) the games may be used as a class activity whilst the pupil is dealt with or alternatively as an activity to calm the disruptive pupil down.

It was noted during discussions that systems such as Wordshark (see Requirements Appendix section 2.1 for a more detailed discussion of this system) take a lot of time to set up. With SEN pupils it is not always possible to give instructions to the whole class (such as what word list to use or what game to play) so the teachers will often need to do this themselves for each pupil. This can cause problems in a larger class as pupils may be left waiting while other games are set up. It was subsequently suggested that it would be useful to be able to set the game activities up beforehand. However there must also be ways of dealing with unplanned use.

In addition to this the teachers highlighted the fact that it should be possible to lock the system in some way to make sure that the pupils aren't changing the levels themselves (i.e. making it easier). This often occurs with systems where students have control over the levels that they are playing.

Possible Learning Content

The area of content for the tutor has already been partially covered in the literature where the literacy strategy was covered as well as possible approaches to teaching literacy. However it is also necessary to discuss this with SEN teachers to understand what they feel the system should cover.

As highlighted in the literature review, the nature of SEN teaching is that content should be adapted to the different needs of the children and their various stages of development.

Areas highlighted in the discussion as possible areas for the tutor to cover are as follows:

- Syllables
- Prefixes/suffixes
- Spelling patterns such as vowel diagraphs and consonant blends (e.g. th, bl).
- High frequency irregular words such as said, pretty, goes, does etc.
- Homophones – words that sound the same but are spelt different e.g. two, to and too.
- Silent letters.

It was suggested that these areas could be covered in the games within the tutor and possibly that each game could cover a different skill.

An important discussion in this area was on the subject of synthetic phonics. Whilst synthetic phonics is a key area of learning highlighted in the literature reviews and subsequently an early requirement of the system, this area was raised as a possible problem.

Although the school is hoping to implement a synthetic phonics approach to learning in the near future this has not yet been possible due to resource problems. It was felt that it would be inappropriate to introduce a game using this technique to pupils who have not encountered it yet. Subsequently it was decided that for the purposes of the prototype, no synthetic phonics learning content should be implemented.

3.2.2 Teaching Observations

In order to fully understand the nature of SEN classroom dynamics and teaching practice a series of these lessons were observed. The purpose of these sessions was to view current teaching techniques and gain an understanding of where a multimedia system could fit into and complement these processes.

The process used for these observations was “Quick and Dirty” observations with the aim of finding out the general classroom dynamics. However, when gaining more specific data about teaching practises and environmental constraints a holistic form of field study observation was used.

TOE BY TOE[®] Observation

TOE BY TOE[®] is a structured reading scheme that has been devised by Keda and Harry Cowling (Cowling, K. & Cowling, H., 1993). It is aimed at providing step by step reading activities that promote structured reading development in children (including those that are non-dyslexic).

The sessions require each child to possess their own *TOE BY TOE[®]* book for which they are responsible for bringing to each session and taking home with them in order to do work with their parents. The books contain a series of exercises that begin at a very simple single letter level and work up to longer more complicated words and sounds. The students are required to work through these exercises in order.

Exercises involve the child reading a letter, letter blend, real word or nonsense word aloud. If this is read correctly then the child receives a line by this to indicate that it was completed successfully, but if this is not read correctly a dot is placed there. The child must read each word in an exercise correctly 3 times in order to progress to the next stage. This technique is used as it is believed that although reading it once may place the sounds in the child’s short term memory; repeated exposure will allow it to be transferred into long term memory. At the end of each section sentences are presented that use the words and sounds recently covered. The child needs to read these correctly to progress to the next stage.

Session Observation

In order to see how these sessions take place in a classroom environment it was necessary to attend and observe a session that was happening. A description of this session is as follows:

The sessions are held twice weekly with identified dyslexic students. Each session lasts 15 minutes and should be accompanied by work at home with the pupil's parents.

Each student sits with an allocated Learning Support Assistant (LSA) who works through from the point that the student last reached. The session is carried out in one classroom around a table or on comfortable chairs placed around the classroom.

During the observed session LSAs would give feedback as the child reads. If necessary the meaning of unknown or unfamiliar words could be discussed or rules that the child uses to help read words (for example how to tell the difference between letters b and d). The LSAs also marked the book with the applicable mark if the student successfully read the word or did not.

If a child was having attention difficulties or there were external disturbances (such as other pupils walking in to the classroom which occurred during this observation session) then an LSA may need to deal with this, leaving their allocated pupil unable to continue with the reading.

This highlights the problems that can occur when a technique relies on one-to-one work as a teacher has other responsibilities beyond the task at hand and may be required to fulfil these higher priority issues at any time. A computer system would not have such issues as teachers would be free to control the class whilst the children continued to use the system.

In addition to this, problems can be encountered if a pupil forgets their book as they cannot fully participate. In this session one pupil forgot their book and instead read a story book aloud. A computer program would provide an advantage in this area as it would always be available to a pupil as long as they had access to a computer.

During these sessions it was clear to see the importance of one-to-one help and how valuable human feedback and interaction can be. For example if a child was not familiar with a word, the LSA could explain the meaning to them. A computer system would not be so capable of this highly tailored feedback. However, as discussed it would overcome the problem of a teachers divided responsibilities when distractions occur.

Observation Summary

It was important to observe this session as it highlighted the areas that a computer system cannot replace such as immediate personalised feedback. However it also demonstrated areas that a computer system could help with such as being much less resource intensive.

Spelling Test Observation

During the observed lessons a spelling test was carried out on two occasions. These tests are given to individuals depending on their needs, however in the lessons observed all pupils in the class were tested. The frequency of these tests also depends on the needs of the pupils but usually ranges from every SEN lesson to once a fortnight.

What follows is a description of the sessions which has also been expanded on as a result of discussions with SEN teachers.

The tests generally occur at the beginning of the session or after a silent reading session. At the start of the test each pupil is given their own personal word sheet. This lists all the words that they have covered in the test to date and also includes one new word that has been added by the teacher. The new word may have difficult sections highlighted and previous words that have caused difficulty may also be highlighted.

The words on these lists are different for each pupil and tend to be high frequency irregular words. They are taken from a master list of words that are ordered by increasing difficulty level. The pupils will have initially been tested to determine the level that they should start at and they then work forwards through the list.

The aim of this stage of the test is for the pupil to familiarise themselves with the new word. They will do this by looking at the word, covering it, saying it and then writing it down in the first column of the sheet. They will then repeat this process in two further columns. They will then also look at the last 9 words on the list to recap on them. These 10 most recent words (including the new word) will be the words tested at the end of the session. This means that each word will be covered 10 times.

After this initial session the word sheets are taken away. The teacher then uses a neuro-linguistic multi sensory approach to further familiarise the pupils with their words. This session involves the teacher asking questions about the words, such as the number of times a letter appears in the word or where it appears, how many syllables the word has or the beginning and end letters. The aim of this session is to get the pupils visualising the words.

During the observed sessions these neuro-linguistic sessions were seen to be highly interactive with the pupils often helping each other with their questions in a friendly non-patronising way. A feeling of teamwork could clearly be perceived.

After the neuro-linguistic session the pupils are tested on their 10 words. As the words are different for each pupil the teacher will go round the group giving them one word at a time. When this is completed the lesson continues and the tests are marked by the teacher before the end of the lesson. At the very end of the session the pupils are given their tests back and before they leave are asked a couple of questions about the words they got wrong.

Observation Summary

These spelling sessions complement many of the areas recommended in the literature review such as the need for overlearning as well as a multi sensory approach to learning. They also provide an idea as to how spelling tests are carried out in a special needs classroom. Techniques from these sessions could be incorporated into the final game such as the process of repeated exposure to words.

General Classroom Observations

During the time that this work was being carried out many SEN lessons were observed, whether fully or partially, during the course of other requirements analysis. This enabled a range of different group sizes, year groups and general dynamics to be observed. This has allowed a variety of environmental constraints to be recognised as well as more general requirements.

The general findings from the sessions are as follows:

- Each session involved very different pupil dynamics. Due to the diverse nature of the pupils' personalities and the difficulties they have, each SEN group can be widely different. Some groups have high cohesion levels where the pupils work well as a unit, however in other groups the personalities are very different and the pupils may not get on with each other.
- The nature of the classroom activities varies amongst the groups. Some groups have very structured lessons with set activities; however year 10 and 11 groups tend to use some of the lessons to obtain support with their coursework in other subjects. This means that each pupil may be getting on with very different work.
- For good work the pupils are given stickers that they are able to decorate their folders with. They are able to choose stickers that are of interest to them such as animals and football. They also collect stickers that they can make up their names with on their folders. Although this is a technique more common with younger schools the pupils seem to be enthusiastic about this as they are able to be creative and decorate their folders.

Observations of computer usage in the classroom:

- Pupils often asked to use the computers at the end of a session. This was more common in the year 10 and 11 sessions as when a piece of coursework is completed they may have 5 – 10 minutes free at the end of the lesson when it is too late to begin another piece of work.
- Pupils often used computers to carry out other pieces of work such as coursework or when given the task of writing something up for the lessons. In general were very competent on the computers although it must be recognised that these were the pupils who had chosen to use the computers.

3.2.3 Pupil Interviews

These sessions again used a semi-structured interviewing technique. The aim of the session was to establish what pupils enjoy using computers for (to understand what may motivate them), and also to gather ideas that they have for a new system.

Open questions were asked to allow freedom. This technique allowed the interviews to be steered towards problems and ideas that the pupil appeared to have and subsequently to gather information that may not have been anticipated prior to the interview.

However, before the interviews took place a question plan was drawn up listing the types of information that needed to be gathered from the sessions to guide the interviews.

During the sessions themselves, high-level notes were taken that were then revisited immediately after the session to highlight the main points that came out of the sessions.

General Computer Usage

When asked about computer usage in their spare time, all pupils said that they had access to a computer at home and appeared to be competent and familiar with computers.

The tasks that they used the computers for varied quite significantly, especially across sexes. In general the boys said that they use computers and the internet for playing computer games such as war games, racing games and fighting games. However, they also used the internet for downloading music.

This was the one main overlap with the computer usage of the girls who liked to use the computer for downloading music, as well as sending emails and talking to their friends on MSN.

Summary

From these responses it was possible to establish that nearly all potential users of the system will be familiar and comfortable with computers and will have access to these at home.

Importantly it has been possible to see what motivates the pupils to use computers. The difference in sexes may be problematic, as to create a game theme that will appeal to all users may be difficult.

Ideas for a New System

During these interviews, the pupils were keen to suggest ideas for a new game, often prompted by discussing the problems of the previous two games (See section 3.3). These suggestions are highly important in this process as they represent what the users would like when playing games at school.

One common suggestion by both sexes was focused around the scoring within the games. A number of users wished to be able to keep their score and save their progress between sessions on the games. A number of pupils also suggested that they would like to be able to enter their score onto a central scoreboard. The motivation for this seemed to be both the ability to see their improvements as well as the ability to compete with fellow class members.

One interesting suggestion by a pupil was that games like Pac Man and Ping Pong could be used as reward games. The reasoning behind this was that games of this format “never get boring”. This idea is interesting as it raises the idea of traditional arcade games. Games of this type have a seemingly universal appeal and have survived the test of time as they still remain popular today in a world where highly complex games have become the ‘norm’.

A desire was also expressed by pupils to be able to play the games at home. This backs up the comments made about the currently (see section 3.3.1).

3.3 Existing Software Evaluation

In order to further understand the requirements of a new system it was decided that evaluations of two systems that are currently used within the school would be beneficial. These systems are Wordshark and Education city. For a more detailed overview of these games as well as related screen shots see the Requirements Appendix.

In order to gain a varied perspective on the systems and to uncover a variety of usability problems with these systems, the evaluations were carried out in two formats.

The first of these sessions was evaluations with the pupils. These were undertaken by observing the pupils using the systems and questioning them as they did this. This allowed

clear problems to be observed but also allowed for discussions with the users on their opinions of the systems.

A heuristic evaluation of two commonly used systems was also undertaken with an HCI specialist (an HCC MSc student). This was done in order to uncover more specialised usability problems within the systems.

3.3.1 Primary User Evaluations

These sessions were held with a number of pupils from a number of year groups in their Special Needs lessons.

Prior to the sessions an agenda was developed so that each session followed the same pattern.

Wordshark

Agenda

For the purposes of the evaluation the users were first asked to play a game that they were familiar with and had used before. The reasoning for this is that it would be possible to observe how the users interact with a game when they have used it before. Do they take shortcuts in the game? Are there problems with the game that they have had to learn to overcome? Do they appear motivated by the game after repeated use?

In addition to this the users were then asked to play a game that they were unfamiliar with. This was in order to observe how users approach a new game, the ease with which they pick up the game controls and any issues that arise due to this unfamiliarity.

Findings

The findings from these sessions were as follows:

General Opinions

In general the consensus on the system was that it is satisfactory (or “ok” as many pupils stated). A number of pupils commented on the fact that it helps them with their spellings and one pupil specifically commented that it helps them to understand more about the sounds of words.

A large number of pupils commented on the patronising and childish nature of the games. One pupil stated that “the game has useful objectives, but the way it is put across can be portrayed as patronising and may put off the pupils”. These comments support the need for a game that targets older users, as efforts can be made to avoid patronising the users.

Other comments on the system included the fact that the game helps by telling you where you have gone wrong in the game when you improve.

Help and Documentation

During observations, one pupil choose to first play a game called Snakes and Ladders as this was a game they were very familiar with. From an observers perspective it was difficult to establish the exact rules of the game but the pupil got on with it straight away and seemed highly familiar with the game controls although it seemed quite complicated. This was also the case in a number of observations as when familiar with a game the users were very proficient at it and worked very quickly through them.

When one pupil was asked how they knew how to use the game they stated that their classmates had explained it to them. When asked if there was a help function or user guide, they were not aware of one.

When using a new game within Wordshark a few users became confused by the omission of any instructions before or during the game. The reaction to this problem was mixed. Some users quickly picked up the game through trial and error whilst others took much longer and got frustrated by this.

It was also noted that, although one particular user was clearly competent with computers (as they were studying IT for their GCSEs) they could not work out how to exit the game as it was not explicitly obvious.

Learning Content

One problem noted from a series of observations was that the pupils did not make use of the extensive word lists and instead used the default words. Some pupils highlighted the fact that the words in the games were too easy which may be caused by this fact. In fact, only one pupil set up more difficult words without being prompted to do so. When questioned about their choice of words a pupil stated that normally the teacher would set their words up for them.

Subsequently in one situation it was possible to carry out an evaluation when the teacher had set the system up for that user. In this situation the game seemed much more challenging and required greater concentration. However, it takes time for a teacher to set up games for each pupil in the class.

Input/Output

Within Wordshark speech output is used within the games. In one session the pupil misheard a word and subsequently got frustrated when they could not spell the word correctly. The clarity of the speech output was quite low on the computer speakers.

Controls

During another session the game chosen by the pupil relied purely on mouse controls. As mice in schools tend to get broken through large amounts of use the mouse on this computer was not working correctly. This made game control difficult and subsequently the user became frustrated by this and instead chose to play another game that did not rely on mouse control to such a large extent.

Reward Scheme

When observing users during the reward section of the game it was noted that they seemed generally enthusiastic about it and got involved in the results of the game. Many pupils commented on the fact that the games were good “as they make you feel like you have improved with your learning”, however others stated that the reward games can get boring. This indicates once more that designing a reward scheme that appeals to everyone may be complicated.

One pupil, when talking about the reward schemes, stated that they enjoyed the fact that at the end of the game they won rewards but didn't like the fact that you only get points if you do well on the reward game (which is seemingly random).

Education City

Agenda

The agenda for this session differed to the Word Shark testing. The reasoning behind this is that the users have not been using Education City for a long period of time and are not as familiar with any of the games. In addition to this the games in Education City take a much longer period of time to complete. For this reason it was decided that the users should all play a game that is unfamiliar to them as a higher number of usability problems are likely to arise with a new game than with a game where the users have learned to overcome these issues.

For the purposes of the evaluation all users were asked to open Education City and select an activity at year 3 or 4 level (as recommended by their teacher prior to the sessions). They were then asked to play 'Sten the Wizard' (again suggested by the teacher).

Findings

General Opinions

In general the pupils tended to agree that Education City was better than Wordshark. The reasons for this included the graphics and better layout of the game.

However, there were a large number of comments on the speed of the game with many stating that it was too slow and too time consuming. The probable reasoning behind this is the fact that EducationCity has a large amount of speech output and all text in the games is read out slowly. There is no way to progress through the game until this has been completed. Despite these problems the pupils appreciated the speech output in general. One pupil did, however, comment on the fact that the voice sounded "too much like a nursery rhyme teller".

Once again the pupils commented on the childish nature of the games with one pupil stating that it could "make you feel like you have the IQ of a child". This provides yet more reinforcement for the need to develop a game specifically aimed at a more mature age group.

One area of confusion with this game was the availability of the game at home (as it is a website). One pupil mentioned that it is only possible to play the game at school as you need the school password to access the system; however another pupil stated that they had played the game at home using this password. Regardless of the confusion in this area, these comments indicate that pupils have an interest in playing games outside of school and subsequently it may be desirable to produce a game that allows this type of access.

Input/Output

When using EducationCity no users appeared to have a problem logging into the system (as they had all used it before). However one user, due to their reading difficulties, could not spell the address for the website and had to have this spelt out to them.

During 'Sten the Wizard' the users are required to use an on screen keyboard to spell words. A problem that one user voiced (but was also observed in other sessions) was that they were having trouble with the keyboard layout as it was organised alphabetically rather than in the typical 'qwerty' layout. This confused the user who automatically looked for letters where they are usually located on the keyboard.

Help and Documentation

When playing 'Sten the Wizard' one pupil became confused as to what the instructions were. When they were instructed that they could re-hear the word (from an external source) they

used this facility but had not been aware that it was available as it is not clearly stated anywhere in the game.

User Control

When working with a pupil with slight behavioural difficulties the pupil became frustrated by the speed of the game and started randomly clicking buttons in order to try and progress to the next stage. This then increased their frustration when nothing happened. After this incident the pupil seemed to have decreased concentration levels and was easily distractible. In addition to this the pupil seemed generally unenthusiastic about the game, to the point of deliberately misinterpreting the game instructions.

An interesting comment specific to the domain of Dyslexia was that one pupil said that they could not change the background colour of the games. As highlighted in the literature review this is often something that Dyslexic users like to be able to alter to make reading easier. Further investigation into the game uncovered the functionality to change the background and text colour of the games but this was hidden away in a highly tiered menu system and was not obvious to the users. Subsequently the new system should make this functionality much more visible to the users if it is implemented.

Reward Scheme

Comments on the reward scheme within Education City were again varied. Some pupils stated that there were no reward games, whilst others commented that they were good. This confusion could be caused by the fact that the games have a purely visual reward at the end of each game such as a humorous animation, yet some pupils may not perceive such a visual reward to be a “reward game”.

This indicates that a reward scheme that allows the users some kind of control or input may be preferable as this will be a more obvious reward to the users.

Summary

Results from these sessions reinforce many of the aims of this project as it can be seen that the pupils find the games patronising and childish and that this can effect their enjoyment of them. It also highlights the need for a system to be tailored to the needs of dyslexic students by being multi-sensory and also enabling the users to customise the look of game to their specific needs.

However, the new system must ensure that these needs are not met at the expense of the speed of the game as it was clear to see the effects that slow running games can have on users who have attention problems. It is subsequently important to ensure that situations that could lead to such frustrations are avoided where possible.

It also raises important issues to be addressed during the design of the system such as the reward scheme. Designing a theme that is recognised as a “reward” and appeals to all users is likely to be highly challenging.

Another interesting finding was the fact that some issues which seemed to be clear usability problems such as no instructions before or during games did not seem to affect some pupils as they enjoyed learning by trial and error. However, some users did find this frustrating. It is therefore important that both user preferences (trial and error learning vs. instructions) must be catered for within a system.

3.3.2 Heuristic Evaluation

These sessions, as previously mentioned, were carried out with the help of an HCC MSc student. The sessions were carried out over a period of around 2 hours in total and used Neilsens 10 usability heuristics as guidance. These heuristics can be found in the Literature Review Appendix.

The two pieces of software used in these sessions were Education City and Nessy. It was not possible to carry out an evaluation of Wordshark outside of the school, as the software needs to be installed from a CD and this was unavailable. In place of this, a package called Nessy was used. This is a system designed specifically for Dyslexic pupils and has been recently considered by the SENCOs for future use. The version tested in this evaluation was a selection of trial games available online. This subsequently means that the main navigational aspects of the game could not be evaluated.

The main issues that arose during these sessions are presented below. The results have been split up into key areas of usability.

Education City

The game selected to test during this evaluation was Slam Dunk. An explanation and screen shots of this game can be found in the Requirements Appendix section 2.2.

Navigation

- The back button is located in different places on different screens. This raises consistency problems.
- All buttons make the same sound when the mouse is on them. This could cause confusion between the buttons.
- The confirmation screens for the 'quit' and 'again' buttons are the same. It is good that a confirmation screen exists but as they are so similar it would be easy to get them confused.
- There are no tool tips available on the buttons to explain what they do. This is the case throughout the entire game.
- On the Success Tracker page (when a game has been completed) there are two buttons visible, one says 'again' and the other 'quit'. It is not clear what quit will do when it in fact takes you back to the main menu. This should be made clearer because quit indicates that a user will be quitting the whole system.
- During games users can access a verbal prompt of the word to help remind them of the word they are supposed to be spelling. However the speaker icon used to access this is not clearly pointed out.

Input/Output

- On the login page the user cannot automatically enter their details, they first have to click in the text box. In addition to this they cannot log in by pressing return when the details have been entered. They must click on the button.
- When printing a certificate from the Success Tracker page users must enter their name and class. However, if they press return to submit this (which doesn't work) it deletes the text in the field that is currently selected. This is a clear bug rather than a usability problem. However it should be possible to press return to submit the details rather than just working with the button.

User Control

- During the game the graphics and instructions play at a slow rate with no way of speeding them up or skipping them.
- To control the game the users is required to use the keyboard arrow keys to move the shooter around and then space bar to shoot. Ergonomically this is a problem as you could not play with one hand very effectively. Users can use the mouse as an alternative control which is good but this has not been highlighted by the game and subsequently is not obvious.
- When scrolling through the basket ball hoops the hoop selector does not wrap around – so when you get to the left hand side of the page it does not wrap to the right. Instead you must scroll back across the page.
- A plus point is that once a user has correctly shot a ball into the basket they cannot shoot at this again and an explanation of this is displayed. This helps with error avoidance.
- Users are unable to pause or save progress in the game. This is crucial as the game has a time limit.
- Users can access the verbal hint functionality any number of times during the game with no effect on their score. This could cause problems with people cheating.
- When the certificate on the Success Tracker page prints it lets the user choose a printer and print without any verification. This should have a security measure for a teacher to confirm the printing.
- On the main games page a section intended for the teachers is accessible to the pupils.

Presentation

- When logging into the system the login button is not clear. This is because the colour of the button (green) is too similar to the background colour (green grass) so the figure ground contrast is not enough. Instead of noticing the login button the users' eye is drawn to the activities button in the centre of the page which is coloured yellow. However this takes you to a free timed trial of the system.
- The 'back' navigation button is coloured blue and is located at the top left of the page (although sometimes is actually located in a different place). When the background has a blue sky the contrast is not enough and the button is subsequently not highly visible.
- During the games a number of unnecessary graphics are on the screen such as benches and an extra player. These could cause distractions to the user.
- On the games pages there are too many things on the page that are moving. This can distract the users from the game content.
- Within games the timer continues to count down during animations and also if help functionality is accessed.
- Within games the timer is not very obvious and there is no warning when time is running out.
- The scoring system during the games is not very well explained.
- Within the game screen the question number and score areas are seemingly randomly placed and are not aligned properly. This looks untidy and also looks as though something is missing from the screen.
- The animation is very slow during the game including the shooting animation. This could lead to the users forgetting what they are doing.
- During the game the character is at the centre of the screen. This is a good aspect of presentation.
- The confirmation screen graphics for the quit and again buttons are the same. This could cause confusion for users.

Help and Documentation

- On selecting a game from the games page you are taken to an introductory page. Here you can choose to play the game or to view instructions. It was suggested that perhaps the rules should be displayed automatically before a game (with a button to skip them). This would avoid them being accidentally ignored.
- Once in the game there is no way of seeing the game instructions again.

Nessy

The games used during this evaluation were ‘Squish’em’, ‘Whack-a-rat’ and ‘Doggy Din Dins’. These games can be viewed on the Bristol Dyslexia Centre Website. Brief explanations and screen shots of these games can be found in section 2.3 of the Requirements Appendix.

Navigation

- When viewing instructions for the games the word ‘ok’ is used to indicate the ‘continue’ functionality. This meaning may be confusing to users.

Input/Output

- When entering spellings into the games the typing does not occur automatically. Instead users are required to click in a text box. This can be very frustrating in a game with constant input requirements.
- At the end of each game, good feedback is provided to the users.

Presentation

- During the games the incorrect words are placed at the left hand side of the screen but this is not obvious to users as it isn’t mentioned in the instructions.
- In the game instructions there is too much text.
- The background to the game instructions has graphics in it which makes the difficult to read the words accurately. This is especially relevant for dyslexic users.
- The fonts used within the games are quite obscure and can impede reading. This is again relevant to dyslexic users.

User Control

- A user must read through several pages of instructions before playing the game. It should be possible to play the game without viewing all of these.
- In the games the controls are difficult to use. During the Whack-a-rat game the hammer did not always work and during the Squish’em game the flies were moving too fast to be able to click on them.

3.4 Requirements Specification

Requirements are traditionally divided into functional and non-functional requirements. The functional requirements detail what the system should do and the non-functional requirements state what constraints are on the system and development.

When designing an interactive system these will not be sufficient as they are too broad. So instead the requirements will be further broken down into the following groups as detailed by Preece et al. (2002):

- Functional – what the product should do.

- Non-functional – the constraints on the system and its development.
- Data requirements – the data types, accuracy and value of the required data in the system.
- Environmental requirements – the circumstances in which the product will be expected to operate.
- User requirements – the characteristics of the intended users.
- Usability requirements – the usability goals of the product.

What follows is a listing of the requirements gathered during the requirements analysis process. A description of each requirement is listed, as well as the source(s) of the requirement and a rationale for it.

3.4.1 Functional Requirements

1. **Description:** The tutor should, if applicable, provide appropriate scaffolding for learning when covering new material and activities.
Source: Literature review section 2.2.1.
Rationale: Scaffolding provides structured learning. It can be achieved through giving information, prompts and reminders to the learners and then reducing these as the learner becomes more proficient.
2. **Description:** The tutor must provide tasks of an appropriate level for the various abilities of the users.
Source: Literature review section 2.2.1
Rationale: In order to develop, a pupil needs to be provided with tasks of an appropriate level that will stretch them with a challenge but not frustrate them by being too difficult.
3. **Description:** The tutor should provide teaching in phonics.
Source: Literature review section 2.3.2.
Rationale: Phonics is seen as the key to developing reading by many experts.
4. **Description:** The tutor should support a synthetic phonics approach to teaching literacy.
Source: Literature review section 2.5.2.
Rationale: The government's new literacy strategy recommends that this approach is taken. To remain current the tutor must provide for these future changes. *N.B This requirement will not be enforced due to results from further requirements analysis. See section 3.2.1.*
5. **Description:** The tutor should provide opportunities for overlearning whilst ensuring variety occurs during this.
Source: Literature review section 2.4.4.
Rationale: For a process to become automatic usually takes longer in Dyslexic children (Fawcett, 2003) but this may become frustrating. Therefore repeated activities should be made interesting for the pupils.
6. **Description:** Word lists in the game should be adaptable by the teachers.
Source: Requirements analysis section 3.2.1.
Rationale: During interviews the SEN teachers discussed their desire to be able to use the game to support the main activity in the lesson by including words in the game that have been used within this session.
7. **Description:** Teachers must be able to take an active role in setting up the tool.

Source: Literature review section 2.6.2.

Rationale: A study conducted by Underwood (2000) indicated that tutors are more likely to be successful where teachers make interventions into the learning environment of a tutor.

- 8. Description:** The tutor should provide for learning over a prolonged period rather than intensive short term learning.
Source: Literature review section 2.7.1
Rationale: Dix et al (2004) state that learning is more effective over a long period of time. (This is not to say that the sessions will be long – just the time period that they are spread over). Ebbinghaus (cited in Dix et al 2004) also states that the amount learned is directly proportional to the time spent doing this learning.
- 9. Description:** The system must be quick to set up or able to be used effectively without any teacher intervention.
Source: Requirements analysis section 3.2.1 and 3.2.2.
Rationale: When interviewing teachers it was stated that the system must be quick to set up for unplanned use of the game. In addition to this, observations of classroom sessions showed that systems are often used by pupils at the end of the lesson if they have finished their work. In these circumstances it is essential that the game can be ready to play immediately.
- 10. Description:** Any levels within the system must be lockable so that pupils cannot change them.
Source: Requirements analysis section 3.2.1 and 3.3.1.
Rationale: The SEN teachers pointed out that pupils may change the level of games to an easier level and said that this should not be possible as users will not be sufficiently challenged. This was also observed during evaluations of the existing software as many pupils choose to use the easy level of the games they were playing.
- 11. Description:** The system should cover learning content as suggested by the SEN teachers in section 3.3.1.
Source: Requirements analysis section 3.2.1.
Rationale: These areas of learning content are familiar to the pupils and are also recommended by their teachers. It is preferable that the learning content of the game matches current teaching practices and is familiar to the users.
- 12. Description:** The system should provide pupils with repeated exposure to the words within the system
Source: Requirements analysis section 3.2.2.
Rationale: Both the techniques used in these sessions were based on the idea of repeated exposure to words until the words have been placed into long term memory.
- 13. Description:** Any reward scheme implemented should be as universally appealing as possible.
Source: Requirements analysis section 3.2.3.
Rationale: When asked about the reward schemes in current software the opinions of the pupils were mixed. To make the game as motivating as possible it is necessary to make the reward scheme appeal to as many users as possible.
- 14. Description:** The game should allow for users to add their score to a central scoreboard if they wish to.
Source: Requirements analysis section 3.3.1.

Rationale: A number of users requested this functionality. However, this should be optional so that users cannot be embarrassed by other class members knowing their score.

- 15. Description:** Users should be able to keep their scores between sessions on the game.
Source: Requirements analysis section 3.2.3.
Rationale: Users requested this functionality as it is not always available in currently used software packages. Currently, when the software is installed independently on each machine a user will need a separate account on each of these.
- 16. Description:** If a timer is used during a game (or time is being measured in any way) then the counter should pause during help activities and unnecessary animations.
Source: Requirements analysis section 3.3.2.
Rationale: It will be frustrating to users if they the clock counting down during an animation or whilst they are accessing help.
- 17. Description:** The system should provide as much feedback as possible to the users.
Source: Requirements analysis section 3.2.2 and 3.2.3.
Rationale: The TOE-BY-TOE session observation highlighted the importance of feedback to the pupils. This was then reinforced by the pupils during interviews as they said that feedback in the current games was useful to them.
- 18. Description:** Appropriate feedback should be provided to the user when a learning activity is completed.
Source: Requirements analysis section 3.3.2.
Rationale: If a user has made mistakes throughout the game the corrections to these should be made clear to the user so that they can learn from their mistakes.

3.4.2 Non-Functional Requirements

- 19. Description:** The tool should be designed to be used 'little and often' rather than in long sessions.
Source: Literature review section 2.6.2 & Requirements Analysis section 3.2.1.
Rationale: A study conducted by Underwood (2000) found that pupils gained more benefit from a tool when used in short sessions of 15-20 minutes 3 times a week than in longer sessions. This requirement was also backed up by teachers requesting that the system could be used as short starter or plenary activity within a lesson.
- 20. Description:** The game speed must not be affected by slow sound and graphics sequences.
Source: Requirements analysis section 3.3.1.
Rationale: Users often become frustrated from the current software when slow sound and graphics slow down the game.
- 21. Description:** Any teacher controls provided in the system must be inaccessible to the pupils.
Source: Requirements analysis section 3.3.2.
Rationale: In Education City pupils can access teacher documentation that may be confusing for them. In addition to this teacher information could be confidential.

3.4.3 Environmental Requirements

- 22. Description:** If only one control is available in a game (where there is no control redundancy – see requirement 44) the primary method of control should be with the keyboard.
Source: Requirements analysis section 3.3.1.
Rationale: In school classrooms, due to the high levels of computer usage, the mice are often not working correctly through wear and tear. Keyboards appear to be more reliable in this situation. Subsequently to avoid user frustration, the keyboard should be used for control during games where possible.
- 23. Description:** If a printing option is given to users there must be a teacher control/confirmation on this.
Source: Requirements analysis section 3.3.2.
Rationale: A users could accidentally print the work to the wrong printer or print out an unnecessary number of copied. To avoid this the teacher should have ultimate control on the printing.

3.4.4 User Requirements

- 24. Description:** The program must cater for range of abilities and operational stages.
Source: Literature review section 2.2.
Rationale: The children in SEN classes are likely to be between the concrete operational and formal operational stages of cognitive development. This will mean differing levels of cognitive ability and also differing perspectives on the world.
- 25. Description:** System must avoid patronising the users or making them feel behind in their level of literacy.
Source: Literature review section 2.2 and Requirements analysis section 3.3.1.
Rationale: Adolescent egocentrism can lead to a teenager becoming very aware of themselves and can thus cause perceived imperfections to lead to anguish. These findings were backed up by the pupils when asked to comment on existing systems.
- 26. Description:** The tutor must take into account the many difficulties faced by dyslexic pupils.
Source: Literature review section 2.4.2.
Rationale: There are many difficulties (other than reading and writing) that dyslexic children face such as difficulty remembering information and poor concentration. These may affect the way that they learn and also how they interact with the tutor.
- 27. Description:** The system should employ HCI redundancy.
Source: Literature review section 2.7.1
Rationale: As many interaction channels as possible should be targeted with dyslexic users in order to enable as many routes as possible to learning and also to cater for differing learning preferences.
- 28. Description:** The system must where possible meet the requirements of the Dyslexia Style Guide presented in section 2.7.2 of the Literature Review.
Source: Literature review section 2.7.2.
Rationale: This Style Guide combines suggestions from both the British Dyslexia Association and www.dyslexic.com on how to best present information for dyslexic users.

- 29. Description:** The game should allow for users who learn by reading instructions as well as those who learn through trial and error.
Source: Requirements analysis section 3.3.1.
Rationale: When observing users using the current systems it was possible to see that some users needed instructions before playing again, whilst others were happy to learn by trial and error (especially where their reading was not strong).
- 30. Description:** The system should look appealing and eye catching.
Source: Requirements analysis section 3.3.1..
Rationale: Comments made by the pupils during interviews and evaluations were often related to the look of the game. The games that looked better were often preferred by the pupils.

3.4.5 Usability Requirements

- 31. Description:** The system should not overload the pupils with information.
Source: Literature review sections 2.4.2 and 2.7.1.
Rationale: Dyslexic children may have memory problems and this combined with Schneidermans 8th Golden Rule indicates that the display should be kept simple and large amounts of information should not be presented at once.
- 32. Description:** Information must be presented in a meaningful, structured and familiar way.
Source: Literature review section 2.7.1.
Rationale: Information that is presented in such a way is more likely to be remembered as it can be related more clearly to the semantic and categorised structure of long term memory (Dix et al., 2004).
- 33. Description:** The system must allow users to have control over its appearance.
Source: Literature review section 2.7.1 and section 2.7.2.
Rationale: Dyslexic users often like to be able to choose the colour of the background of a system they are using. This is backed up by the Dyslexia style guide as proposed by the British Dyslexia Association and www.dyslexic.com.
- 34. Description:** The navigation of the system should be as simple as possible.
Source: Literature review section 2.7.1.
Rationale: Due to the memory problems that often affect dyslexic users, complex menu sequences can prove difficult to remember. Subsequently, menu options should be as visible as possible.
- 35. Description:** Tooltips or clear labelling should be made available on all navigation buttons. Both options would be preferable.
Source: Literature review section 2.5.7 and Requirements analysis section 3.3.2.
Rationale: Icons can often cause confusion if they are not self explanatory. They should subsequently always be accompanied by some form of explanation to avoid confusion for users.
- 36. Description:** All buttons should be clearly visible, and positioned appropriately.
Source: Requirements analysis section 3.3.2.

Rationale: Buttons in EducationCity were often not eye catching enough to be noticed and placed in obscure positions on the screen. This means that users may not notice them quickly enough.

- 37. Description:** Any non-central functionality such as settings options should be made as clear to possible to users.
Source: Requirements analysis section 3.3.1.
Rationale: When using current software users were not aware of possibilities such as being able to change the appearance settings of the games. As this is useful functionality to dyslexic users, this should not be hidden away in menu systems.
- 38. Description:** The speech output within the game should be clear.
Source: Requirements analysis section 3.3.1.
Rationale: When using the existing software packages problems were encountered by users when they could not here the speech output clearly.
- 39. Description:** Any game controls should represent real world controls as accurately as possible.
Source: Requirements analysis section 3.3.1.
Rationale: When using the existing software packages problems were encountered by the users when asked to input data user an onscreen keyboard that did not have the same layout as a typical 'QWERTY' keyboard. Controls within the game should therefore represent real life where possible to avoid confusing the users.
- 40. Description:** It should be possible to click and skip through animations in the game.
Source: Requirements analysis section 3.3.2.
Rationale: In order to provide user control it is important that experienced users can skip through graphics and explanations that may become repetitive after prolonged use. This will help to avoid frustration.
- 41. Description:** Where games require constant attention or where timing occurs, games should be able to be paused.
Source: Requirements analysis section 3.3.2.
Rationale: If a user is using a game they may get distracted by classroom activities. As they will not wish to loose their place in the game it is important that they can pause the game for when they come back to it.
- 42. Description:** Instructions for a game should be presented prior to it loading, however it must be possible to skip through these.
Source: Requirements analysis section 3.3.2.
Rationale: If instructions are not clearly displayed prior to a game they may be missed. However, if they are very long and require several pages of text or sound clips this may become frustrating to experienced users. Subsequently there should be a way of skipping these.
- 43. Description:** Control redundancy in data entry should be implemented if applicable. For example you should be able to press return to confirm details in a text box as well as pressing an on screen button. The cursor should also automatically default to a text box when the screen is loaded.
Source: Requirements analysis section 3.3.2.
Rationale: This redundancy is common in websites and applications so in order to provide consistency the game should also provide this.

- 44. Description:** It must be possible to control a game with one hand and control redundancy such as both keyboard and mouse control should be possible (see requirement 22).
Source: Requirements analysis section 3.3.2.
Rationale: If a user is only able to use one hand (i.e. if they have broken their arm) then keyboard control using multiple keys that are far apart could be difficult. Additionally players may prefer to use the mouse to control a game if this is more familiar to them.
- 45. Description:** Where applicable it should be made difficult to make basic errors in the game.
Source: Requirements analysis section 3.3.2.
Rationale: To allow for appropriate error prevention users should be stopped from making simple mistakes. For example in Education City users were stopped from throwing a ball into a hoop that had already been filled.
- 46. Description:** Confirmation screens should be used when critical buttons such as 'quit' are pressed.
Source: Requirements analysis section 3.3.2.
Rationale: To avoid users making errors such as quitting the system when they intended another action then they should be asked to confirm their actions. This will enable them to recover from the erroneous action.
- 47. Description:** Unnecessary graphics that are not used directly within a game should be avoided as they can distract and confuse users.
Source: Requirements analysis section 3.3.2.
Rationale: If a graphic appears on a game screen a user may believe that this is part of the game and become confused if they cannot interact with it. Excessive graphics can also be distracting.
- 48. Description:** All game information such as scores, timers should be appropriately grouped and labelled.
Source: Requirements analysis section 3.3.2.
Rationale: If these elements are not grouped together and/or labelled their meaning may be ambiguous to the user.
- 49. Description:** When indicating right and wrong answers/actions, colours must not be the sole indication.
Source: Requirements analysis section 3.3.2.
Rationale: If a user is colour blind they may not be able to distinguish between these colours so would not be able to interpret the results.
- 50. Description:** If an activity has a time limit users should be made aware when the time is nearly running out.
Source: Requirements analysis section 3.3.2.
Rationale: To enable good visibility of system status users should be warned when time is running out so that they are not taken by surprise when the time limit is over.
- 51. Description:** All text in the game must be on a plain background so that users can read it clearly.
Source: Requirements analysis section 3.3.2.
Rationale: This is particularly important for users with reading difficulties. If text has a graphic behind it, it may obscure the letter or word shapes making it more difficult to read.

3.5 Conclusion

The requirements gathering process has been an important process as it has enabled a varied range of requirements to be established that take into account the needs and wants of a range of stakeholders.

The requirements specification in section 3.4 provides a full listing of these requirements, which should all be addressed when implementing the system fully. However, as this implementation is to be to a high fidelity prototype level only, the requirements may not all be addressable at this stage. In addition to this, solutions need to be provided where conflicts exist between these (which is highly likely). These issues will all be addressed during the design of the high fidelity prototype in the next section.

4 Design

4.1 Introduction

Whilst the requirements analysis and specification have provided the problem space for the project it is now necessary to provide solutions to these by moving onto the design of the system. The aim of this stage is to review the requirements and endeavour to provide design solutions to satisfy these, resolving any conflicts that may arise between these.

However, to take this approach alone is not satisfactory when designing a system. Following guidelines and requirements from a single designer's perspective could lead to key issues being overlooked, and it is not sufficient to believe that interviewing users in the requirements process will uncover all the requirements of the system and constraints on its use.

Subsequently it is necessary to take a user centred approach to the design process. This can be done in the form of an iterative process of evaluations with potential users to assess how usable they find the system, whether it meets their needs and how enjoyable they find it.

During this project the evaluations will be done using low fidelity prototyping techniques which will be used to represent early design ideas. These formative evaluations will then be used to feed back findings into the iterative design process.

These formative techniques will begin with informal 'quick and dirty' evaluations where the emphasis is on gaining fast input from the users to ensure that the initial design ideas are in line with their needs and that any early usability issues can be addressed (Preece et al, 2002).

The designs will then be refined and more in-depth evaluations will be undertaken to gain specific feedback and insight from the users to inform more detailed design and usability issues. At the end of this process a final high fidelity prototype will be produced and an in depth summative evaluation will be carried out along with more formal testing.

4.2 Early Design Decisions

Prior to beginning an in depth design of the project some key high level design issues need to be addressed. These are as follows:

4.2.1 Prototype

The final high fidelity prototype will be a fully functioning system. All learning elements will be implemented so that this functionality can be accessed during testing. The only limitation on this will be that the learning content will be restricted to that which is needed during the week of testing that will take place. This may mean a restricted set of levels or word lists.

In addition to this section 4.2.2 provides discussions on a further limitation of the prototype.

4.2.2 Teacher Section

A further key decision taken at this stage of the design process was that, in order to concentrate on the design and implementation of the pupils interface and learning content, a teachers section of the interface would not be implemented during the course of this project.

The reasoning behind this decision is that the focus of the project is on designing a system to be used and tested by the pupils. It is necessary to carry out a final summative learning and usability evaluation on this section with the pupils before adding a teachers section to a programme that might not be of benefit to the main stakeholders – the children.

However, if this evaluation is successful then it will be appropriate to follow a further iterative design process to design a teachers section, again working in close partnership with the SEN teachers

4.2.3 Learning Techniques

A key aim of the system is that it should capture and hold the users' attention and be fun to use. For this reason an early design decision was that the learning interactions would be implemented in the form of games.

Within the literature review the area of tutors was discussed, with the conclusion being that the system would be likely to involve the use of a drill and practice system due to the opportunities it provides for overlearning. However, this is unlikely to provide high levels of motivation to the user (which is of high importance with Dyslexic users) and may cause attention to drift from the game. Subsequently a game application was chosen as it provides a way of bringing interest and motivation to learning. In addition to this overlearning can also be provided within this environment by allowing repeated exposure to words within a stimulating environment.

This idea also originates from the requirements analysis where a pupil suggested the use of games such as Pac Man as a reward game. The reasoning underpinning this was that they are different every time you play them and do not get boring. As a result of this the idea of games was pursued not as an idea for a reward scheme but as the learning interaction itself.

The games within the system will subsequently be designed in the format of an arcade game. A format that requires constant attention by the user and that can be played in short sessions. The aim of this is to provide the users with an activity that they do not 'drift away' from in times of low activity in the game. It will also provide a visually stimulating environment.

4.3 System Architecture

By looking at the requirements for the system it can be seen that there will be five sections of key functionality within the system that will need to be addressed. To gain an understanding of how these areas will link together and to proceed with the design of the system an early plan of the system architecture was sketched. This architecture represents the screens within the game as well as the flow of navigation between these pages. This can be seen in figure 4.1.

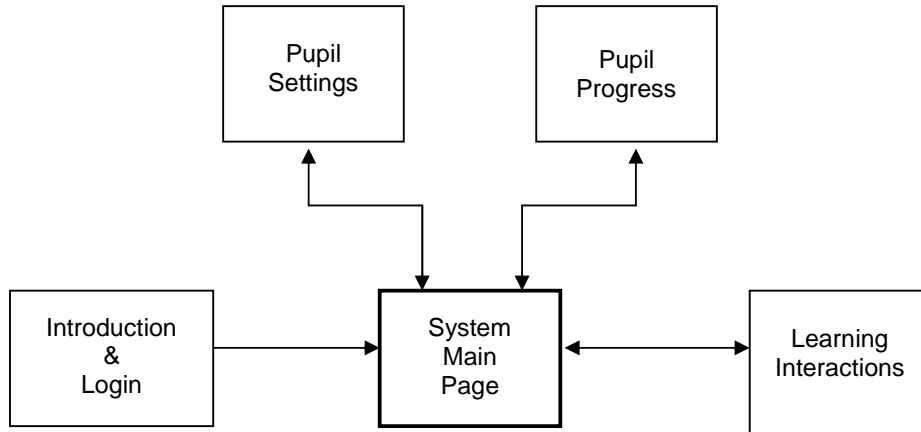


Figure 4.1 High-level System Architecture

The five key areas of the system are as follows:

Introduction and Login – Where new users can gain an introduction to the system and set up their account and returning users can log in.

Main Page – Where users access the games. This is where users will be taken to when they first log into the system and after each game has been completed.

Learning Interactions – These will be split into 3 subsections (See figure 4.2):

- **Game Introduction** – Where the users can view instructions for the game or skip through these to begin the game.
- **Game** – The game itself.
- **Feedback** – Where users can view their final score and corrections to any mistakes they made.

Settings Page – Where users can alter the way that the text and colours of the system look to suit their reading needs.

Progress Page – Where users can view their progress in the games, such as their highest score.

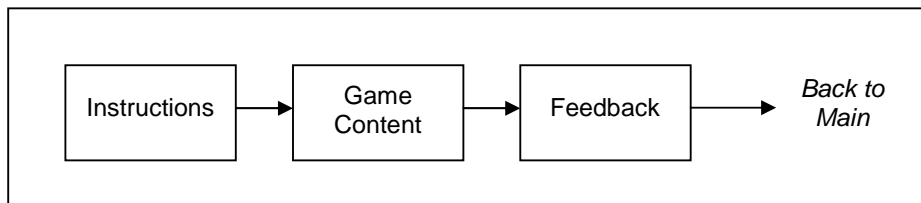


Figure 4.2 Learning Interaction Structure

4.4 Detailed Design

Once the high level design decisions had been made it was possible to move onto the detailed design of the system. What follows is an overview of the main areas of design considered during the creation of the first low-fidelity prototype, focussing particularly on problems encountered with requirements conflicts and areas requiring more complex considerations.

4.4.1 Game Design

Once the idea of games for the learning aspect of the game had been developed it was necessary to try and link learning content into these ideas. Initial ideas included Pac Man style word searches where the player has to 'eat' the correct words, and Tetris style games where users need to drop letters in groups that form words. However, when the specifics of the game implementations were considered it was discovered that these could potentially become very difficult to implement due to the large amount of processing required at the backend of the system. As a result of this, many arcade games on the internet were investigated to see which familiar game styles could have words incorporated into them.

This led to the idea of a space invaders style game with syllables. The idea of the game is that a word will drop down from the top of the screen and the user will need to shoot at the block as many times as the word has syllables in it. So if the word 'interesting' was to fall from the screen it would need to be hit four times before it disappeared.

The use of syllables within the game was chosen as, not only is this area a concept that can easily be applied to a game environment, it is also common problem encountered by dyslexic children when learning to read (Griffiths, 2002). It is a skill that can be beneficial to pupils when attempting to spell words, as they are able to divide it into syllables and attempt the spelling based on these sounds.

A simplified version of this game was subsequently developed for the purposes of the first prototype that involved just one word falling. A space ship was designed to act as the user in the game and basic ActionScript coding was used to make the space ship shoot lasers.



Figure 4.2 Syllable Invaders Screen Shot

This game was named Syllable Invaders and can be seen in more detail in the Design Appendix.

4.4.2 Dyslexia Style Guidelines

One of the key design issues faced during the design of the system was the adherence to the Dyslexia Style Guide proposed in lit review section 2.7.2. Requirement 28 states that these guidelines must be followed where possible. Subsequently at each stage of the game design it was necessary to refer to these guidelines to ensure compliance to these.

The guidelines are a series of suggestions for making a computer based tool easier for Dyslexic users to access. The following issues were subsequently considered in the design of the user interface.

Font

Dyslexia friendly fonts tend to reflect ordinary cursive writing. However, as dyslexic individuals tend to be different in their font preferences it was decided that it needs to be possible for users to be able to alter the fonts used in key textual sections of the game (excluding titles).

The sections identified as key are sections of text that give explanations to the users and any text boxes used within the games as part of the game play.

For these sections it was decided that users should be given control over the font, font colour and background colour. A large size of text (size 12 or 15) will be used within the game at all times so it was felt that it would not be necessary to change the font size as this could cause difficulties with the presentation of the text as large fonts may leave some text missing from the pages.

In order to make the implementation of the settings simpler during the implementation phase, where possible, text has been placed inside a box. This will enable coding of this area to be simpler and also will least affect the overall look of the game when the settings have been changed.

Navigation

Another key area when designing the system was designing a shallow navigation system that will be easy for users with Dyslexia to use. A multilayered model (such as the sub menus in a Windows application) can cause problems for users with memory problems (a common problem for dyslexic individuals) so a simple shallow system is required. To enable this it was deemed necessary to place as many key navigation buttons onto each page. This would be achieved by using a consistent toolbar, which for the purposes of the prototype looked as follows:



Figure 4.3 - Top Navigation Bar



Figure 4.4 - Bottom Navigation Bar

Further Considerations

As well as conforming to the Dyslexia Style Guide, Requirement 26 states that the tutor must also take into account the many difficulties faced by Dyslexic pupils. Many of these difficulties were listed in the literature review within section 2.4.2 and were therefore consulted throughout the design process. One main difficulty highlighted by Goodwin and Thomson (2004) is difficulties with remembering information.

Early design decisions indicate the need for a user log in to the system in order for users to store their scores from the game as well as save their display settings. This could cause problems for Dyslexic users if they are required to remember a password. Subsequently at this stage (although implementation of this functionality will not be included in the low fidelity prototype) it was decided that to assist users in remembering their password a reminder functionality will be provided. When setting up an account users will be asked to add a password reminder. This will then be available to them at login (if they have forgotten their password) to provide them with a memory prompt.

Another of the difficulties listed is a problem with physical coordination. This could cause problems within a game environment if difficult controls are required. Therefore when implementing the games, the controls will be kept to a minimum and made as simple as possible.

4.5 Low-Fidelity Prototype 1

For the purposes of the first low fidelity prototype it was decided that rather than the typical use of a paper based version, the prototype would be implemented on the computer. The reasoning for this is that, as the pupils involved may have attention issues related to their special needs, it would be a more effective session if the system could be represented on the computer. This would provide a more eye catching focus for the sessions and would enable the pupils to more clearly visualise the game. An addition to this it would also provide a higher level of motivation for the pupils enabling an optimal environment in which to gain feedback.

It was also important at this stage to check that the text and backgrounds used in the design were clear to the users. As a computer can affect the display of text and colours considerably, the use of a computer based prototype meant that the display could be evaluated more effectively in its final environment.

Whilst the prototype was created on the computer, it was designed to be a slide show of the game and its screens. This meant that each screen had to be mocked up on the computer but that the functionality was not yet in place. However, the navigation was implemented to give a clearer idea of how the end system would fit together. In addition to this a short clip of a game in action was implemented (See section 4.4.1) as the nature of the game required movement to be perceived by the users.

A series of screenshots from the initial prototype can be found in the Design Appendix section 3.1.

4.5.1 Iterative Prototype Design Session

These sessions were undertaken using a 'quick and dirty' evaluation approach as described in section 4.1. The aim of these iterative design and evaluation sessions was to gain initial feedback on the system, such as the pupils' general reaction and enthusiasm for it.

In addition to this overall aim the sessions were also designed to gain more specific feedback on design problem areas. These areas were as follows:

- The display of text on the screen and how easy/difficult the users find this to read.
- The motivation in the game. Whether a reward scheme would be desired by the users or if the games themselves will provide sufficient motivation.
- Whether the effort to restrict the game theme has succeeded in making the game appeal to an older and larger audience?
- Whether the look of the game is interesting enough to capture the users' attention.
- To establish which settings the user would like to have control of in the game (i.e. text colour).

The sessions themselves took the form of semi structured interviews. The aim of these was to introduce the users to the system by giving them a tour through it. Questions were then asked during the session to gain their feedback. These questions were based around the areas listed above but deviations were made where appropriate. This allowed extra information to be gathered that had not been anticipated.

The sessions were held with a variety of pupils and were carried out either in one-to-one sessions or with a pair of pupils and lasted around 5 – 10 minutes per session. In addition to this, a separate session was held with both a pupil and SENCO together. The aim of this session was to gain additional qualitative data and also to give the SENCO the opportunity to provide feedback related to the system.

Session Findings

- All pupils said that they could read the text on the screens clearly, although in places some small text appeared less clearly as the screen resolution differed to that which the game was designed for. The school computers have a resolution of 800 x 600.
Proposed Solution - The game will be re-implemented onto a screen size of 800 x 600. This size is compatible with higher resolutions but will mean that no text appears small and distorted on the school computer resolutions.
- Although the background caused the text to be clear to read it was commented by some pupils that it was a bit plain and pale. In addition to this some pupils commented that there were not enough pictures in the game.
Proposed Solution - Although Dyslexia guidelines state that no distracting backgrounds and graphics should be used other elements of the game such as buttons and borders could be re-designed to be more eye catching. Small graphics will also be considered for pages where it will not inhibit the readability of the text.
- In the demonstration version of Syllable Invaders some users found that the text block moved too quickly to respond in time (as you could not fire quickly enough).
Proposed Solution - Users will be given control over the speed of key objects in the game.

- When asked whether they would still like a reward scheme in the system some pupils said that they were happy as it was, but others suggested that to have one would be a benefit.
Proposed Solution – A reward scheme will be implemented into the game as it is necessary to provide this form of motivation for users who may not otherwise be motivated to play the game.
- During the more in-depth session with the pupil and SENCO the theme of the game was discussed in detail. It was suggested by the pupil that the space theme could be seen by some as immature (although the intention of the game was to have no theme it had become slightly space based).
Proposed Solution – The theme for the game will need to be reconsidered.
- In relation to the space theme, ideas for a possible reward scheme were discussed. An idea suggested jointly between the pupil and SENCO was to be able to create an alien at the end of the game. You would then be able to view all the aliens that had been created. The pupil stated that this would be good as it involves creativity.
Proposed Solution – A reward scheme should be implemented that involves some form of creativity.
- In general, whilst the pupils' response to the game was polite it was perceived that enthusiasm levels were not high for the game topic.

4.6 Re-Design

As a result of the prototype evaluation sessions it was decided that as there seemed to be both a need for a clearer theme in the games and some form of reward scheme the basis for the game should be re-considered.

Whilst the initial aim had been to refrain from giving the game a theme, in order to make its appeal more universal, this had appeared to remove the users' enthusiasm for the game. It was subsequently necessary to re-evaluate this decision and incorporate a theme into the game.

Discussions with SEN teachers after the sessions highlighted the fact that many SEN pupils are enthusiastic about cars. This is also supported by initial requirements analysis where a number of pupils mentioned that they enjoyed playing car and racing games in their spare time. However, a problem with this theme is that it could be interpreted as a 'boyish' theme. However, there are a three times as many boys than girls who receive additional teaching due to their Dyslexia (www.dyslexia-inst.org.uk) and it was also noted by the SEN teachers that the girls in the lessons also share this interest in cars.

The result of these discussions was that the game would be redesigned with a car theme. The idea for this was that the reward scheme could also be incorporated into the theme.

The basic game premise for the system will now be as follows:

At the beginning of the game users will select a car to represent them during the game. The learning sessions will then revolve around the car theme by incorporating words into racing games or themes such as bumper cars. After a game a user will be rewarded by being able to modify their car by adding car parts and changing its colour.

The system architecture and navigation for the game will remain as it was for the initial prototype apart from where game re-designs require this.

The following sections discuss in further detail the major areas of re-design required for the production of a second low-fidelity prototype.

4.6.1 Reward Scheme Design

As mentioned during the initial re-design discussion, the reward scheme will involve users modifying their car. These modifications will then be preserved throughout the rest of the game.

A selection of car components were subsequently designed within the game, including an exhaust pipe, hubcaps and various car transfer graphics. It was decided that these should be added to the car using a drag and drop, direct manipulation interaction style. This decision was made as it is a natural action to take in this type of environment, and is commonly found in games involving modifications of an object on the internet.

In order to make the screen layout simple, all additions and colour options were placed on a single screen with the car appearing at the bottom. Users are then able to drag and drop the car additions onto the car as they wish. In addition to this, colour buttons can be used to change the colour of the car and its additions.

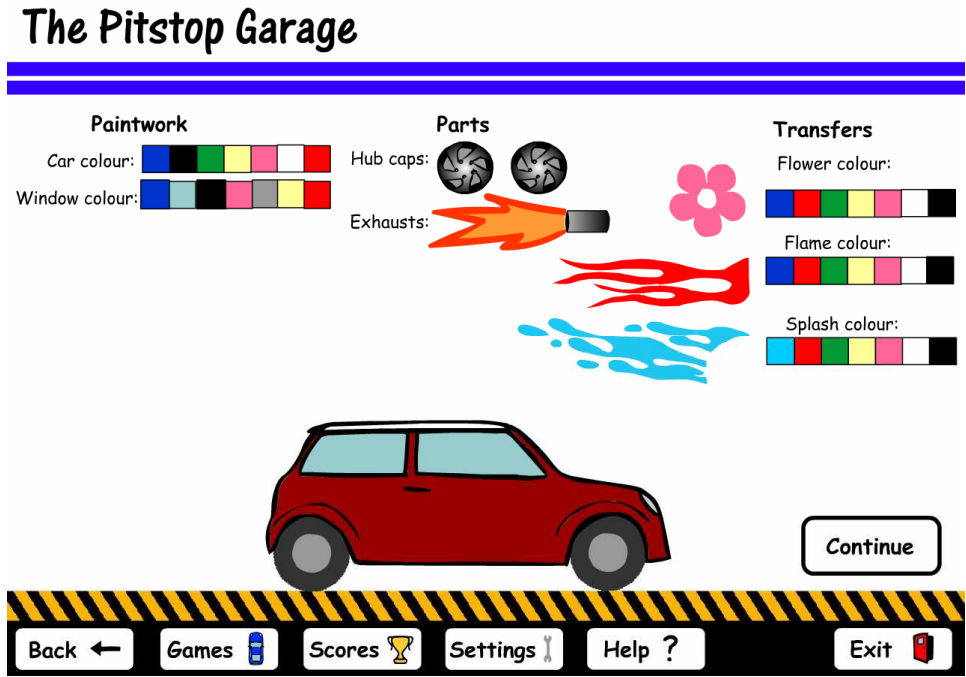


Figure 4.5 Reward Scheme Screen

For the purposes of this low fidelity prototype the additions made to the car will not be preserved throughout the game, although the car colours will. This is because the implementation of such functionality would be complex and therefore will not be implemented until the high fidelity prototype is created. This enabled the focus to be placed on further design of the games and system interface.

4.6.2 Bumper Syllables

This game will be similar in nature to Syllable Invaders as the game will continue to be based around the concept of syllables. However, this concept needs to be transferred to a format that will work with a car based game.

To develop the idea of the game to fit the car theme, it was decided that the concept of bumper cars (or dodgems) could be used. The aim of the game will be for the user to control their car to bump into other cars that contain a set number of syllables.

Initial ideas for this game included making the game screen resemble a bumper cars ride, with all the cars on the screen at once moving around as if being driven. However, this could provide users with too much data to process at once, which conflicts with requirement 31 saying that the system should not overload the pupils with information. Subsequently, it was decided that the cars should appear on the screen one at a time, with them moving down the screen towards the user's car. This will reduce the processing required by the users.

Presentation

To make the game look more appealing for the users, the screen was designed to represent a real road, with white lines in the middle, and double yellow lines at the side. Such graphics are not intrusive enough to be distracting to the users but provide an effective way of creating the illusion that the road is moving (as these lines move during the game).

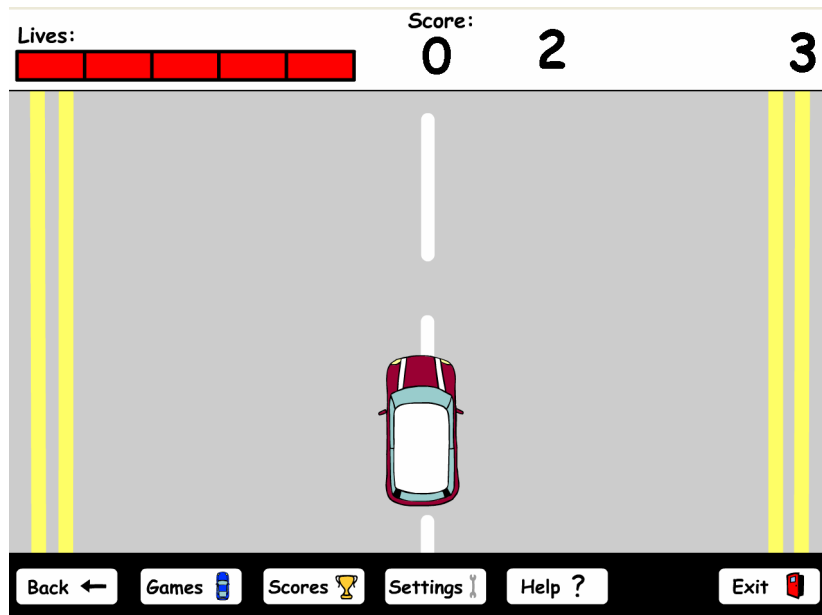


Figure 4.6 Bumper Syllables Screen

Whilst the original premise of the game was to have cars containing the words, it was later decided that the words should in fact appear within simple blocks. This is to avoid unnecessary graphics around the cars which could distract the user from the words themselves.

Scoring

To provide motivation in the game, it was decided to implement a life line. This will be reduced by a set amount each time a user hits the incorrect words (or if they fail to hit a correct word). If the life bar then reaches zero the users will not be able to proceed to the

reward game. The reasoning for this was that without it users may be able to skip through the game without thinking about the words (i.e. by not hitting any blocks, or hitting them all) in order to proceed to the reward scheme.

In addition to this life line, a score was also added to the game, to provide the motivation for users to try and beat their own scores (or their classmates' scores). The points counter will be incremented every time a correct word block is hit.

Feedback

In order to provide some form of feedback in the game it was decided to indicate to users when they had hit an incorrect word. To do this a crash graphic was created. This will be displayed every time a user hits an incorrect word. It also adds further visual stimulation to the game.



Figure 4.7. Bumper Syllables Crash Graphic

Game Summary

The final design of the game for the purposes of the prototype is as follows:

The user's car is centred at the bottom of the screen appearing to move along the screen at a set speed. The score, life bar and syllable number for the game are displayed at the top of the page. The user is able to move the car from side to side but not up or down.

At regular intervals word blocks appear at the top of the screen (accompanied by a sound clip) that the user must either hit (if the syllables match the number set for the game), or avoid (if the syllables do not match).

This word then moves down the screen at a set speed until it hits the bottom. When all blocks for the game have either been hit or reached the bottom of the screen the game is over. If users have not lost all of their lives they are taken to the reward game.

4.7 Low-Fidelity Prototype 2

The aim of this second iterative prototype design and evaluation session was similar to the evaluation of the first prototype, as the new design of the system meant that it was once again necessary to ascertain the users overall opinions of the new theme and reward scheme. However, within this session it was deemed necessary for the users to be able to play the system themselves rather than being given a guided tour. This would enable usability issues such as confusing navigation and controls to be observed. It would also provide an opportunity to assess the level of the learning content.

These sessions followed the same format as the previous sessions with the users working on their own or in pairs for 5 – 10 minutes.

Session Findings

Users seemed much more enthusiastic about the re-designed version of the game with pupils actually asking to have a go at playing the game.

However, the following usability issues were discovered:

- When choosing their car, users were not certain as to how to scroll through the different cars. This may be because the arrows were not clear as buttons as they did not change appearance when the mouse rolled over them.

Design Solution: The arrows were altered so that when the mouse hovers over them so that their changing appearance replicates the behaviour of the rest of the buttons within the system.



Figure 4.8 Car selection button 'up' and 'over' appearances

- When playing Bumper Syllables a number of pupils were frustrated by the speed of the car. The current settings were that the car was set to move at a default speed and users could simply move the car left or right. The pupils asked to be able to make the car accelerate.

Design Solution: The game was updated so that users could use the forward arrow key to make the car move faster. However an upper limit on the speed was implemented so that they could not just skip through the game.

- During Bumper Syllables some pupils found the concepts of lives in the game confusing. The SENCO also stated that this could provide confusion to the users as the concept of a 'life' with a car is not a natural connection to make.

Design Solution: As the game requires some way of stopping users racing through the game without thinking (i.e. not hitting any blocks, or hitting them all), an alternative solution to this was necessary. Early ideas included the use of 'tyres' to represent lives as this has a more natural link with cars. However, with the addition of a leader board in the game (see section 4.8.2), it was decided that all users should be able to reach the reward scheme, as the leader board would provide the motivation to do well within the games. However, to ensure that users do not bump into all cars to get maximum points, it was decided that a point decrease should occur when incorrect hits are made.

- When playing Bumper Syllables, users were not aware of the number of syllables that they should be looking for. This was due to the fact that this number is placed at the very top corner of the screen and is not clear enough to the users.

Design Solution: In order to make the syllable number more clear it was decided that prior to the game starting, a pop-up should appear detailing this to the users. However, as users may forget this the number shall continue to appear at the top of the screen as it did before.

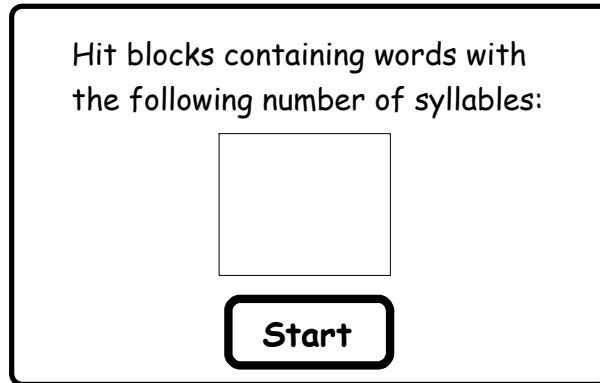


Figure 4.9. Pop-up for Bumper Syllables Game

4.8 Further Design Decisions

After this evaluation session it was necessary to continue the design process, as although much of the system design was still in place, some key areas were yet to be considered.

This section details the final design decisions that were made before moving onto the implementation.

4.8.1 Game 2 – Syllable Grand Prix

In order to provide a greater amount of learning content for the purposes of testing, it was decided that it would be necessary to design a further game.

Game Content

When designing the learning content for this game the SENCO at the school was consulted once more. By working together it was decided that a skill that would work well in a car game would be the ability to choose the correct spelling of high frequency irregular words when the correct spellings are presented alongside common misspellings.

To incorporate this theme into a car related game it was decided that the game would be based around a race track. Initial ideas included the users being set the task of completing a number of laps of the track. On this track they would be presented by a choice of routes, one of these routes would contain the correct spelling of a word whilst the other would be incorrect.

However, to place a whole race track on the screen and then ask users to navigate around it would require complicated controls that could frustrate users. Subsequently it was decided that a single track would move across the screen which would then split into different branches each containing a possible spelling.

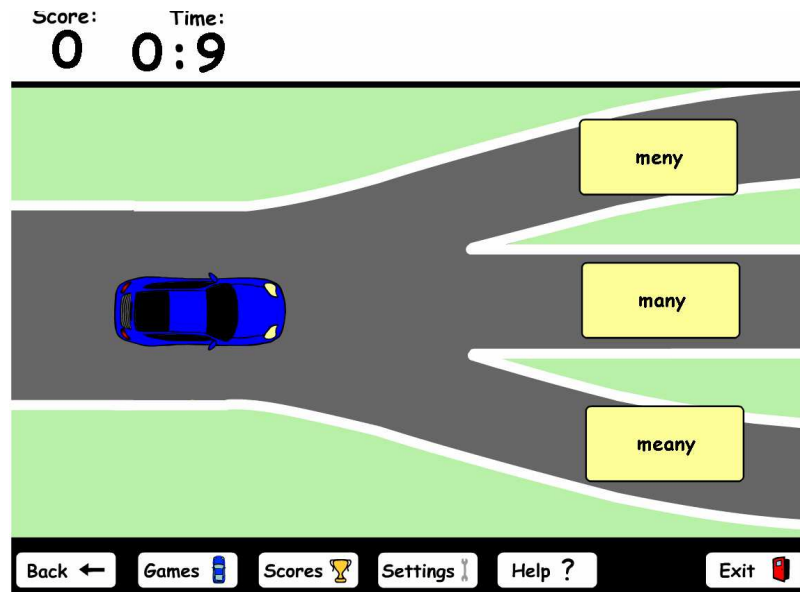


Figure 4.10 Spelling Grand Prix Screenshot

Control

For the car controls it was decided that as words need to be considered carefully (to be able to distinguish between the different spellings), the users should be able to stop the car (which is not possible in Bumper Syllables). This will enable them to clearly view the words without the distraction of movement. In addition to this, the ability to reverse must be possible so that if users have chosen the wrong route they can move backwards to rectify this mistake. In addition to this users will need to be able to move the car up and down in order to control the car through the correct path.

Feedback

In order to provide instant and appropriate feedback within the game (as stated by requirement 17) it was decided that a box should appear on the screen after an incorrect word has been hit to display the correct spelling. This will enable users to see that they have chosen the incorrect spelling, and will allow them to view what the correct spelling should be.

Game Summary

The final design of the game was as follows:

The user starts the game with a stationary car, they can then move the car along the screen (from left to right) until they encounter a branch in the road. They must then choose the route to follow that they believe contains the correct spelling. When all words have been completed, the game ends.

4.8.2 Leader Board

During both the requirements analysis and prototype evaluations the pupils have explicitly requested for a leader board or score board to be added to the system (see Requirement 14). Due to prioritisation issues, this requirement was not addressed during the early designs as it was more important to focus on the game design and subsequent re-design. However, such functionality appears important to the users and will provide motivation to succeed in the games, therefore it was deemed necessary to include this functionality in the final prototype.

One key consideration when designing the leader boards was that although pupils are likely to want to add their score to the leader board if they do well, this may not be the case if the pupil is shy or does not enjoy this competitive nature. Therefore, it was decided that the addition of a users score to the scoreboard should be an optional choice.

It was also decided that to provide additional motivation for the users to perform well in the games, the option of adding a score to the scoreboard should only occur when they have got maximum points in a game. The leader board will subsequently need to be based on times rather than scores, as all scores will be the same.

4.8.3 Learning Content Adaptation

Although the ability for the system to adapt the learning content to the needs of the user is not an explicit requirement in itself, it can provide the solution to Requirement 2 which requires the system to provide content of an appropriate level for the users. For this reason it was decided to implement a basic level of adaptability in the game.

This functionality was designed by referring to the observations of both the TOE-BY-TOE session and the spelling tests (see section 3.2). It was decided that the game should contain wordlists that list words by increasing difficulty (as with the master word lists used for the spelling tests). The users will by default begin encountering words at the beginning of this list (and therefore at the easiest level). They will then continue to encounter these words until they have got them correct 3 times (as with TOE-BY-TOE). When they have got a word correct 3 times, this will then be replaced by the next word in the list.

This functionality means that users will continue to be exposed to words that they have difficulty with, whilst being able to move on to more difficult words where they are finding words easier (and therefore getting them correct). This means that users will only continue to encounter words of a level that challenges them.

This functionality is important as it means that the system will adapt to users needs and so will not rely on a teacher changing these settings. It also removes this option from the users, restricting them from choosing levels that are too easy for them.

4.9 Conclusion

The iterative design has been a highly important process as it has enabled a much clearer understanding of the users' needs and motivation. Through this process it has been possible to refine the design to a stage where implementation of a final prototype can now be undertaken.

5 Implementation

5.1 Implementation Language

Before beginning implementation of the system it was necessary to select a language to carry out this development in.

The key criteria for this language were as follows:

- To provide a tool that will enable the final product to be eye catching and visually stimulating.
Rationale: The tool needs to both capture and hold the attention of the users. This need has been illustrated during the requirements analysis phase, as the look of the system was a common aspect that the pupils judged a system on.
- To provide a system that is fast.
Rationale: Pupils get distracted if they need to sit and wait for a game or its graphics to load, or frustrated if the game play is slow or jumpy.
- To provide an environment that allows for rapid iteration during design and development and does not have a high programming overhead.
Rationale: Due to the iterative nature of the prototyping, the language needs to provide the ability to do this quickly and easily.

Subsequently two programming packages were investigated with regard to these criteria. These two packages were Flash MX Professional 2004 and Java.

In order to investigate these issues, games created using the packages were investigated on the internet as well as simple tutorials introducing the languages.

From these investigations it was discovered that these packages could provide both visually stimulating games and games that ran at an acceptable speed. The decision was subsequently based on the programming overhead that each language would require.

Whilst Java is a language that can support high level graphics it is not naturally suited to the fast and simple development of user interfaces. Conversely, while Flash is a new package to learn, interface development would be fast and effective. In addition to this the Flash development language (ActionScript) is very similar to Java and will provide the means of creating the interactivity in the system.

5.2 System Architecture

The game implementation has been split into 5 distinct sections. These sections have therefore been implemented within separate scenes (a functionality available within Flash MX Professional). This simplified the implementation as it allowed the layers to be scene specific and thus reduced the number of excess layers in each scene. These scenes can be seen in figure 5.1.



Figure 5.1 Scenes within the implementation.

Each scene has been divided into a series of key frames as follows. Each key frame will appear to the users as a new page within the game:

Introduction

Log in page – where users will be able to log into the system with a unique username and password (and will then be taken to the Games Page).

New user – where new users will be able to set up an account.

Instructions – where new users will be presented with a short overview of the main features of the system.

Choose car – where new users can select the car that will represent them throughout the game.

Main

Games page – where users can choose which game to play.

Settings – where users can alter the display settings of the game.

Scores – where users can view their top scores for each game.

Bumper Syllables Leader Board – where users can view the leader board for Bumper Syllables.

Spelling Grand Prix Leader Board – where users can view the leader board for Spelling Grand Prix.

Bumper Syllables

Instructions – where users can view the aim, scoring system and controls of the game.

Game – where users play the game itself

Feedback – where any corrections are displayed to the user and they can add their score to the leader board.

Spelling Grand Prix

Instructions – as with Bumper Syllables.

Game – as with Bumper Syllables

Feedback – as with Bumper Syllables.

Reward Scheme

Garage – Where users can modify their car after completing a game.

The layers used within these scenes can be seen in the Implementation Appendix (section 4.3) along with a detailed overview of the system and its navigation (section 4.1)

5.3 Graphics

In order to produce the graphics within the game the drawing tools within Flash were used.

To reduce the overheads that would occur if all graphics were drawn freehand, the majority of the graphics that appear within the games (the cars and accessories) were adapted from existing photos.

To do this the photos were loaded into flash and traced with the paintbrush tool. The original images were then deleted leaving a line drawing. Any rough lines and imperfections were then smoothed using the smoothing functionality. These images could then be coloured and adapted where necessary before being converted into movie clips to be used within the system.

In order for the cars to be altered in the reward scheme (such as changing the window and body colours) each car graphic has been split into a series of embedded movie clips. The movie clips and their hierarchy are as follows:

- Car (Beetle, Mini, LandRover, Ferrari and Porsche)
 - Main
 - WindowColour
 - BodyColour
 - Wheels
 - Wheel1
 - Wheel2

An example of the Mini movie clip deconstructed into its child movie clips can be seen in figure 5.2.

In addition to the main body of the cars, the same process was followed for each car top again using the hierarchy seen above (excluding the wheels). An example of a car top movie clip can be seen in the Implementation Appendix section 4.5.

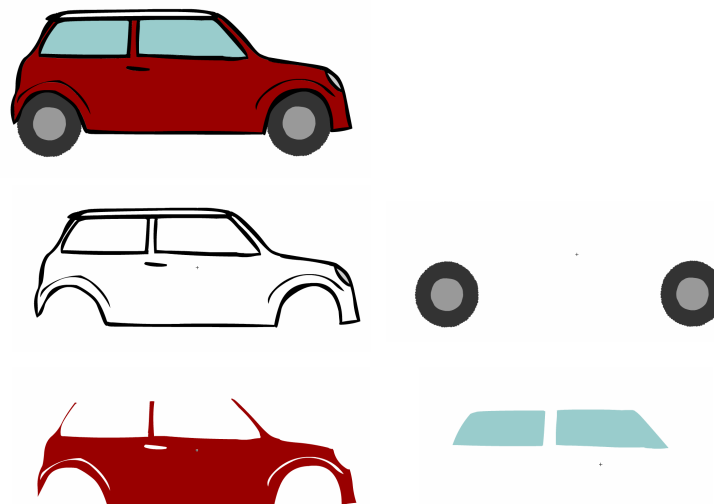


Figure 5.2 Mini Movie Clip and Child MovieClips

5.4 Database Design and Implementation

Due to the need for a large selection of word lists, and the need for users to be able to store scores and car modifications across games it was decided that a database would be implemented as the backend to the prototype. This was felt necessary, as storing this data in arrays could cause a lot of complexity in the game. In addition to this the database will allow for greater flexibility with the data and more robust querying capabilities.

However, when creating the database, due to the nature of the prototype, emphasis was placed on making it simple to access. Subsequently it was decided that it would not be necessary to fully normalise the database, although the repetition of data has been kept to a minimum.

The main functions of the database are to carry out the following activities:

- Store user log in details.
- Store user display preferences.
- Store car details such as colour and additions.
- Store word lists for the games.
- Store each user's progress with the words they encounter in the games.

The tables in the database and their purpose are listed below:

- **users** – to store details of the users such as login details, display settings, and their current level.
- **cars** – to store details of users' cars such as the name and car colours.
- **carAdditions** – to store additions that the users have added to their cars. Each car can have many additions in this table.
- **syllableWords** – to store the words required for the Bumper Syllables. Each word has a set level.
- **spellingWords** – to store details of the words used in Spelling Grand Prix. Again each word has a level as well as one or two common wrong spellings.
- **userSpellings** – to store details of a users progress with each word that they have encountered within Spelling Grand Prix. The number of correct hits and are recorded.
- **userSyllables** – as with userSpellings but for the Bumper Syllables game.
- **syllableScores** – a table used to store any scores and times that have been added to the leader board.
- **spellingsScores** – as with syllableScores but for Spelling Grand Prix.

An entity relationship diagram describing the relationships between the tables in the database can be found in the Implementation Appendix section 4.4.

5.4.1 Database Implementation

The database was implemented in MySQL. MySQL is a popular open source database and was chosen due to its availability.

In addition to this, the use of a server side scripting language was also required to integrate the Flash application with the database. The package for this was PHP and was also chosen due to its availability.

5.5 Integrating Flash, PHP and MySQL

When using external files and databases in Flash it is necessary to load data to and from these files.

To do this within this project the ActionScript class LoadVars was used. LoadVars allows Flash to send variables within a LoadVars object to a specified URL and to load a new LoadVars object (and corresponding variables) back into the script.

This process is carried out by creating a new loadVars object and setting variables in this which will be read by the PHP file. A second loadVars object is then created into which the returned data will be loaded. The initial object is then sent to the PHP file using the sendAndLoad() method. This method posts variables from the loadVars object to a specified URL and loads the server's response to the targeted return object. The variables within this object (as set by the PHP file) can then be used within the ActionScript code.

Below is an example of this code where a selection of words are loaded from the database that match the number of syllables posted to the PHP file.

```
// Create LoadVars object for words and set number of syllables
getWords = new LoadVars();
getWords.syllables = _root.syllableNumber;
// Create LoadVars object to receive data
receivedWords = new LoadVars();
/* Send getWords object to specified PHP file and return variables
to receivedWords object */
getWords.sendAndLoad("getWords.php", receivedWords);
receivedWords.onLoad = function(){
    for(i = 0; i <= 8; i++){
        // Set block to insert data into
        block = eval("_root.block"+i);
        // Get data from received object
        word = eval("receivedWords.word"+i);
        syl = eval("receivedWords.syllables"+i);
        sound = eval("receivedWords.sound"+i);
        ID = eval("receivedWords.word_ID"+i);
        // Fills block with the received words
        block.fillDetails(ID, syl, word, sound, true);    }
}
```

Figure 5.3 LoadVars Code

Within the PHP file, any variables posted into it are accessed by calling \$HTTP_POST_VARS[variable name]. This data can then be used within a query string to be sent to the database for execution. Data retrieved from the database can subsequently be set as the variables of the returned LoadVars object. To do this the variables need to be output from the PHP file as a string. This string contains the variable names and their corresponding values separated by '&'. An example PHP file can be seen below:

```

mysql_connect("mysqlhost.bath.ac.uk", "mnlvs", "*****");
mysql_select_db("csprojWordRaces");
$qqr = mysql_query("select distinct words.word_ID, words.* from
words left outer join userSyllables on words.word_ID =
userSyllables.word_ID where level >= 1 and words.syllables =
'".$HTTP_POST_VARS['syllables']."' and not exists(select distinct
* from userSyllables where words.word_ID = userSyllables.word_ID
and (user_ID = '".$HTTP_POST_VARS['user_ID']."' and
userSyllables.correct > 2)) order by level");
$nrows = mysql_num_rows($qqr);
for($i=0;$i < $nrows;$i++){
    $row = mysql_fetch_array($qqr);
    $rstring."&wrongWord".$i."=".$row['name'];
    $rstring."&wrongWord_ID".$i."=".$row['word_ID'];
    $rstring."&wrongSyllables".$i."=".$row['syllables'];
    $rstring."&wrongSound".$i."=".$row['soundFile'];
}
echo $rstring;

```

Figure 5.4 Example PHP File

5.5.1 PHP File Listing

During the systems implementation a number of PHP files were created. The name and purpose of each of these files is listed in table 5.1.

Table 5.1 PHP File Listing

File	Purpose
addDetails	Adds new user details to the user table.
getCarAdditions	Returns any additions that a user has added to their car. The data is pulled from the carAdditions table.
getCarDetails	Returns a user's car details such as its name and colour from the cars table.
getDetails	Returns user details from the users table when an existing user logs into the system.
getReminder	Returns a users password reminder from the users table when a user has forgotten their password.
getScores	Returns the top scores from the leader board tables.
getSpellings	Returns a selection of words from the spellingWords table to be used within Spelling Grand Prix.
getWords	Returns a selection of words from the syllableWords table to be used within Bumper Syllables.
registerCar	Inserts a new user's car details into the cars table when they select a car.
removeCarAdditions	Deletes the additions from the carAdditions table when a user wishes to clear their car.
saveSettings	Updates the users display settings in the user table when these are altered within the game.
setCarAdditions	Updates the carAdditions table when a user adds an addition to their car in the reward scheme.
setCarColour	Updates a users car colours in the car table when a user changes the colour of their car.
setHubcaps	Updates the hubcaps fields in the user table if a hubcap is added to the car.

setSyllables	Updates or inserts a new field into the userSyllables table to record a user's progress with words in Bumper Syllables.
setSpellings	Carries out the same function as setSyllables for Spelling Grand Prix.
setSyllableScoreboard	Inserts a record into the syllablesScores table when a user selects to do so within the game.
setSpellingsScoreboard	The same function as setSyllableScoreboard for the spellingsScores table.

5.6 Classes

5.6.1 Class Hierarchy

All classes implemented in the system are extensions of the Movie Clip class. This means that they will inherit all methods and properties of the Movie Clip class but that additional methods and properties can also be added.

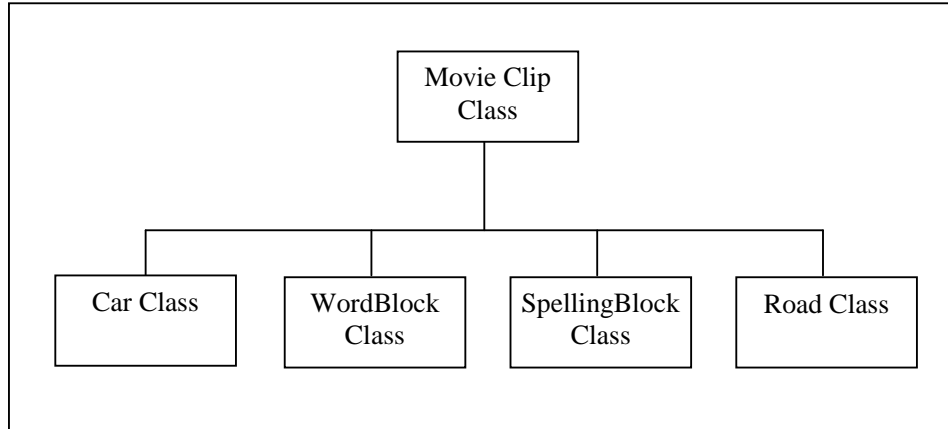


Figure 5.5 Class Hierarchy

In order to make the classes extend the Movie Clip class the following code is used in the Class file:

```
class [classname] extends MovieClip{..}
```

To make the objects within the system instances of this class the Linkage Properties of the objects need to be set in the following way:

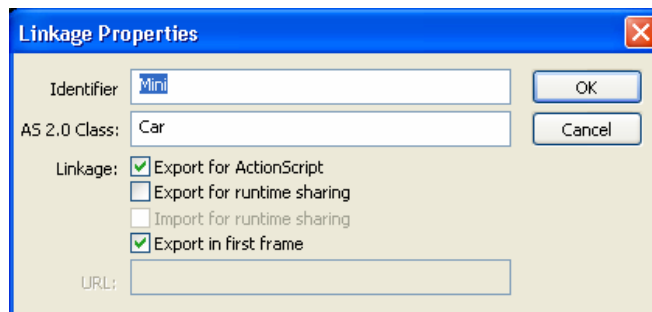


Figure 5.6 Setting Mini Object to be Instance of Car Class

5.6.2 Implemented Classes

During the implementation the following 4 classes were implemented (all are extensions of the Movie Clip class):

Cars

All cars within the game (including the car tops) are instances of this class. The class was implemented to give more control over the appearance and behaviours of the car including:

- Setting up the look of the car on each page, to represent the modifications that the users have made to it.
- Updating the car when in the garage.
- Adding modifications to the car as well as adding these to the database.
- Controlling the car movements during the games.

WordBlock

This class was implemented to control and set the properties of the word blocks in Bumper Syllables by:

- Storing the words and number of syllables.
- Controlling the behaviours when the blocks hit the cars.
- Storing the results of the car hits to the database.

SpellingBlock

This class was implemented, as WordBlock was, to control the properties and behaviours of the word blocks in Spelling Grand Prix by:

- Storing the correct and incorrect spellings of the words.
- Controlling the blocks behaviour and displaying feedback when they are hit by the cars.
- Storing the results of the car hits to the database.

Road

This class was set up purely to control the behaviour of the road segments in Spelling Grand Prix. More details of the functionality can be found in section 5.7.2.

The code for all these classes can be found in the Implementation Appendix section 4.6.

5.7 Game Implementation

What follows is an overview of the key areas of the game implementation.

5.7.1 Bumper Syllables

The main components in this game are the car tops and a selection of word blocks that contain the words the users will be expected to interact with. When implementing the game it was necessary to control both of these component types.

Blocks

The movement for these blocks is a basic down movement. In order to implement this the blocks were set to increment their y coordinate by a set amount on each clipEvent(EnterFrame) event. This causes the blocks to appear as though they are falling down the page.


```

onClipEvent(enterFrame){
    // if block has been activated
    if(this.hasStarted() == true){
        // move the block down the screen
        this._y = this._y + _root.speed;
    }
    ....
}

```

Figure 5.7 Block Movement

The speed of the fall is determined by the `_root.speed` variable which represents the current speed of the game. When the game loads the default speed is set to 1.5 but increases when the users press the up direction key on the keyboard. On each press of this button the speed increases by a set increment of 0.5 (the value of `_root.increment`) until it reaches a maximum of 6 (`_root.maxSpeed`).

```

// if up key pressed
if (Key.isDown(38)) {
    if(_root.speed <= _root.maxSpeed){
        _root.speed = _root.speed + _root.increment;
    }
}

```

Figure 5.8 Block Speed

Conversely if the user presses the down direction key on the keyboard the speed is set back to 1.5. This allows the users time to think about a word if necessary.

In addition to the blocks speed being controlled in this way the background elements such as the road markings are also controlled in this way. This creates the illusion that the road is moving beneath the car.

Car Controls

During Bumper Syllables the movements of the cars has been limited to left and right movements. These movements are controlled by the pressing of the left and right direction keys on the keyboard and has been implemented in the same way as the blocks, using the `onKeyPress` method to react to keys being pressed. An issue that arises with these controls is that the car must not be able to disappear off the side of the screen. It was therefore necessary to insert code to stop this occurring. This was done in the following way:

```

if ((this._x - (this._width/2)) <= 0){
    this._x = 0 + (this._width/2);
}
if ((this._x + (this._width/2)) >= 550){
    this._x = 550 - (this._width/2);
}

```

Figure 5.9 Car Movement Restrictions

This stops the car from ever having x coordinates that are outside the game screen.

Collision Detection

In order for the game to work correctly the cars and blocks need to be able to 'collide'. This means that the computer needs to be able to recognise when one object is hitting another otherwise the cars and blocks will simply carry on moving through each other.

This functionality was implemented using the `hitTest()` method of the `MovieClip` class.

The `hitTest()` method returns true if the movie clip in question intersects with or overlaps the bounding box of the target object that has been sent to the method.

```
if(this.hitTest(eval(_global.car+"Top"))){
    // hit test event handling
    ....
}
```

Figure 5.10 Car and Block hitTest()

This code determines if a block has collided with the currently selected car and then executes the code to deal with this collision. In `Bumper Syllables` this code is within the `WordBlock` class (See section 5.6.2).

The following method within this class is used to deal with a collision between a block and the car.

```
public function carSyllableCheck():Void {
    // decrement the block counter
    _root.blockCounter --;
    // check if the syllables of the block match the game
    syllables
    if(match == true){
        // increment the score, add to correct array and
        remove the block
        _root.points++;
        this.pushCorrect();
        this.removeBlock();
    } else {
        // decrement the score, add to incorrect array and
        remove the block
        _root.points--;
        this.missed();
        this.pushIncorrect();
        this.removeBlock();
    }
}
```

Figure 5.11 carSyllableCheck method within the WordBlock class

In addition to this, within the game itself the following code is called which copies the crash movie clip and sets its coordinates to be where the block was (as this is removed from the screen). This allows users to see when they have hit an incorrect block.

```
duplicateMovieClip(_root.crash1, "crash2", 100);
_root.crash2._x = this._x;
_root.crash2._y = this._y-10;
```

Figure 5.12 DuplicateMovieClip Code

Loading Words

When the game first loads it needs to populate the blocks with words that are stored in the database.

The full details of this interaction with the database are discussed in section 5.5. However once the words have been loaded, the fillDetails() method of the WordBlock class is called which populates the blocks with the words and sets their 'match' property to true (if the syllable number matches the game number) and false otherwise.

Once this has been completed the blocks begin to move down the page in a random order (see Implementation Appendix section 4.7 for the random number function used for this purpose). These blocks fall at a set interval which, like the block speed, is controlled by the speed variable of the game (so that blocks fall more quickly if the speed is higher).

5.7.2 Spelling Grand Prix

As this was the second game to be implemented the process was much faster as much of the game's functionality is similar or identical to that of Bumper Syllables. Subsequently, what follows is a discussion of the areas which differed.

Once again in the game the main components are cars and word blocks, however this time the blocks took their behaviour from the SpellingBlock class as the properties required are of a different nature.

Car Control

This aspect of the game implementation differs from Bumper Syllables due to the increased movement of the car.

In this game, the cars can move forwards (when the right direction key is pressed), up (the up direction key), down and backwards (when the left direction key is pressed). These keys are used as the car is moving across the screen and the keys are therefore more representative of the on screen movement.

In addition to these movements the cars can also move diagonally up and down (when the forward key is pressed at the same time as up or down). To make this movement more realistic the cars were made to rotate on these key presses so that they face the direction that they are travelling in. This was done by using the following code:

```
if (Key.isDown(39) && Key.isDown(38)) {
    this._y = this._y - 0.8;
    this._rotation = this._rotation - 10;
}
if (Key.isDown(39) && Key.isDown(40)) {
    this._y = this._y + 0.8;
    this._rotation = this._rotation + 10;
}
```

Figure 5.13 Spelling Grand Prix Car Control

This causes the cars to rotate as well as move forwards.

As well as controlling the basic movements of the car it was necessary to stop the cars from leaving the road. In order to do this the road sections were surrounded by a series of invisible blocks that were then grouped into one movie clip. The hitTest() method was then used to determine if the cars were colliding with the blocks within this movie clip. Code is then executed that restricts the car from moving any further into the blocks.

Word Blocks

As discussed earlier, the blocks within Spelling Grand Prix are instances of the Spelling Block class and contain either the correct spelling of a word or an incorrect version. Rather than having independent movement these blocks are a child movie clip of the Road movie clips (which are explained in the next section) and therefore move with the road.

Road Movement

In order to create the effect of a continuous road a series of road movie clips need to move across the screen at set intervals. These road movie clips can be of two types:- a three route road (for words with a choice of three possible spellings) and a two route road (for words with two possible spellings).

These road movie clips are both instances of the Road class which is described in section 5.6.2.

At the start of the game words are loaded into 8 sections of road (which lie to the right of the visible screen). These words are loaded into the blocks in a random order so that users cannot memorise the position of the correct spellings.

As the users moves the car these road sections then begin to move. The first of these movie clips (again randomly selected) will move across the screen and when it reaches a set point it then calls a method that activates the next road clip. This process continues until the 8th and final section of road has been activated. When this reaches the specified point the section of road containing the finish line is loaded.

Feedback

A key aspect of this game was the implementation of an instant feedback process. This was carried out by using a movie clip which was designed to look like a pop-up screen (see figure 5.14).

This pop-up is set to be invisible at the start of the game. However, when a user collides with an incorrect block, a method is called that sets the pop-up to be visible for a set length of time. A dynamic text box value within the pop-up is then set to show the correct spelling of the word.

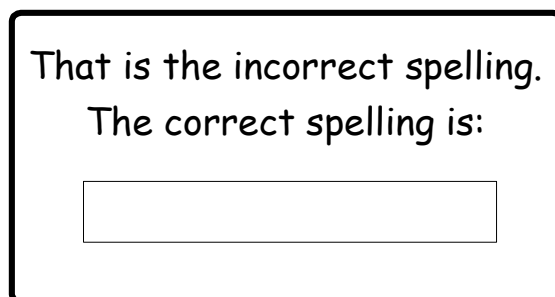


Figure 5.14. Spelling Grand Prix Feedback Popup

5.7.3 Sound Implementation

Both games within the system require the use of sound. This provides a multi-sensory output for the users and will hopefully assist the learning process.

In order to implement the game sound, each word has to have a sound clip to accompany it. To do this, voice clips of the words were recorded using Nero Wave Editor and converted to mp3 format using mp3PRO. An mp3 format was necessary as, to allow for additional words to be added to the game more easily, the sound was loaded into flash from an external source rather than from the object Library. This requires the sound files to be stored in mp3 format. For the purposes of the prototype, all word sound files were stored in a 'Words' folder within the same directory as the system's .swf file.

As well as recording the sound files, a field in each of the word tables within the database (spellingWords and syllableWords) was set to store the sound file of the word. This data is then loaded into the games along with the other word details.

Within the games, the following code is used to load the sound files from their location:

```
sound = new Sound(); //create sound object
/*load sound from Words folder where the file name matches the
loaded sound file location. */
sound.loadSound("Words/"+this.soundLoc, true);
sound.play(); // play sound
```

Figure 5.23 Sound loading code

This code creates a new sound object which then loads in a sound from the Words folder and plays this. Within the games, this code is called when the words appear on the game screen.

5.8 Reward Scheme Implementation

Another complex stage of implementation was the development of the car modification reward scheme. In the initial prototype, this functionality was implemented using a simple drag and drop functionality. However, this did not allow the modifications in the car to be preserved throughout the rest of the game.

To implement this, it was necessary to store the modifications to the database, as well as load these modifications onto the cars on the subsequent screens of the game.

The basis for this functionality was as follows:

Car and Window Colour

As detailed in section 5.3 the cars were designed in such a way that the body colour and window colour of the cars are separate movie clips embedded within the main car movie clip. Therefore, in order to change the colours of these movie clips, the following code was used (this code can be found within the Car class):

```
public function setBodyColour(newColour):Void {
    ....
    bodyColour = newColour;
    // create a colour object and set target as the car body
    // colour
    var carColour = new Color(main.colour);
    carColour.setRGB(newColour);
}
```

Figure 5.16 Car Class setBodyColour function

This creates a new colour object with a target of the cars body colour movie clip (main.colour). It then sets the RGB value of this to the colour string that is passed into the function. It also sets the car's body colour property to the same value. This will be used later.

This process detailed above changes the colour of the car body on the screen. However, in order to write this to the database when the user saves these changes, the following code is used:

```
public function setColours(user):Void{
    var setColours = new LoadVars();
    setColours.name = user;
    setColours.carColour = bodyColour;
    setColours.windowColour = windowColour;
    var Results = new LoadVars();
    setColours.sendAndLoad("setColours.php", Results);
    Results.onLoad = function() {
        if(Results.added == true){
            ...
        }else{
            ...
        }
    }
}
```

Figure 5.17 Car Class setColours function

This writes the cars bodycolour and windowcolour values to the car table of the database using the LoadVars functionality detailed in section 5.5.

Adding Modifications

In addition to changing the cars colours, the modifications that the users have dragged onto the car need to be recorded so that they can be re-loaded on subsequent screens. To do this the x and y coordinates of the components, as well as the chosen colours need to be recorded. This is done using the following code:

```
public function setAdditions(userName, partName, x, y, colour){
    var setAdditions = new LoadVars();
    setAdditions.carID = _global.car_ID;
    setAdditions.userName = userName;
    setAdditions.partName = partName;
    setAdditions.x = x;
    setAdditions.y = y;
    setAdditions.colour = colour;
    var test = new LoadVars();
    setAdditions.sendAndLoad("setCarAdditions.php", test);
    test.onLoad = function() {
        if(test.added == true){
            ....
        }else{
            ....
        }
    }
}
```

Figure 5.18 setAdditions Function

This code is called for each addition that has been placed on top of the car (or in close proximity to the car) using the following code:

```
_root.attachObjects = function() {
  if ((eval(_root.flames._droptarget) == _root.carTarget)
    || (eval(_root.flames._droptarget) ==
    eval(_global.car).main) ||
    (eval(_root.flames._droptarget) ==
    eval(_global.car).main.colour) ||
    (eval(_root.flames._droptarget) ==
    eval(_global.car).main.windows) ||
    (eval(_root.flames._droptarget)._parent ==
    eval(_global.car)) ||
    (eval(_root.flames._droptarget)._parent ==
    eval(_global.car).wheels) ){
    eval(_global.car).setAdditions(_global.user,
    "flames", _root.flamesx, _root.flamesy,
    _root.flamesColour);
  }
  ..(Repeat for all components)..
}
```

Figure 5.19 Code to Check DropTargets

This code is called when a user saves the changes to their car. The code checks if the 'flames' object has been dropped on any areas of the car or a box that is placed round the car (so that users can place modifications around the car and not just on it). If it has, then the above setAdditions() function is called.

In order to retain these changes it is necessary to load them on all the subsequent pages that the car appears on. To do this the additions are loaded from the database using the following code:

```
var getAdditions = new LoadVars();
  getAdditions.name = user;
  var Additions = new LoadVars();
  getAdditions.sendAndLoad("getCarAdditions.php", Additions);
  Additions.onLoad = function() {
    ...
  }
}
```

Figure 5.20 Code to Load Additions

The additions are then attached to the cars from the returned LoadVars object (Additions) using the following code:

```
eval(_global.car).attachMovie(eval("Additions.name" + i),
eval("Additions.name" + i)+i, i+50);
```

Figure 5.21 Code to Attach Additions

This attaches the specified type of movie clip to the car from the components library (using its linkage identifier). It also gives the object an instance name, and a layer number.

In addition to this, the object's x and y coordinates and colours are set using the following code:

```
var target = eval("Additions.name" + i)+i;
eval(_global.car+"."+target)._y = eval("Additions.partY" + i);
eval(_global.car+"."+target)._x = eval("Additions.partX" + i);
if(eval("Additions.colour" + i) != "null"){
    var colour = new Color(eval(_global.car+"."+target));
    colour.setRGB(eval("Additions.colour" + i));
} else {
    ....
}
```

Figure 5.22 Setting Additions' Coordinates

However, as this loading process can take a long time to complete this is only carried out on the 'Games Page' (as this is the page that loads after the garage section). On all subsequent pages, the data is loaded from a global array that is populated when the data is initially pulled from the database on the 'Games Page'. This is much faster and saves processing power.

The code for this is again stored in the Car class and can be found in the Implementation Appendix section 5.6.

5.9 Learning Content Adaptability Implementation

As discussed during the design section it was decided to implement a simplified form of adaptability within the system to tailor the words that each user encounters within the games to their ability.

In order to implement this, two tables within the database were set up to support the functionality:- userSpellings and userSyllables. The purpose of these tables is to record a pupil's progress with each word that they encounter.

Each table contains a user_ID and word_ID field, and a 'correct' field. For each word that is encountered during the games a record is placed into the corresponding table where the current user_ID and the word_ID serve as a primary key (in the case that a record already exists, the existing row is updated). If the encounter is correct the 'correct' field is incremented.

As decided during the design, these tables will subsequently need to be analysed when loading words into the games to ensure that no words are loaded that the user has previously got correct three times. The query used for this functionality, therefore, needs to select only those words that have not been encountered before by the user (and therefore do not exist in the corresponding table) or those words where 'correct' is greater than 3. This was implemented using the following query (this is taken from the Spelling Grand Prix game):


```

SELECT DISTINCT spellingWords.word_ID, spellingWords.* FROM
spellingWords LEFT OUTER JOIN userSpellings ON
spellingWords.word_ID = userSpellings.word_ID WHERE level >=
[level] AND NOT EXISTS (SELECT DISTINCT * FROM userSpellings WHERE
spellingWords.word_ID = userSpellings.word_ID AND (user_ID =
['user_ID'] AND userSpellings.correct > 2)) ORDER BY level;

```

Figure 5.23 Adaptability Query Example

5.10 Display Settings Implementation

One key area that needed to be implemented within the final prototype is the ability for users to be able to change their display settings. This enables users to change the font, font colour and background colour of any key text within the system.

In order to implement this, it was necessary to convert all areas of text background colour into movie clips. This enables the colour of these objects to be changed. In addition to this it was also necessary to convert all key areas of text into dynamic text fields. This enables the font and font colour to be changed easily.

Within the settings screen a series of buttons are used to change these settings. The changes are then demonstrated using a preview pane (a background movie clip and a dynamic text field). When the buttons are pressed the following sections of code are used to update the preview pane:

```

on(release){
    _root.setFont("Times New Roman");
}

_root.setFont = function(font){
    var textFormat = new TextFormat();
    textFormat.font = font;
    _root.text.textField.setTextFormat(textFormat);
    _root.fontName = font;
}

```

Figure 5.24 Font button code and corresponding function

```

on(release){
    _root.setTextColour("0x000000");
}

_root.setTextColour = function(colour){
    colourObject = new Color (_root.text);
    colourObject.setRGB(colour);
    _root.textColour = colour;
}

```

Figure 5.25 Font colour button code and corresponding function

```

on(release){
    _root.setBackgroundColour("0x000000");
}

_root.setBackgroundColour = function(colour){
    colourObject = new Color (_root.background);
    colourObject.setRGB(colour);
    _root.backgroundColour = colour;
}

```

Figure 5.26 Background colour button code and corresponding function

The following code (see figure 5.26) is then called when a user chooses to save their settings. This saves the settings to the database whilst also storing the settings in three global variables (`_global.textColour`, `_global.backgroundColour` and `_global.fontName`). These global variables are used throughout the rest of the game to update the display settings. When the user next logs into the system, their settings are loaded from the database into the same global variables.

```

on(release){
    _root.changeSettings(_root.fontName, _root.fontSize,
        _root.textColour, _root.backgroundColour);
    gotoAndPlay("GamesPage");
}

_root.changeSettings = function(font, size, textColour,
backgroundColour){
    _global.textColour = textColour;
    _global.backgroundColour = backgroundColour;
    _global.fontName = font;
    var saveSettings = new LoadVars();
    saveSettings.fontColour = textColour;
    saveSettings.backColour = backgroundColour;
    saveSettings.font = font;
    saveSettings.user = _global.user;
    var test = new LoadVars();
    saveSettings.sendAndLoad("saveSettings.php", test);
    Details.onLoad = function() {
        ....
    }
}

```

Figure 5.27 'Save Settings' button code and corresponding function

When subsequent pages load within the game, the following code is used to set up the colour of each of the background movie clips as well as the font and font colour for each text box on the page:

```

colourObject = new Color (_root.settingsBackground.colour);
colourObject.setRGB(_global.backgroundColour);
colourObject = new Color (_root.text1);
colourObject.setRGB(_global.textColour);
var textFormat = new TextFormat();
textFormat.font = _global.fontName;
_root.text1.text.setTextFormat(textFormat);

```

Figure 5.28 Display Settings Code

This code is also included in the SpellingBlock and WordBlock classes so that these blocks also adhere to the users preferred settings.

5.11 Testing

Throughout the implementation process individual components of the system have been tested on their own before being integrated with the rest of the system. This was due to the complex nature of some areas of the implementation and helped ensure that each section functioned as intended before the system was integrated, to avoid complications at later stages.

An example of this testing was when integrating flash with the database backend of the system. During implementation of these areas, all queries were tested within the MySQL command line interface to ensure that these were retrieving the correct data from the database. Once this was established, the queries were then placed into the necessary PHP files which were then tested on the internet to ensure that the connection was functioning correctly and the data was being output as intended.

Once these files were functioning correctly they were then integrated into ActionScript with the data firstly being output in dynamic text boxes before being used within the system.

This process allowed any intermediate problems to be dealt with and if necessary fixed before becoming part of a complex system that would be much more complicated to debug.

5.12 Publication

To allow users to access to the system at both school and home, and to allow users to store scores and settings between games, it was decided that the system would be published on the internet. This also allows the system to connect to the database and load all necessary data.

In order to publish the site, the systems .fla file was run and saved as a .swf file. These files can then be run on the internet as long as the correct flash player is installed. As the system was created in Macromedia Flash MX Professional the corresponding .swf file requires Flash Player 7 to function correctly. This Flash player is available on all the school machines and can be easily downloaded to a user's home computer if necessary.

Once the .swf file was created, this (along with all required PHP and sound files) was placed on the university web server. The game can subsequently be accessed at the following address:

<http://www.bath.ac.uk/~mn1vs/WordRaces.swf>

5.13 Conclusion

Having carried out an in depth implementation process it is now necessary to evaluate this high fidelity prototype. Details of this can be seen in section 6.

A large selection of screenshots from this prototype, as used in the evaluations, can be found in the Implementation Appendix section 4.8.

6 Evaluation

6.1 Introduction

During the design process formative evaluation has been carried out using low fidelity prototypes to test design ideas and to identify early usability problems.

Now that the final high fidelity prototype has been implemented it is necessary to carry out a summative evaluation of this with the potential users of the system. Here the focus is on establishing how well the design fulfils the users' needs, whether they enjoy using it and whether the learning content is effective in its intended environment.

To assist with the planning of the evaluations the DECIDE framework (Preece, 2002) has been used. This is a framework that is used to guide evaluations by establishing a series of questions as follows:

1. Determine the overall goals that the evaluation addresses.
2. Explore the specific questions to be answered.
3. Choose the evaluation paradigm and techniques to answer the questions.
4. Identify the practical issues that must be addressed such as selecting participants.
5. Decide how to deal with the ethical issues.
6. Evaluate, interpret, and present the data.

(Preece, 20002)

Section 6.2 details how the questions have been addressed during the planning of the evaluation sessions.

6.2 Planning the Evaluation

6.2.1 Determine the overall goals that the evaluation addresses

The overall goal of the evaluations as discussed in the introduction are to determine the usability of the system, whether the users enjoy using it, whether they are motivated by it and finally whether the learning content of the system is effective in improving the pupils literacy skills.

6.2.2 Explore the specific questions to be answered

The goals of the evaluation can be split into more specific questions that the evaluation should seek to answer.

Is the system easy to use?

- Are the main instructions easy to follow?
- Is the navigation simple and clear?
- Are the navigation buttons clear in their purpose?
- Are the individual game instructions clear?
- Are the game controls easy to use?

Do the users enjoy using the system?

- Do the users enjoy using the system?

- Do the users enjoy the style of the games?
- Do they like the way the game looks?
- Do the users like the theme of the game?
- Would the users want to use the system again?

Does the system motivate the users?

- Do the users play the games to get the rewards?
- Is the reward scheme enjoyable for all users?
- Do the users feel motivated by the presence of a leader board?
- Do they try harder to get onto the leader board?

Is the learning content of the game effective?

- Does the pupils performance improve after playing the game?
- Is the improvement greater than if they had received normal teaching?
- Do the games help improve the skills that they were intended for?

6.2.3 Choose the evaluation paradigm and techniques to answer the questions

In order to best answer the questions in section 6.2.2 a series of paradigms and techniques will be required. This will be a combination of observations, questioning and tests of performance.

To gain an understanding of the systems ease of use, usability testing techniques will be required. However, this will not be full usability testing in the sense of controlled laboratory experiments, as the users will be in their natural environment and the data collected will be of a qualitative nature. The technique used in this situation will be as follows:

Users will be asked to explore the system for a set amount of time whilst their actions are observed. To establish if certain functionality in the system can be used effectively without problems, the users will then be given a set of structured tasks to carry out which will cover the key areas of the system. Their responses to these tasks will once more be observed and any problems recorded.

Throughout the free exploration the users will be questioned using a 'quick and dirty' technique to gather their opinions on the system. This will answer questions on their enjoyment levels and motivation.

To answer questions on the learning content of the game, a more formal evaluation approach will be taken. This will test the performance of a set of users using the system (the experimental group) against the performance of a set of users receiving normal teaching (the control group). Performance will be measured using a pre-prepared test (see section 6.2.4 for more details).

In order to make these evaluations as accurate and fair as possible the following approach will be used:

Two sets of matched participants will be selected. Both of these groups will then be given the pre-prepared test and their results recorded. The experimental group will then use the system for a period of time while the control group receives normal teaching. At the end of this period, both groups will be re-tested and their performances compared.

To ensure that taking the test itself does not affect performance on the second test (through practice effects), a third counterbalanced group will be used. This group will be divided into two (again with matched participants). The first half of the group will take the test and then play the game whilst the other half will play the game and then take the test. The results of these tests will then be compared with the results of the experimental and control groups to assess if practice effects have been instrumental in the results.

6.2.4 Identify the practical issues that must be addressed

Selecting Participants

For the learning content tests participants were selected by the SENCO to try and match the expected user population. They were then matched by their ability so that, in each group, where possible, there were even mixes of abilities (for example - a matching number of statemented pupils).

In most testing the sample size must be large enough to be representative of the population in order to gain statistical significance. However, within special needs there are large variations and individual differences among the pupils meaning that this cannot be as applicable. In addition to this, due to the number of SEN pupils in the school and the number of test groups needed, the final number of participants in the experimental group was 8. This means that in terms of statistical testing, the sample size is not big enough to test for significance. Therefore statistical tests of significance will not be applied to the results.

Designing the Learning Content Test

In order to design a test of an appropriate level and to avoid ceiling effects (where participants get all questions correct) the test was designed in partnership with one of the school's SENCOs.

The test was designed to assess the learning content of both of the games within the tool. Subsequently the test was divided into sections that would assess the skills targeted by each of these games.

Spelling Grand Prix aims to improve a pupil's ability to spell high frequency irregular words by exposing the users to both correct and incorrect spellings of the words. The first part of the test was therefore designed to test their spelling of a selection of these words. The full test can be found in the Evaluation Appendix.

Bumper Syllables aims to improve a pupil's ability to count the number of syllables in a word. This is a skill that can aid spelling, rather than a spelling skill itself. Therefore the second part of the test was designed to test the pupils' ability to do this by asking them to list the number of syllables in a variety of words that do not feature in the game.

Finally, to test if this skill has subsequently helped the pupils to spell, a further 3 words were included that are words that can benefit from the knowledge of the number of syllables they contain when spelling them.

6.2.5 Decide how to deal with the ethical issues

Prior to the users being involved in any form of testing or evaluating users were all given informed consent forms to be taken home to their parents or guardians.

The forms explained the purpose of the evaluations and testing as well as the nature of the tasks that the pupils would be involved with. It also made clear that all information would remain anonymous and offered the parents the chance to request a copy of the data.

In addition to these forms all participants themselves were briefed on the purposes of the tests by the SEN teachers and were given the opportunity to withdraw from the process at any time.

A further ethical decision taken was that due to the year 11s impending GCSEs, this year group would not be involved in any testing as this could disturb their revision time.

6.2.6 Evaluate, interpret and present the data

Details of this process are detailed in the remaining sections of this chapter.

6.3 Usability Evaluation

6.3.1 Pupil Observations

As discussed during the DECIDE framework these tests were done using a combination of ‘quick and dirty’ observation techniques combined with small scale usability testing and interviews.

These tests involved asking the pupils to do a 5 - 10 minute free exploration of the tool. This time should give them the opportunity to play the games and experience the reward scheme at least once.

The aim of these sessions was to observe the pupils using the tool in its natural environment without any prompts. This would enable any usability issues to be uncovered as well as allowing the users to familiarise themselves with the system.

The key observations from these sessions were as follows. Where applicable future solutions have been proposed.

Logging In

- When the pupils first sat down to play the game many of them tried to log in to the system straight away rather than creating a new account. They expected the log in to be the same as their school network log in details.
Proposed solution: This functionality is a possibility in the future if such a system was to be implemented in the school. It would reduce the number of user names and passwords that the users need to remember.
- When creating a new account one or two users were not familiar with the password reminder facility but quickly understood the concept when it was explained to them.
Proposed solution: This functionality could become optional so that users who are not familiar with the concept would not become confused.
- When asked to choose their car, the buttons for scrolling through the cars were still not immediately clear to all users although all pupils noticed them eventually without prompting.
Proposed solution: The buttons should be redesigned to make them clearer. The graphics could be given a 3 dimensional appearance as this is common with buttons.

- When reading the instructions prior to the game all users read the instructions well but on some occasions tried to hover over the buttons that have been used as graphics (to explain the buttons function). However, although many users initially hovered over them, nobody tried to click them as they soon realised that they were not buttons in this context.

Proposed solution: As this did not seem to cause users any concerns it could be left as it is. However, the buttons could be placed within a box to make it clear that they are a graphic rather than working buttons.

The Games

- When starting a game, pupils seemed to take into account the instructions for them and did not require prompting on how to control the car or the points system. However as the game controls differed in each game some pupils were initially confused as to why the car in Spelling Grand Prix did not move automatically as it does in Bumper Syllables.

Proposed solution: The car in Spelling Grand Prix could be programmed to begin moving automatically. It would still however retain the ability to stop as this is an important aspect of the game.

- When observing one pupil playing Spelling Grand Prix the pupil encountered a word that they had spelt incorrectly in the initial test. When they hit the spelling that they believed to be correct they realised the error that they had made and noted this to themselves (they later got the spelling correct in the second test).

Reward Schemes

- All users understood the drag and drop nature of the garage reward scheme immediately and appeared to enjoy modifying the cars.
- In regards to the scores section and leader board, users were not fully aware of this until asked to view them in the final section of the testing.

Proposed solution: The presence of a leader board should be made clearer in the instructions page. It could perhaps also be mentioned in the text box of the main games screen.

6.3.2 Structured Tasks

This section of the evaluation was intended to cover common tasks in the game that had not been covered during the free exploration. The aim was to observe the ease with which the users could carry such tasks out.

Due to the nature of the game there were not many tasks that needed to be included here as the core functionality of the system is within the games and reward schemes.

The chosen tasks were as follows:

- Visit the score boards and view your score as well as the leader board for each game.
- Change the settings of the game so that the text and background are different.
- View the help section of the game.

The findings from these sessions were as follows:

Viewing the scores:

Users did not have any hesitation carrying out this task and were excited to realise that they could view the leader board and their scores. However, the fact that the users were not aware of this functionality before means that this should be made more obvious in the future.

Changing the display settings:

Users again were able to access this area and seemed to be familiar with the use of a preview pane. Not all users were sure of the purpose of the section as a whole but those with Dyslexia were happy to see that they could change the settings to meet their needs. One user instantly changed the background colour to blue and said that they did this because they used to have blue tinted glasses to help them read and knew that it was a clearer colour for them.

Viewing the help page:

All users were able to do this efficiently.

6.3.3 User Interviews

During the evaluation sessions the pupils were asked a series of questions in the form of semi structured interviews to elicit their opinions on the experience and pleasure afforded by the game.

In general the overall opinion of the system was that it was fun to use. The pupils were keen to voice their enthusiasm for the games and spoke positively of it. When asked if they would like to play the game again, all users said they would.

When asked if they enjoyed the reward scheme all pupils said that they did. Some commented that it would be better to have more modifications that you could make to the car and even that these options could be unlocked as you progress through the game.

Pupils were also asked what they thought of the games. All users asked said that they enjoyed playing the games, although one female user expressed a desire for some more “girly” games. When probed about this they could not think of a specific example but stated that they enjoyed playing games with a bit of creativity involved. However, this user was still happy to play the existing games and due to the creative nature of the reward scheme, greatly enjoyed this section.

When playing the games the users were observed using the leader board. When talking to users they seemed very keen on this area of functionality and often raised their concentration levels in the game to try and get onto the board. They stated that the presence of the boards motivated them to perform better in the games (as the pupils must get 100% to be added to the board).

6.3.4 Teacher Observations

As it was not possible to observe the pupils at every stage during the week, the teachers were asked to feedback their observations and opinions of the sessions at the end of the week. The following comments were made.

It was stated that, when playing the games, the pupils got on with the task well and really seemed to concentrate on it. During Bumper Syllables the pupils could be seen mouthing the words to themselves as they thought about how many syllables the words had.

When the pupils were told that they would be playing on the system again, the pupils were apparently keen to play the games and enthusiastically began the task.

Pupils were excited about the prospect of getting to be on the leader board and made an effort to do so. They were excited to see when they had appeared on the board.

It was observed that during testing of the game there were some problems with the words loading slowly for the Bumper Syllables game. The reasoning behind this was the increased load on the database caused the processing to be slowed down. This is a problem that would need to be fixed eventually as some pupils with attention problems found this frustrating. This could be achieved by placing a loading status bar while the words load so that users are aware of what is happening and can estimate how long they will need to wait.

The Games

In addition to these comments on the pupils' reactions, the teachers also stated their opinions on the games themselves, having seen them in practice over the week. These were as follows:

Bumper Syllables

It was stated that this game was good as it "required the pupils to visually discriminate the different syllables. They didn't have time to clap out the syllables (as they usually would) and therefore had to rely on their visual and auditory perception alone."

The speed of the game was also pointed out as being a key factor as it gave the pupils a challenge.

It was mentioned that having one syllable words in the boxes was good "as the pupils are often better at multi-syllabic words and recognising some mono-syllables e.g. 'thought' is actually quite challenging as it has seven letters."

Spelling Grand Prix

The game was noted by one of the SENCOs as being "much more effective in terms of improvement than I had anticipated it could be". It was also highlighted as being very effective for high-frequency irregular words due to the fact that it combines auditory and visual modalities.

In addition to this it was mentioned how well the students responded to the game. Several pupils were heard to have immediately commented on mistakes that they realised they had made in the initial spelling test.

One aspect highlighted as working particularly well was the fact that the game allows pupils to slow down and stop the car to consider alternative spellings before making their choice of the correct one. It was pointed out that the nature of the game meant that the pupils had to choose an option, whereas in their free writing the pupils are able to put anything. This then meant that the pupils' success was related to being able to go faster when they could recognise a word without having to slow down so much.

The instant feedback feature was highlighted as being particularly effective, as the pupils are seeing the correct spelling at the time of the mistake rather than at the end of the game. It was stated that SEN pupils often haven't got the patience or motivation to look at their mistakes after the event.

The game was also commended for its “effective and clear format”. It was mentioned that similar ideas had been seen before using mazes, but that these got complicated when mistakes were made which could lead to confusion and frustration. Spelling Grand Prix was said to be effective as the choice of spelling was limited to two or three and that subsequent progress in the game was not dependent on previous choices.

6.4 Learning Content Evaluation

During this stage of the evaluation the learning content was tested. Details of this test plan can be found in sections 6.2.3 and 6.2.4. A breakdown on the results from this section can be found in the Evaluation Appendix.

6.4.1 Test 1

Testing at the beginning of the week yielded a variety of results amongst the participants, demonstrating the varying ability levels of Special Needs pupils. The results from these tests can be seen in Table 6.1.

Table 6.1 Test 1 Results

Experimental Group		Control Group	
Pupil	Result	Pupil	Result
P1	9	P1	9
P2	6	P2	14
P3	6	P3	15
P4	6	P4	13
P5	11	P5	12
P6	14	P6	4
P7	7	P7	12
P8	13	P8	10
Total	72	Total	89
Mean	9	Mean	11.13

Many children struggled with the spelling section of the tasks, often being unable to complete any of these correctly. However, with regards to the syllable related questions the performance here was much stronger. The reasoning behind this was suggested by the pupils’ teacher as possibly being down to the fact that this has been a current area of teaching in the lessons and is a concept that once understood can be applied by pupils quite effectively.

6.4.2 Test 2

Prior to testing, the SENCO stated that, due to the nature of the pupils difficulties, if any progress was to be seen in the second round of testing it may be negligible. This relates to the fact that dyslexic pupils require a large amount of overlearning in order to develop - a task that would not be possible during such a short testing period, but may be the result of repeated exposure over a longer time frame, indicating the need for a longitudinal study.

However, despite this, the results of the second round of testing appeared to be very positive, with all pupils improving or in one case remaining at their previous level. In some cases pupils made as much as a 25% improvement on their first test.

The results of these tests can be seen in Table 6.2. In addition to this, a detailed breakdown of the experimental groups' results can be found in the Evaluation Appendix.

Table 6.2 Test 2 Results

Experimental Group		Control Group	
Pupil	Result	Pupil	Result
P1	9	P1	8
P2	8	P2	14
P3	10	P3	15
P4	8	P4	14
P5	14	P5	12
P6	15	P6	4
P7	8	P7	12
P8	15	P8	11
Total	87	Total	90
Mean	10.88	Mean	11.25

It was noted by the SENCO that even when pupils had not improved to the extent of getting the spelling 100% correct they had improved their attempts considerably. For example one severely dyslexic pupil when first asked to spell 'first' had spelt it 'free' (which did not result from mishearing the word), whilst in the second test their attempt was 'fist'. Another student when asked to spell 'sentence', initially spelt it 'seturs'. By the second test their attempt had improved considerably to 'sentenc'.

With the syllable section of the test, as there was a high level of understanding in the initial tests, this area did not show such high levels of improvement. However, improvements were still seen in this area with some pupils who got one or two of these questions wrong now getting 100% in this section of the task. One pupil in the counterbalanced group was specifically mentioned by the SENCO. This particular student was new to Special Needs lessons and when they first attempted to play the game did not understand the concept of syllables. However, by the end of the testing they were able to answer all these questions correctly.

6.4.3 Summary

Having looked at the results of both the control group and the counterbalanced group, it can be seen that the experimental group's results have improved between the initial testing and the final test. The mean change in the experimental group was 1.875 while the mean change in the control group was 0.125. This shows that there is a higher increase in the group using the system than the group receiving normal testing. See Table 6.3 for a summary of the test comparisons.

Table 6.3 Test 1 and Test 2 Results Comparison

Experimental Group				Control Group			
Pupil	Test 1	Test 2	Change	Pupil	Test 1	Test 2	Change
P1	9	9	0	P1	9	8	-1
P2	6	8	2	P2	14	14	0
P3	6	10	4	P3	15	15	0
P4	6	8	2	P4	13	14	1
P5	11	14	3	P5	12	12	0
P6	14	15	1	P6	4	4	0
P7	7	8	1	P7	12	12	0
P8	13	15	2	P8	10	11	1
Total	72	87	15	Total	89	90	1
Mean	9	10.88	1.88	Mean	11.13	11.25	0.13

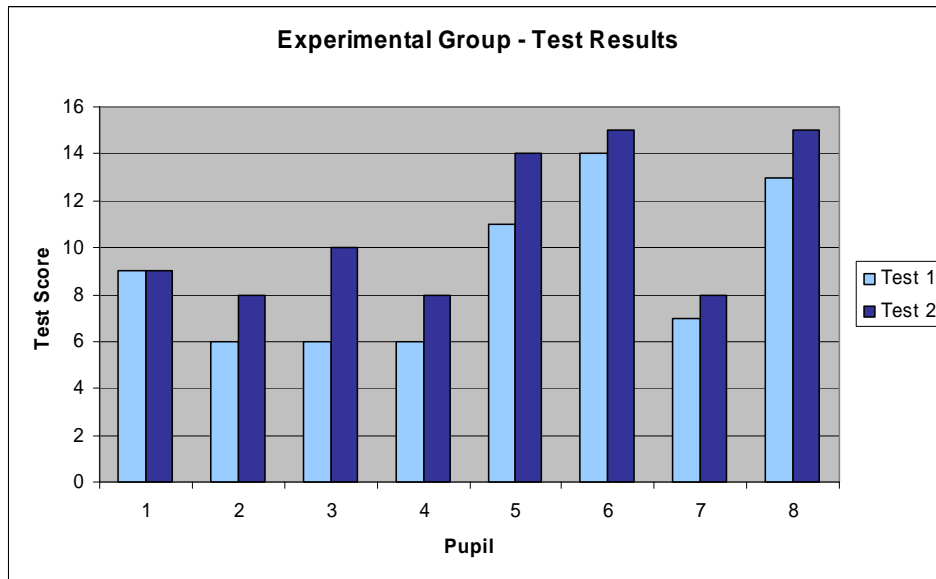


Figure 6.1 Experimental Group – Results Comparison

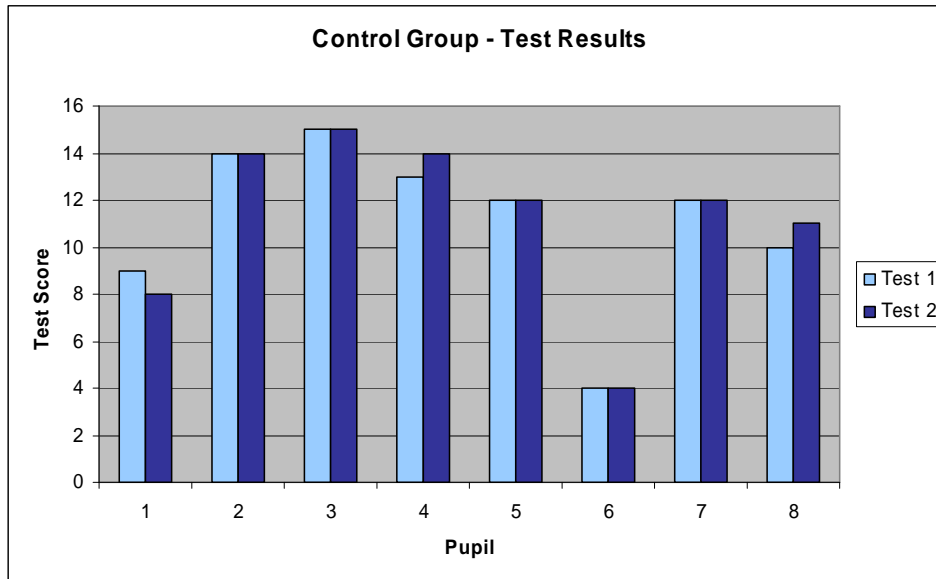


Figure 6.2 Control Group – Results Comparison

In order to check the validity of these results the counter balanced groups' results have been analysed. If taking the test itself creates a practice effect this will be highlighted by the results of this group. This can be established by comparing the mean difference between the two sets of counterbalanced results with the mean improvement of the experimental group. If taking the test gives no improvement on its own, these means should be similar in value.

Table 6.4 Counterbalanced Group - Results

Counterbalanced Group				
Test - Game	Test	Game - Test	Test	Difference
P1	12	P1	15	3
P2	9	P2	11	2
P3	7	P3	7	0
P4	10	P4	12	2
Total	38	Total	45	7
Mean	9.5	Mean	11.25	1.75

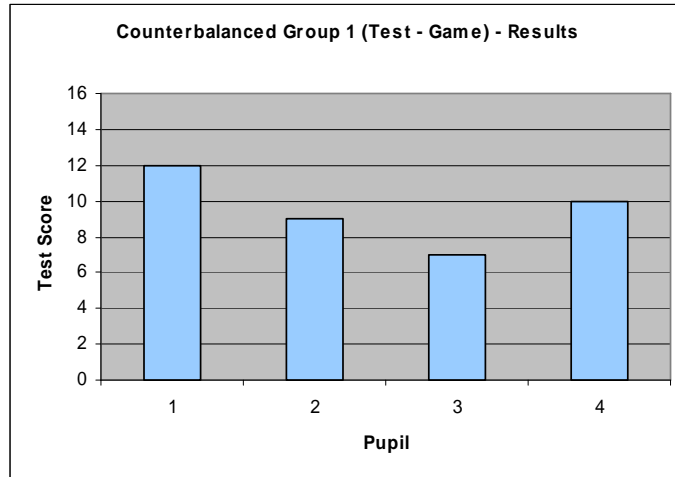


Figure 6.3 Counterbalanced Group – Test the Game – Results

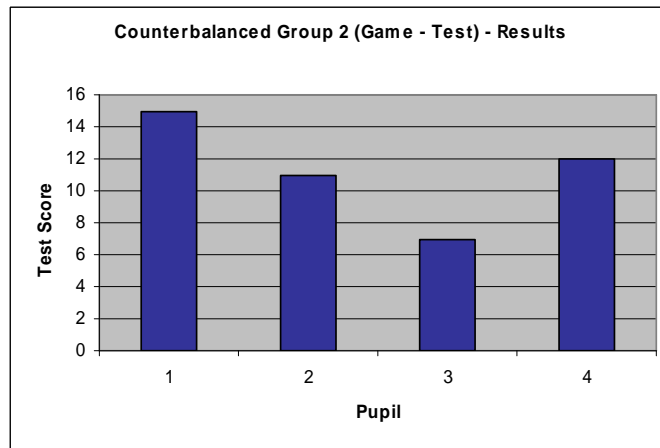


Figure 6.4 Counterbalanced Group – Game then Test – Results

The mean difference between the counterbalanced groups' results was 1.75. As stated previously the mean improvement of the experimental group was 1.875. This is a difference of 0.125, which is exactly the improvement seen in the control groups' results. This indicates a very slight practice effect, but still implies that the majority of the improvement in the experimental group is from using the system.

Whilst these results are promising it is important to recognise that, given the appropriate time, it would be necessary to test the programme over a prolonged period of time to fully assess the success of the system. In these circumstances it would be preferable to test the system at the beginning of the week and again after a longer period of use such as 2 weeks (where the pupils would have a total of 5 lessons to play the game). On top of this testing it would be interesting to also test the pupils after a period of time not using the system to test the extent to which the knowledge gained is retained by the pupils.

As such testing has not been possible during the course of the project, it is necessary to use the results gathered so far as a preliminary indication of the potential of the game and not a confirmation of its success.

6.5 Conclusion

As stated in section 6.2.1 (within the DECIDE framework), the aim of the evaluation process was to determine the usability of the system, whether the users enjoy using it, whether they are motivated by it and whether the learning content of the system is effective in improving the pupils literacy skills. These goals were then split down into a series of questions which were used during the evaluation design to ensure that these objectives were met.

The results from these evaluations can subsequently be summarised as follows:

Is the system easy to use (is the system usable)?

From the usability testing and observations it can be seen that there were no major usability problems uncovered by the users and all users could use the system effectively. However, minor problems did arise that would need to be solved before further testing.

Do the users enjoy using the system?

From observations of the users when using the system, high levels of enthusiasm could be seen. In addition to this, feedback from the pupils confirmed that they enjoyed the system and would like to use it again in the future.

Does the system motivate the users?

During the design process, the primary motivation within the game was designed to be the car modification reward scheme. The aim of this functionality was to get users to complete the games. During evaluations this could be seen to motivate the pupils to play the games and there was much enthusiasm towards the garage section where users enjoyed modifying their cars.

However, the leader boards, which were initially added due to repeated requests from the users, became a key motivation within the game. The motivation from this appears to be highly valuable as the users set out in the games to get on the leader board and therefore appeared to put in much higher levels of concentration in order to get maximum points in the games. Subsequently this area could be considered to be one of the greater successes of the system and highlights the importance of a user centred design process.

Is the learning content in the game effective?

The preliminary results from the learning evaluation, as discussed in section 6.4, are that when compared against traditional teaching methods, the game has caused an increase in test performance.

These improvements, although not large, have been much greater than anticipated by the SEN teachers and are therefore very promising. Indications from the results and teacher feedback indicate that Spelling Grand Prix was the most instrumental game in these results.

However, as mentioned previously these results are preliminary and would need to be backed up by more in depth and prolonged testing.

7 Conclusion

When beginning this project the aim was to produce a system designed specifically for users with reading difficulties between the ages of 11 and 16. The reasoning behind this is that many systems in this domain are either targeted at a very broad age group or at young children. The need was subsequently seen for a system that could specifically deal with the issues of a reading difficulty in an environment that would stimulate and motivate older pupils who may find existing systems patronising (high interest/low level software). Later empirical evidence supported this need, as many potential users, when asked about existing software, were frustrated by the immature nature of these systems and the way that this made them feel.

Early findings suggested that a potential problem with such a system may be the fact that intervention schemes with older children can not be as effective as with younger children (Sylva & Hurry, 1995). However, subsequent findings from the literature review suggest that although this may be true, computer assisted learning in secondary schools can be both effective and cost effective (Lynch et al., 2000). It was therefore hoped that this project would be able to find similar findings with regards to the effectiveness of the learning content of the final product.

Through further research into the domain during the Literature Review, it was possible to see the specific benefits that can be gained through using computer assisted learning, not just in teaching reading but more specifically in teaching children with reading difficulties.

Firstly, a computer can provide multi-sensory output, providing both visual and auditory representations of words and therefore providing as many paths to learning as possible. (BECTA Overview of SEN and ICT Provision, 2002).

In addition to this, a computer provides an ideal environment for the use of overlearning. Overlearning is a key technique for dealing with reading difficulties concerned with repeated exposure to words. This was highlighted by Nicholson and Fawcett (2000, cited by Fawcett, 2003) who stated that for performance to become automatic with Dyslexic children it can take longer by a factor of the square root of the time it would normally take to acquire the skill. However, such a large amount of repeated exposure to a set of words can become tedious for pupils if carried out in a stationary environment. A computer can benefit this process by providing variety to this learning in a visually stimulating and dynamic environment.

A further benefit that computers provide is the ability for users to be able to adapt the screens appearance to suit their specific difficulties (see section 2.7.2). This is key to Dyslexic users who have specific preferences for the way that text is presented (including font and colour).

An additional issue to be addressed within the literature review, prior to designing the system, was to look at ways in which a system should be presented in order to make the system as accessible to Dyslexic users as possible. This was achieved by undertaking a detailed look into current guidelines and literature in the area which was then presented in the literature review as a Dyslexia Style Guide (see section 2.7.2). This was a crucial output from this stage of the project as it provided a clear set of guidelines to follow during the design. However, to make the guidelines as succinct and compact as possible (for constant referencing), these guidelines have been combined into a one page check list. This can be

found in the Conclusion Appendix section 1. This list provides a quick simple reference point that all design decisions can be checked against.

These findings from the Literature Review were key to the development of the system as they were able to highlight areas of potential learning content for the system as well as the areas in which computer assisted learning can best benefit users with reading difficulties. It also enabled the production of a set of guidelines to ensure that the system design meets the variety of needs that Dyslexic users may have.

However, despite these considerations taken, to produce a system that simply addresses the findings from the literature review is not enough when developing a system for such a varied range of users. Further understanding of not just their needs, but what motivates and interests the users is required as well as an understanding of the environment that the tool will be used in.

Subsequently during the requirements and design process, users and wider stakeholders were involved as much as possible. However, working with SEN pupils presented difficulties at times due to the sometimes unpredictable behaviour of the pupils (as many SEN pupils often suffer from behavioural difficulties and reduced motivation levels).

In early sessions with the pupils it was sometimes difficult to gain high levels of feedback from the users as they often appeared shy or disinterested. To overcome this, concentration was then placed on building a rapport with the pupils to give them more confidence to express opinions and thus the ability to provide a greater level of feedback.

In addition to this, steps were taken during the iterative design process to make the prototype evaluation sessions as stimulating and motivating to the users as possible (to gain their full attention). This was required due to the attention issues that the users often had, as many Dyslexic users have fluctuating levels of concentration (Goodwin and Thomson, 2004). As a result of this, all evaluations took place using computer based prototypes. This provided a more visually stimulating environment for the users and subsequently higher motivation and enthusiasm to take part in the sessions.

These user interactions highlight the importance of adapting the user-centred design process to meet the needs of the specific users taking part in these. Building relationships and providing a stimulating environment to the users proved key in gathering a higher quality of data within this project.

The requirements gathering process also enabled important interactions to take place with SEN teachers. These interactions became vital to the project, as although the literature review provided insight into teaching practices, the reality of these within the SEN classroom can be vastly different. It also enabled specified teaching practises to be observed that provided vital information for the subsequent design process.

Having taken such an in depth look at the requirements and design of the system and having taken into account the users wants and needs, the implementation of the system became reasonably complex. This level of complexity was deemed necessary for the prototype, as much of the complexity came from the reward schemes and motivational aspects of the game. These are important features in the success of a game and therefore could not be overlooked.

This, therefore, led to a much larger than anticipated programming overhead. However, having completed the implementation, the benefits could clearly be seen during the system evaluation sessions.

The aim of the evaluation sessions was to assess the enjoyment value of the system, its usability levels and the educational benefits that it can provide. This was evaluated through a series of usability tests, observations, interviews and learning content tests. The findings from these sessions provide a promising indication of the success of the system as users both enjoyed and felt motivated by the system as well as subsequently improving their performance in a literacy test.

These findings consequently appear to support the findings of Lynch et al (2000), as discussed at the beginning of this chapter, that computer assisted intervention schemes in secondary schools can be effective.

However, as stated within the Evaluation chapter these findings are a preliminary indication of success and will need to be backed-up by a more in depth evaluation.

Summary

Having completed this project, the outputs from the various processes undertaken can be summarised as follows:

- Findings from the literature review have shown the key ways in which children with reading difficulties should be taught, with a focus on the benefits that computer based presentation can provide to this. It has also been possible to look into the best ways to present a system for users with Dyslexia by considering the range of difficulties faced by such users.
- Through working closely with SEN pupils suffering from a range of learning difficulties, it has been necessary to adapt the methods used during these interactions to get the most out of the pupils. This allowed an optimum environment to be provided in which qualitative data could be gathered. The benefits of working closely with a wider range of stakeholders have also been shown to be highly beneficial.
- Empirical evidence has been found that appears to support the findings by Lynch et al (2000) that computer based intervention methods in secondary schools can be effective. It has also demonstrated the benefit of computer based instruction compared to normal teaching methods. However, these findings will need to be confirmed through further evaluations.
- Finally, a system has been produced that has the potential to be developed further and used within SEN classrooms to both entertain and motivate a range of users and through this, potentially develop their literacy skills.

7.1 The Future

Having carried out this project it is natural progression to look to the future and consider the potential directions that the project could take, as well as discussing improvements to the work.

One key issue to be addressed in the future would be to create solutions to the usability issues that arose in the final evaluation sessions and then test the system over a prolonged

period. This is important as it would allow the effect of the system to be more accurately measured. The current testing, although promising, can not indicate whether the learning gains observed will remain over time which is a common problem with Dyslexic individuals and this subsequently is something which should be tested for a more accurate indication of the systems success.

A further development in this area would be to look at the current adaptability in the system and further develop this for prolonged use. The system currently uses a system of repeated exposure to words where a pupil will continue to be exposed to a word until they have got it correct 3 times. If the system were to be used over a greater length of time it would be preferable to design the system so that these words would continue to appear in the game at regular intervals (although not as often) until the child appears to have transferred the word into their long term memory.

After evaluation sessions, the SENCO from the school where the sessions were carried out expressed a high interest in the use of the system if it was to be further developed. Particular interest was shown in Spelling Grand Prix which could even be adapted into a game in its own right. The SENCO felt that there was “a lot of scope to develop the game further but keeping the essentially simple format for choosing the words”. Suggestions for its development from the SENCO included development of the graphics around the track and the addition of animations such as champagne corks popping if the pupils complete the game with maximum points (as you would get in a real Grand Prix). It was also suggested that the pupils could compete directly with each other to provide further motivation.

Working closely with SEN specialists has been a key factor in the success of the system especially in regards to the learning content of the games. In future development it would be highly beneficial to develop this relationship further and participative design should be an important consideration in this area.

A key development of the whole system if it was to be developed fully would be to increase the number of games within the system. Currently there are two games, each testing a different skill. This could be increased to a suite of games that develop a range of skills. One of these should deal with the area of synthetic phonics which has been discussed in depth within the literature review. The prototype implemented during the course of this project did not use this skill as it conflicted with the current teaching methods of the school at present. However, when this technique is fully rolled out in secondary schools it will be an ideal skill to be incorporated into a game.

Much of the pupil feedback gathered during system evaluations showed that pupils were motivated by and enjoyed the reward scheme implemented. However for extended use this would need to be developed further so that it can provide prolonged motivation to the pupils. Suggestions from the pupils on this topic included the idea of gradually unlocking different modifications as you progress through the game. For example, the game could begin with allowing the users to change the colour of their car. When the users progressed to the next level they could then be allowed to add parts such as hubcaps to their car. This would progress until they had unlocked all available options. As well as this it would be necessary to add more car additions to the game to add more variety to the reward scheme.

A key area of functionality that was not implemented for the purposes of the prototype is the teacher’s management system. The need for such functionality was a key finding of both the literature review and requirements analysis yet the justification for this omission was that the game needed to be tested with the primary users to confirm its potential success before development in this area should begin. Now that this evaluation has been completed and the

results appear promising, this area should be considered in further design iterations. Ideas for this section include the functionality to view a users progress in the games, with key problem words highlighted (i.e. when a pupil has got a word incorrect more than 3 times). The teachers should also be given the chance to set a user's default level in the game. A further possibility includes giving teachers the ability to alter the number of times a user is exposed to the words in the system (depending on the severity of their reading difficulty).

These settings would cater for the wide range of abilities of the pupils. More advanced users would be able to avoid having to go through sets of 'easy' words before they reach a level that sufficiently challenges them, whilst more severely dyslexic users could be provided with greater amounts of exposure to words.

Due to technological developments in classrooms, the use of interactive whiteboards is becoming increasingly common. The school that was visited throughout this project had an interactive whiteboard in both of the SEN classrooms. The development of the game for use on these was out of the scope of this project; however it would be interesting to pursue this further. The current nature of the games is perhaps not appropriate for such use, due to their highly interactive nature and the complex game controls. However, within an increased game suite there is potential for games to be created that are more appropriate for this use.

The scope of the system could also be widened to provide a tool for the screening and diagnosis of pupils with expected reading difficulties. The tool would provide a fun and enjoyable environment for these assessments as well as allowing the pupils responses to be recorded in detail. These results could then be presented to specialist to allow a diagnosis to be made. For this functionality the tool would need to be developed significantly to allow for a much more detailed level of analysis to be made on the users' performance.

These suggestions are all important considerations to be taken if the system was to be developed further. The most important of these initially would be to further evaluate the system over a prolonged period. After this, it would be beneficial to add a teacher management section to the system as the findings of the Literature Review highlight this as an area that could further improve the effectiveness of the system. With these improvements in place, a key area of development would be the revision of the learning content adaptation. The current functionality was sufficient for the purposes of the prototype, but there is much potential in this area that should be considered.

With the existence of so many future considerations, it can be seen that this is an area with much potential. This demonstrates the need for such systems to be further investigated and developed.

8 Bibliography

8.1 Books

- Cowling, K., Cowling, H., (1993). *TOE BY TOE®. A Highly Structured Multi-Sensory Reading Manual for Teachers & Parents*. K & H Cowling: Basildon, West Yorkshire
- Dix, A., Finlay, J., Abowd, G., Beale, R., (2004). *Human-Computer Interaction*. 3rd Ed. Pearson Education: Essex
- Fawcett, A., (2002). Dyslexia and Literacy: Key Issues for Research. In: Reid, G., Wearmouth, J., eds. *Dyslexia and Literacy: Theory and Practice*. John Wiley & Sons: Chichester, pp.11-28.
- Goodwin, V., Thomson, B., (2004). *Making Dyslexia Work for You: A self-help guide*. David Fulton Publishers: London
- Goswami, U., Bryant, P., (1990). *Phonological Skills and Learning to Read*. Lawrence Erlbaum Associates Ltd, Publishers: East Sussex.
- Griffiths, M., (2002). *Study Skills and Dyslexia in the Secondary School*. David Fulton Publishers: London.
- Harrison, C., (2004). *Understanding Reading Development*. SAGE Publications: London.
- Knight, D., Hynd, G., (2002). The Neurobiology of Dyslexia. In: Reid, G., Wearmouth, J., eds. *Dyslexia and Literacy: Theory and Practice*. John Wiley & Sons: Chichester, pp.29-43.
- Merrill, P., Hammons, K., Vincent, B., Reynolds, P., Christensen, L., Tolman, M., (1996). *Computers in Education*. 3rd Ed. Allyn & Bacon: London.
- Preece, J., Rogers, Y., Sharp, H., (2002). *Interaction Design*. John Wiley & Sons: New York.
- Sylva, K., Hurry, J., (1995). *Early intervention in children with reading difficulties: an evaluation of reading recovery and phonological training*. School Curriculum and Assessment Authority: London.
- Woolfolk, A., (2004). *Educational Psychology: International Edition*. 9th Ed. Pearson Education: Boston.

8.2 Journals

- Dickinson, A., Eisma, R., and Gregor, P. (2003). Challenging interfaces/redesigning users. In *Proceedings of the 2003 Conference on Universal Usability* (Vancouver, British Columbia, Canada, November 10-11, 2003). CUU '03. ACM Press, New York, NY, 61-68.

- Lynch, L., Fawcett, A., Nicholson. (2000). Computer-assisted reading intervention in a secondary school: an evaluation study. *British Journal of Educational Technology*, 31(4), pp. 333-348.
- Underwood, J. (2002). A comparison of two types of computer support for reading development. *Journal of Research in Reading*, 23(2), pp 136-148.

8.3 Reports

- BECTA (2005). *The Becta Review 2005*. Coventry, (02/DD04-05/1011/089/PC/5K
- DfES (2004). *ICT in Schools Survey 2004*. London. (DfES/1122/2004)
- Department for Special Education and Skills (2001) *Special Educational Needs Code of Practise*. DfES Publications. Nottinghamshire.
- House of Commons Educational & Skills Committee (2005) *Teaching Children to Read*. The Stationary Office Limited. London.

8.4 Online Resources

- BECTA (2002) *An overview of special educational needs and ICT provision*. [online] Available from: http://www.ictadvice.org.uk/index.php?section=tl&rid=1955&catcode=as_inc_sup_03. [Accessed 30th November 2005].
- Bristol Dyslexia Centre (2006). *Nessy Learning Programme* [online]. Available from: <http://www.dyslexiacentre.co.uk/> [Accessed 28th March 2006].
- Channel 4 (1995). *Dispatches: The Dyslexia Myth*. [online] Available from: http://www.channel4.com/news/microsites/D/dyslexia_myth/ [accessed 17th October 2005]
- Department for Education and Skills (2005). *TeacherNet, ICT Facts* [online]. Available from: <http://www.teachernet.gov.uk/wholeschool/ictis/facts/> [Accessed 4th December 2005]
- Department for Education and Skills (2005). *TeacherNet, SEN Glossary of Terms*. [online]. Available from: <http://www.teachernet.gov.uk/wholeschool/sen/senglossary/> [Accessed 28th November 2005].
- EducationCity Ltd (2006) *Education City*. [online]. Available from <http://www.educationcity.com> [Accessed 28th March 2006]
- Parker, A., Litterick, I. (2005). *Dyslexic.com: Fonts for Dyslexia* [online]. Available from: <http://www.dyslexic.com/database/articles/fonts.html> [Accessed 1st December 2005].
- Rose, J., (2005) *Independent Review of the Teaching of Early Reading: Interim Report*. [online] Available from: <http://www.standards.dfes.gov.uk/rosereview/> [Accessed 4th December 2005]

- The British Dyslexia Association (2005). *Dyslexia Style Guide*. [online] Available from: <http://www.bda-dyslexia.org.uk/main/information/extras/x09frend.asp> [Accessed 1st December 2005]
- The Dyslexia Institute (2006) *Frequently Asked Questions on Dyslexia by Teachers, SENCOs and Parents* [online] Available from: <http://www.dyslexia-inst.org.uk/faqs.htm> [Accessed 6th May 2006]
- Wikipedia (2005). *Overlearning*. [online] Available from: <http://en.wikipedia.org/wiki/Overlearning>. [Accessed 29th November 2005]
- White Space (2006). *Wordshark* [online] Available from: <http://www.wordshark.co.uk> [Accessed 1st May 2006]

8.5 Additional Reading

- Donaldson, M., (1978). *Children's Minds*. Fontana Paperbacks: London.
- Dickinson, A., Gregor, P., and Newell, A. F. 2002. Ongoing investigation of the ways in which some of the problems encountered by some dyslexics can be alleviated using computer techniques. In *Proceedings of the Fifth international ACM Conference on Assistive Technologies* (Edinburgh, Scotland, July 08 - 10, 2002). Assets '02. ACM Press, New York, NY, 97-103.
- Reason, R., Boote, R., (1994). *Helping children with reading and spelling: A Special Needs Manual*. Routledge: London.

Appendices

Contents

1	Literature Review Appendix.....	A2
1.1	<i>Design Heuristics.....</i>	<i>A2</i>
2	Requirements Appendix	A4
2.1	<i>Wordshark.....</i>	<i>A4</i>
2.2	<i>Education City</i>	<i>A10</i>
2.3	<i>Nessy.....</i>	<i>A16</i>
3	Design Appendix	A22
3.1	<i>Prototype 1 - Screen Shots.....</i>	<i>A22</i>
3.2	<i>Prototype 2 – Screen Shots.....</i>	<i>A25</i>
4	Implementation Appendix	A29
4.1	<i>System Overview and Navigation</i>	<i>A29</i>
4.2	<i>Game Objects.....</i>	<i>A32</i>
4.3	<i>Layers</i>	<i>A33</i>
4.4	<i>Database E-R Diagram</i>	<i>A35</i>
4.5	<i>Car Graphics</i>	<i>A36</i>
4.6	<i>Classes</i>	<i>A36</i>
4.7	<i>Random Number Function.....</i>	<i>A48</i>
4.8	<i>High Fidelity Prototype</i>	<i>A49</i>
5	Evaluation Appendix	A61
5.1	<i>Learning Content Test</i>	<i>A61</i>
5.2	<i>Test Results</i>	<i>A62</i>
6	Conclusion Appendix.....	A64
6.1	<i>Condensed Dyslexia Style Guide</i>	<i>A64</i>

1 Literature Review Appendix

1.1 Design Heuristics

Schneiderman's Eight Golden Rules of Interface Design - taken directly from Dix et al (2004).

1. Strive for consistency in action sequences, layout terminology, and command use.
2. Enable frequent users to use shortcuts such as abbreviations, special key sequences and macros, to perform regular, familiar actions more quickly.
3. Offer informative feedback for every user action, at a level appropriate to the magnitude of the action.
4. Design dialogs to yield closure so that the user knows when they have completed a task.
5. Offer error prevention and simple error handling so that users are prevented from making mistakes and, if they do, they are offered clear and informative instructions to enable them to recover.
6. Permit easy reversal of actions in order to relieve anxiety and encourage exploration, since the user knows that he can always return to the previous state.
7. Support internal locus of control so that the user is in control of the system, which responds to his actions.
8. Reduce short-term memory load by keeping displays simple, consolidating multiple page displays and providing time for learning action sequences.

Neilsen's Ten Usability Heuristics – taken directly from Dix et al (2004).

1. Visibility of system status – Always keep the users informed about what is going on through appropriate feedback within reasonable time.
2. Match between system and real world – The system should speak the user's language, with words, phrases and concepts familiar to the user, rather than system-orientated terms. Follow real-world conventions, making information appear in a natural and logical order.
3. User control and freedom – Users often choose system functions by mistake and need a clearly marked 'emergency exit' to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
4. Consistency and standards – Users should not have to wonder whether words, situations or actions mean the same thing in different contexts. Follow platform conventions and accepted standards.
5. Error prevention – Make it difficult to make errors. Even better than good error messages is a careful design that prevents a problem from occurring in the first place.
6. Recognition rather than recall – Make objects, actions and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for the use of the system should be visible or easily retrievable whenever appropriate.
7. Flexibility and efficiency of use – Allow users to tailor frequent actions. Accelerators – unseen by the novice user – may often speed up the interaction for the expert user to such an extent that the system can cater to both inexperienced and experienced users.
8. Aesthetic and minimalist design – Dialogs 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.
9. Help users recognise, diagnose and recover from errors – Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

10. Help and documentation – Few systems can be used with no instructions so it may be necessary to provide help and documentation. Any such information should be easy to search, focussed on the user's task, list concrete steps to be carried out, and not be too large.

2 Requirements Appendix

2.1 Wordshark

Wordshark is a system aimed at pupils with specific learning difficulties. It has been developed by professional computer programmer Roger Burton and practising SEN co-ordinator Ruth Savery (White Space).

The system is designed for use in both primary and secondary schools and contains 41 different games which target skills such as:

- Phonics
- onset and rime
- homophones
- spelling rules
- common letter patterns
- visual and auditory patterns
- prefixes and suffixes
- roots
- word division
- high frequency words
- use of words in context
- alphabet and dictionary skills

(www.wordshark.co.uk)

Extensive word lists are contained within the system and are arranged into the following structured courses:

- The English “Literacy Hour” UK spellings and word recognition
- Secondary school subject vocabulary / Adult learners
- The 3 stages of the book “Alpha to Omega”
- Phonics
- Course for older users and adults

(www.wordshark.co.uk)

For the purposes of the stakeholder evaluations a range of games were played. Below are screen shots and brief explanations of a selection of these games, as well as the reward schemes available within the system.

2.1.1 Games

Below is a selection of game screen shots and short descriptions of these.

Rolling

In this game, the aim is to correctly guess the location of the letter in the alphabet. Letters drop down from above and the users are required to aim the shot so that the ball lands in the correct slot. Marks are awarded for the accuracy of the shot i.e. a close shot = 5 points whilst an exact hit is 10 points.

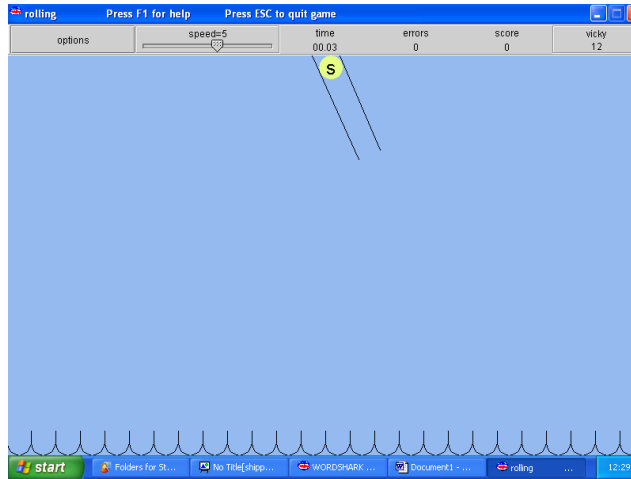


Figure 2.1 'Rolling' - Screenshot 1

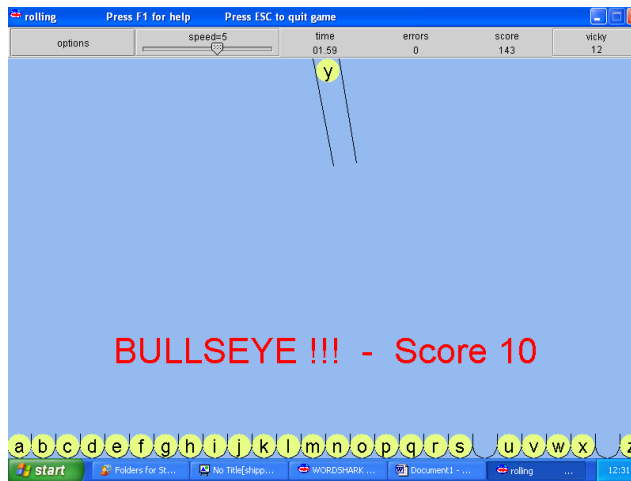


Figure 2.2 'Rolling' - Screenshot 2

Sharks

The aim of this game is for users to click on a shark. They then hear speech output of a word and are required to spell this word. In addition to this sound a picture is provided as an additional clue.

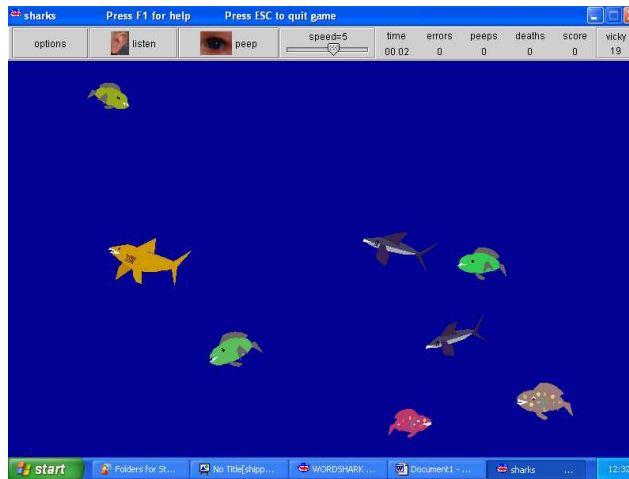


Figure 2.3 'Sharks' Game – Screenshot 1

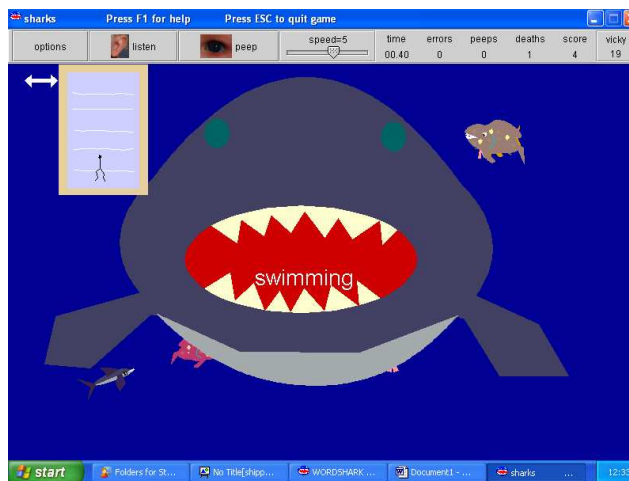


Figure 2.4 'Sharks' Game – Screenshot 2

2.1.2 Reward Games

Below are a series of screenshots and descriptions of the reward games within Wordshark.

Bowls

This reward game involves users aiming the ball to knock down the skittles. Points are awarded for the number of skittles hit.

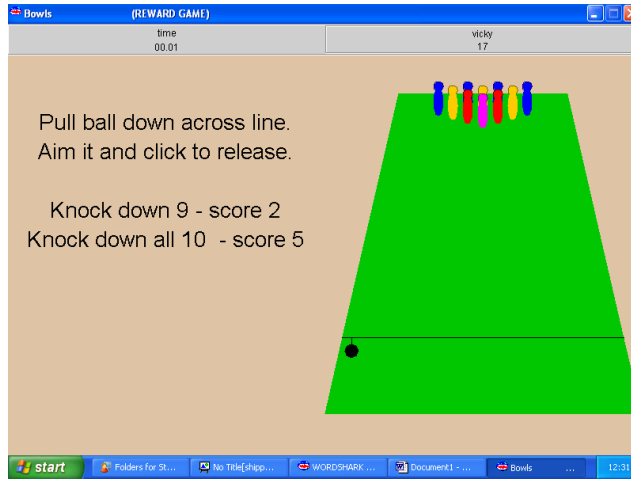


Figure 2.5 'Bowls' Reward Game.

Race

This reward game involves betting on an animal to win the race. Points are awarded for the placing of the selected animal.



Figure 2.6 'Race' Reward Game.

Lottery

This reward game involves selecting 3 numbers. The lottery is then drawn and users are awarded points for the number of matches they make.

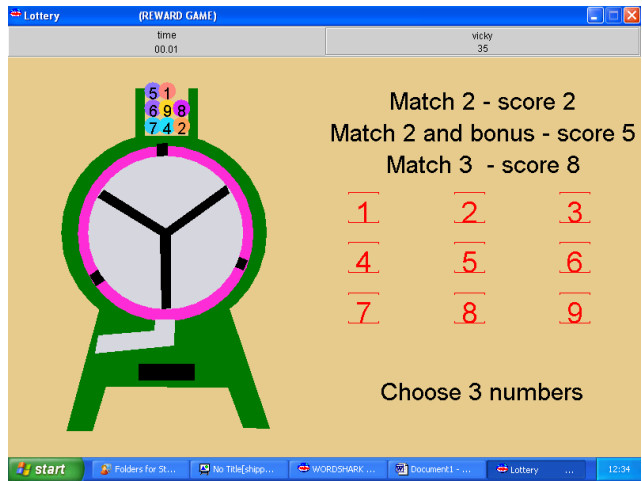


Figure 2.7 'Lottery' Reward Game

2.1.3 Further Screenshots

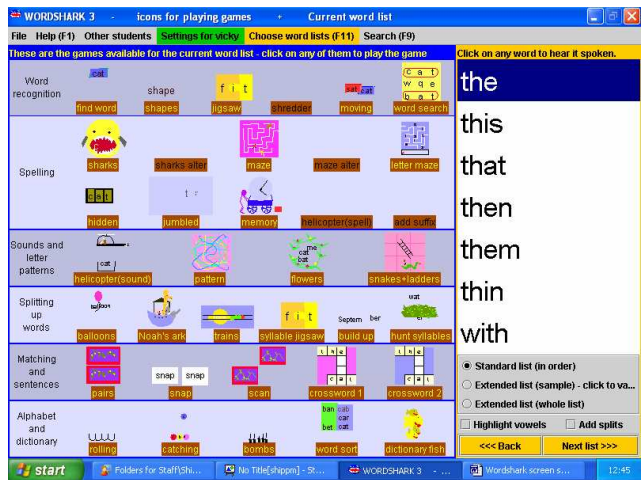


Figure 2.8 Opening Screen

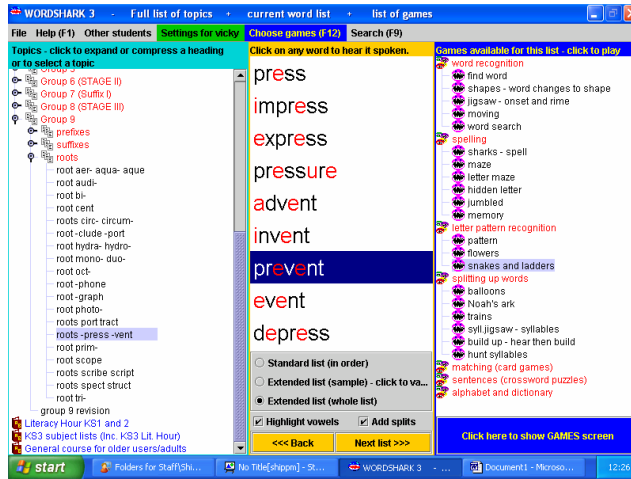


Figure 2.9 Wordlist selection in Wordshark

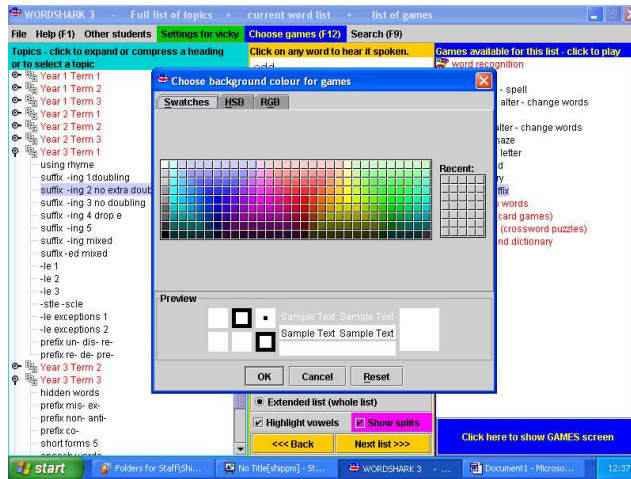


Figure 2.10 Settings Screen

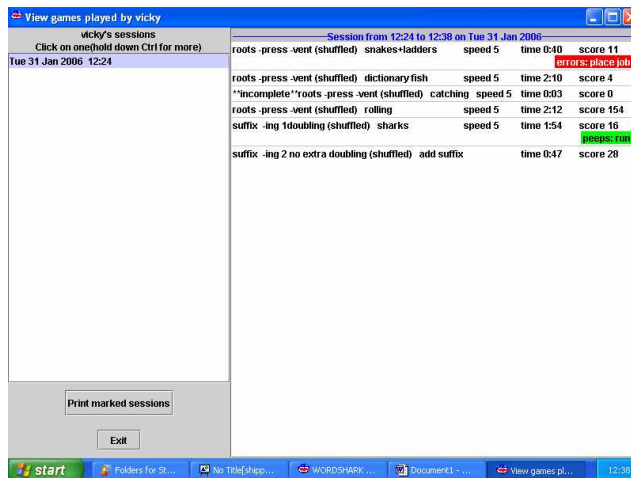


Figure 2.11 Teacher Feedback Screen

2.2 Education City

EducationCity is an e-learning product designed for use in both schools and homes. It aims to reinforce learning in a “fun and engaging way”. (www.educationcity.com).

As well as English, it also covers Maths, Science and French up to the ages of 12. The system is used within SEN lessons as the content provided is appropriate for the pupils’ abilities, even though the targeted age group is generally lower than that of the pupils in the class.

It is designed for use on both computers and interactive whiteboards.

What follows is a selection of screen shots from the system:

2.2.1 Logging In



Figure 2.12 Education City Login Page



Figure 2.13 Education City Game Choice

2.2.2 Sten the Wizard

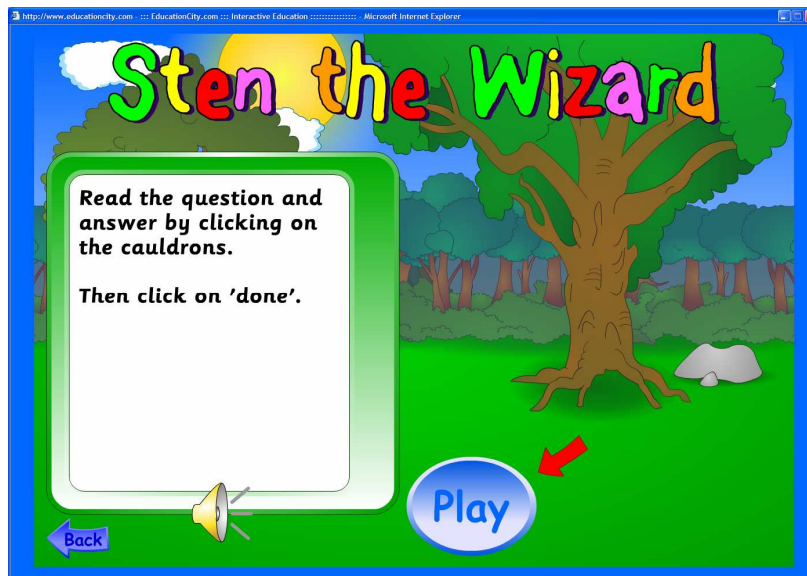


Figure 2.14 Sten the Wizard - Instructions



Figure 2.15 Sten the Wizard – OnScreen Keyboard

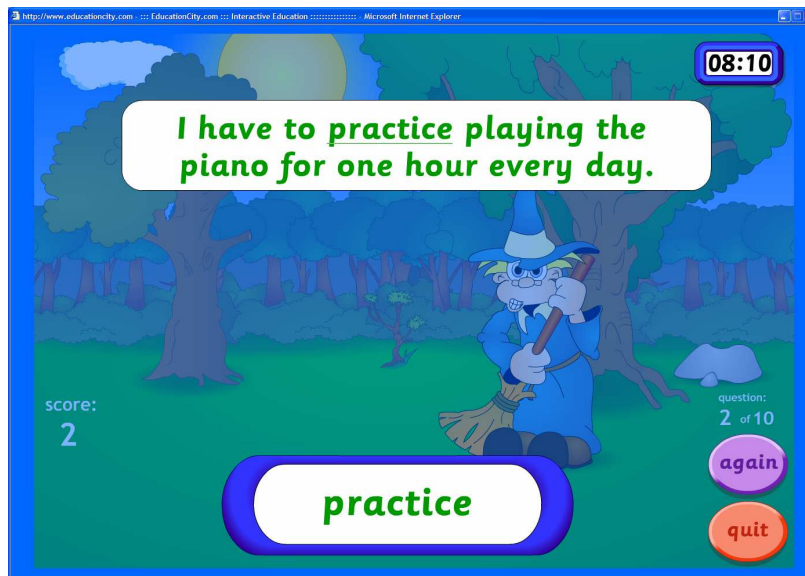


Figure 2.16 Sten the Wizard – Correct Feedback



Figure 2.17 Sten the Wizard – Incorrect Feedback



Figure 2.18 Sten the Wizard – Reward Graphics

2.2.3 Slam Dunk

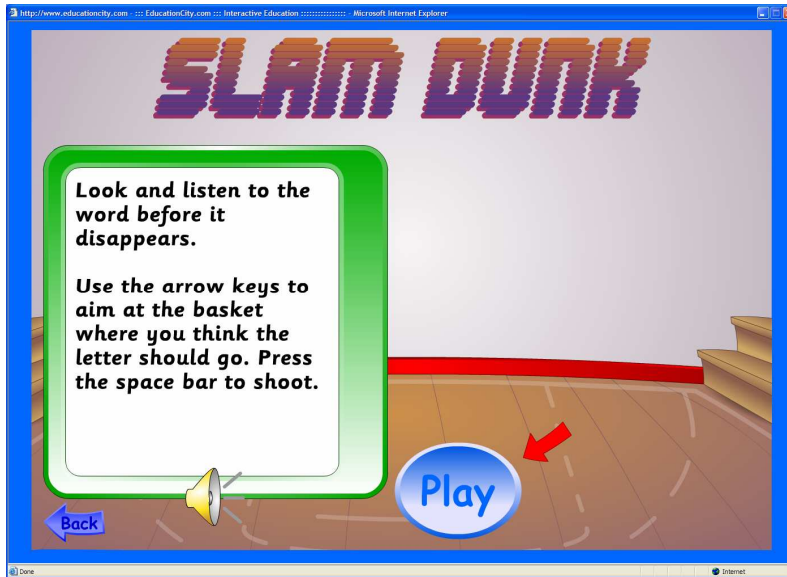


Figure 2.19 Slam Dunk Instructions



Figure 2.20 Slam Dunk – Game Screen



Figure 2.21 'Quit' Confirmation Screen

2.2.4 Feedback Screens



Figure 2.22 Success Tracker

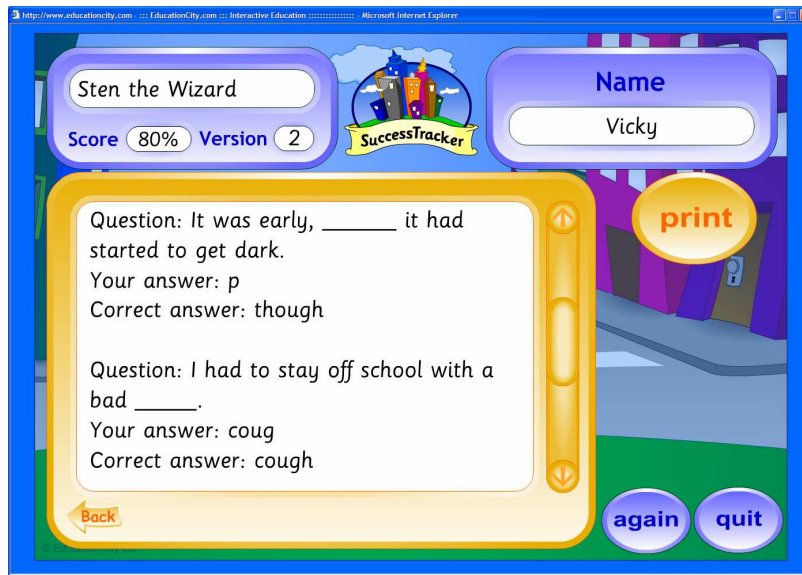


Figure 2.23 Feedback Screen

2.3 Nessy

Nessy is a structured learning system specifically targeted at pupils with Dyslexia between the ages 5 (KS1) and 16 (KS4). The system was developed by Pat Jones, teacher and founder of the Bristol Dyslexia Centre and Belgrave School together with her Dyslexic son Mike Jones.

The system contains a series of animated computer games each targeting a learning skill. These games are as follows:

- Brain Drain- encourages reading speed and vocabulary development
- Chase The Chicken - improves instant word reading recognition
- Poke the Pig - improves instant word reading recognition
- Doggy Din Dins - reinforces the suffixing rules
- Glynn Goes Fishing - reinforces single word spelling
- Squish 'Em -. reinforces single word spelling
- Hands Off My Bananas - reinforces single word and sentence spelling
- House of Fear - Cloze Reading for vocabulary and context
- Monster Sound Sorting - recognition of the different sounds made by 'ough'
- Power Protectors - reinforces single word spelling of the Protector rule
- Wet Kipper (2 Player) - reinforces single word and sentence spelling
- Whack-A-Rat - reinforces single word spelling

(Bristol Dyslexia Centre Website)

The Game itself is centred around a series of colour coded islands each focusing on a different level of ability.

The games used in the Heuristic Evaluation of this tool can be found on the Bristol Dyslexia Centre Website. Screen shots of these games can be seen below:

2.3.1 Whack-a-Rat

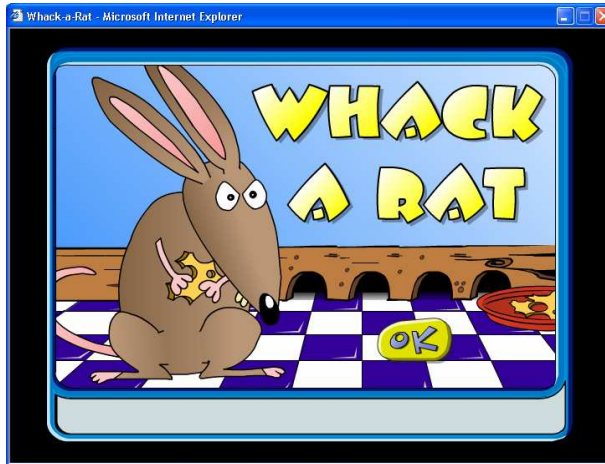


Figure 2.24 'What A Rat' Introduction Screen

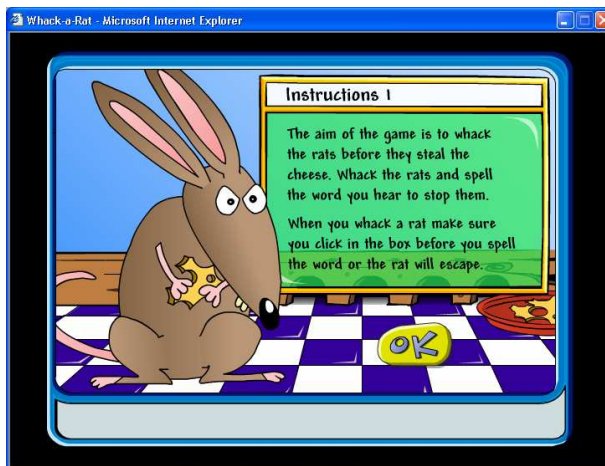


Figure 2.25 'What A Rat' Instructions Screen



Figure 2.26 'What A Rat' Game Screen 1



Figure 2.27 'What A Rat' Game Screen 2

2.3.2 Doggy Din Dins

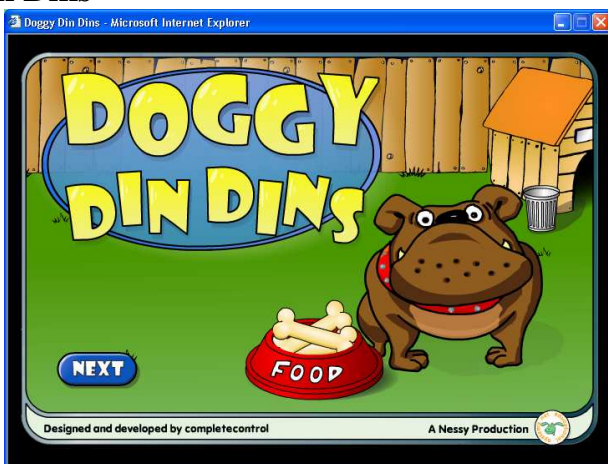


Figure 2.28 'Doggy Din Dins' Introduction Screen



Figure 2.29 'Doggy Din Dins' Instructions Screen

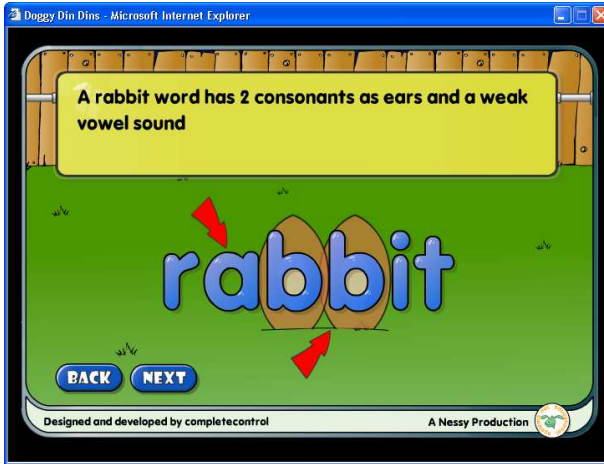


Figure 2.30 'Doggy Din Dins' Learning Rules Screen



Figure 2.31 'Doggy Din Dins' Game Screen 1



Figure 2.32 'Doggy Din Dins' Game Screen 2



Figure 2.33 'Doggy Din Dins' Feedback Screen

2.3.3 Squish'em



Figure 2.34 'Squish'em' Introduction Screen

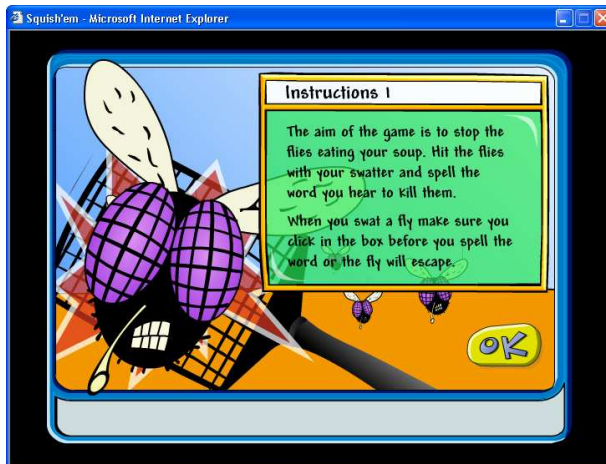


Figure 2.35 'Squish'em' Instructions Screen



Figure 2.36 'Squish'em' Game Screen 1

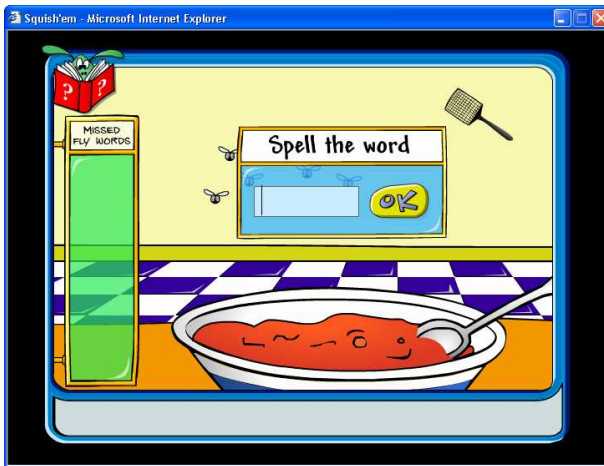


Figure 2.37 'Squish'em' Game Screen 2

3 Design Appendix

3.1 Prototype 1 - Screen Shots

What follows is a selection of screenshots from the prototype used for the first iteration of the user prototype evaluations.

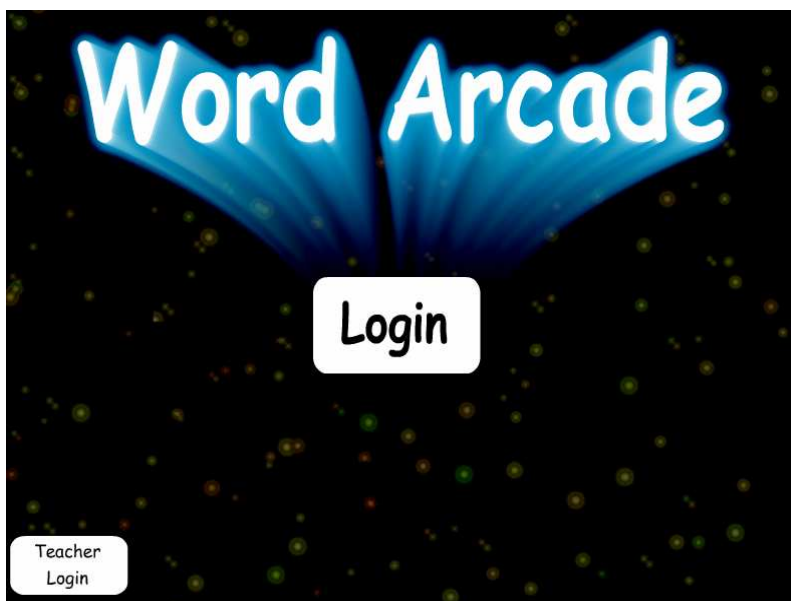


Figure 3.1. Prototype 1 – Intro Page

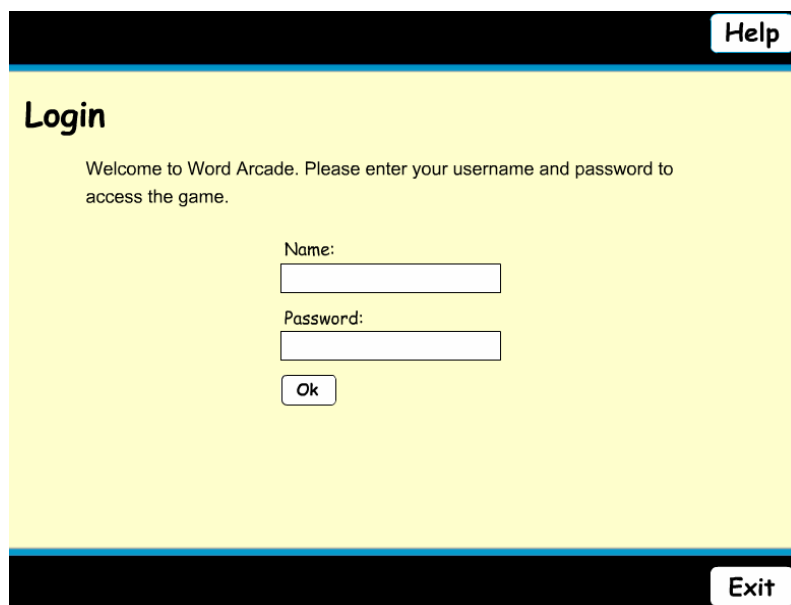


Figure 3.2. Prototype 1 – Login Page

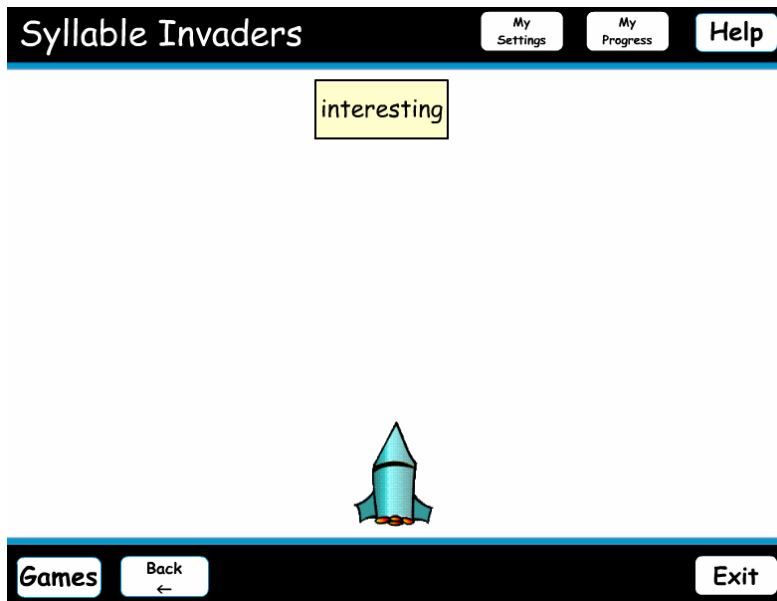


Figure 3.3 Prototype 1 – Syllable Invaders I

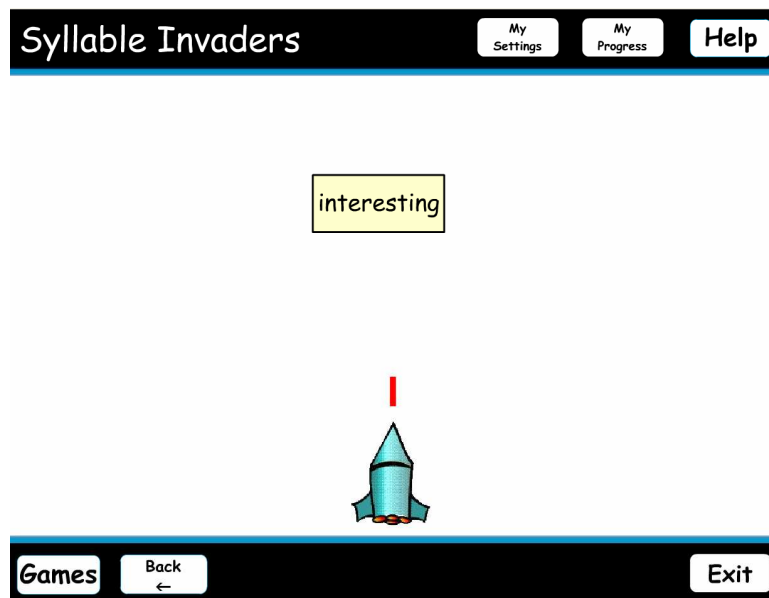


Figure 3.4. Prototype 1 – Syllable Invaders II

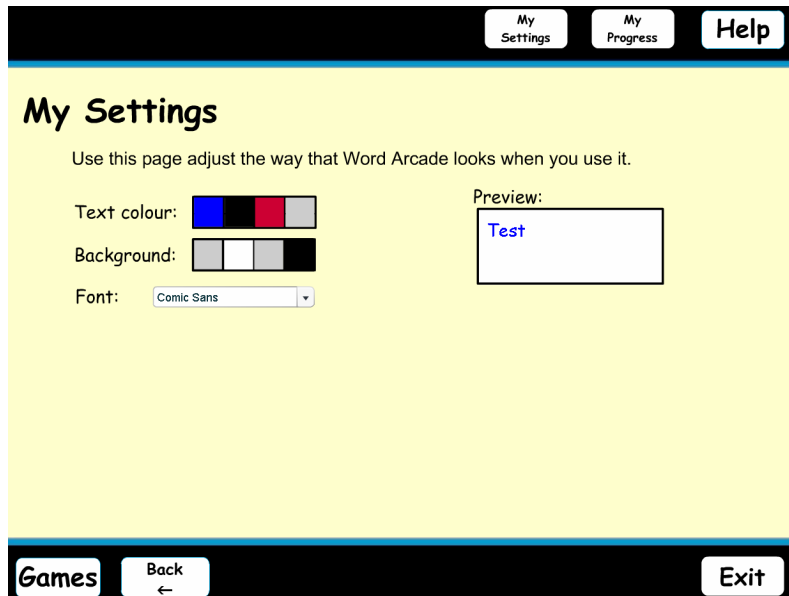


Figure 3.5. Prototype 1 – Settings Page

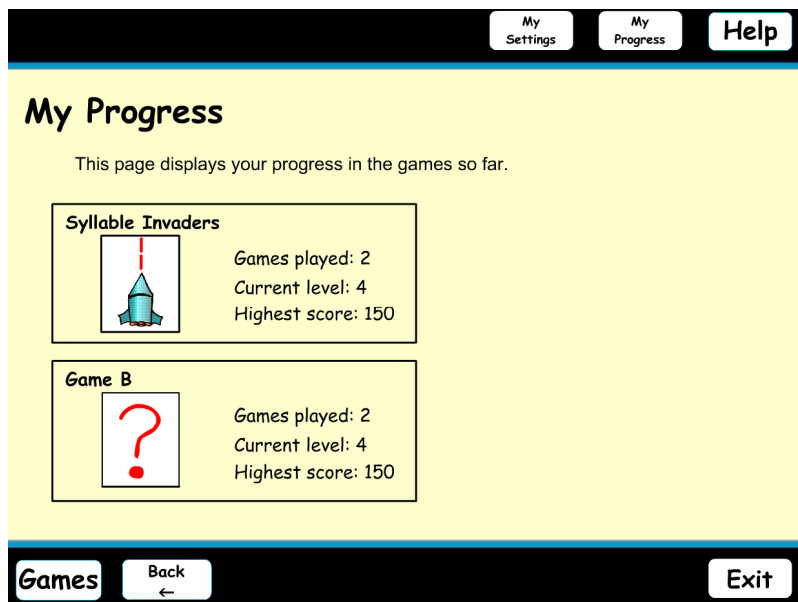


Figure 3.6. Prototype 1 – Progress Page



Figure 3.7. Prototype 1 – Button Behaviour and Tool Tips

3.2 Prototype 2 – Screen Shots

Below is a selection of screen shots from the second evaluation prototype.

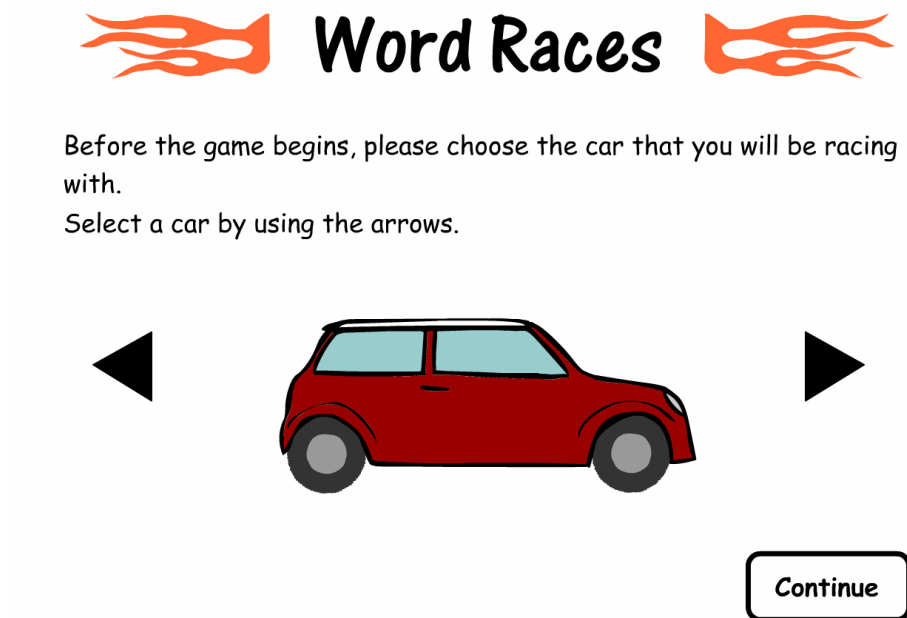


Figure 3.8. Prototype 2 – Intro Screen

Games Page

Welcome to Word Races. The aim of the game is to modify your car. To do this you need to play the games.

To select a game, click on its button below (more games will appear soon).

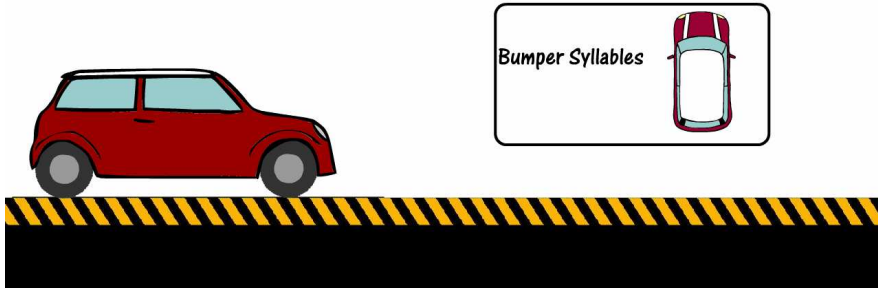


Figure 3.9 Prototype 2 – Games Page

Bumper Syllables

Aim

Bump into blocks that contain words with the same number of syllables as shown in the corner of the screen.

Controls

Use the direction keys to move the car around the screen.

Score

1 point for each correct bump. Life will be lost for hitting an incorrect block or for missing the correct ones.

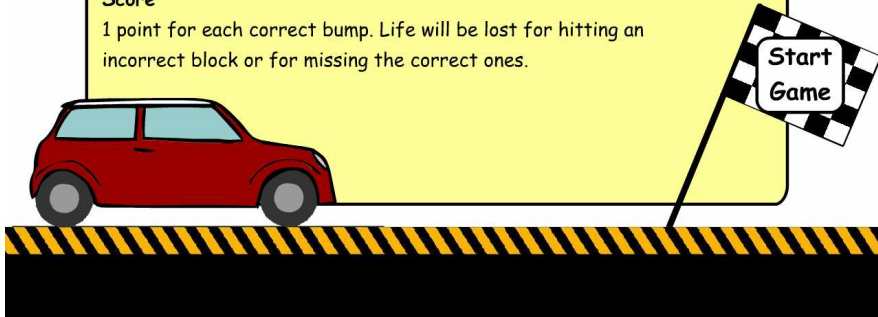


Figure 3.10 Prototype 2 – Bumper Syllables Instructions

Bumper Syllables

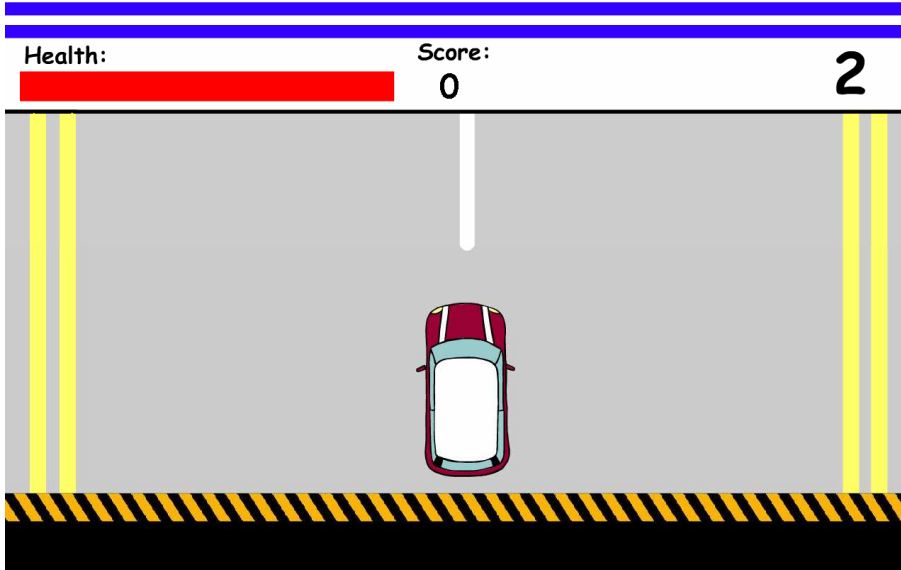


Figure 3.11 Prototype 2 – Bumper Syllables Game

The Pitstop Garage

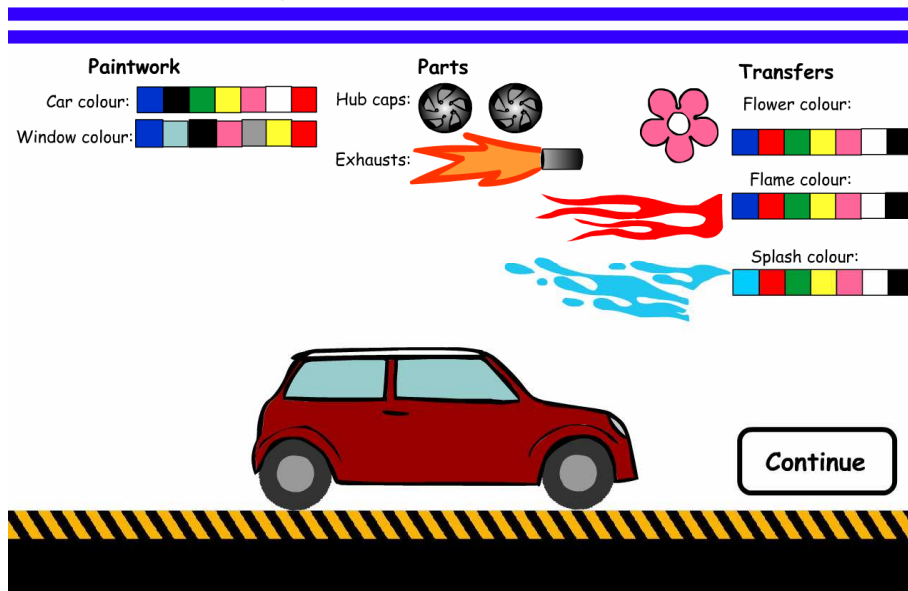
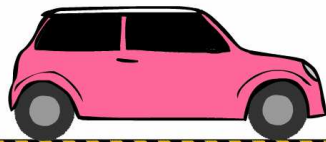


Figure 3.12 Prototype 2 – Reward Scheme Garage

Games Page

Welcome to Word Races. The aim of the game is to modify your car. To do this you need to play the games.

To select a game, click on its button below (more games will appear soon).



Bumper Syllables



Figure 3.13 Prototype 2 – Car after colour modification

4 Implementation Appendix

4.1 System Overview and Navigation

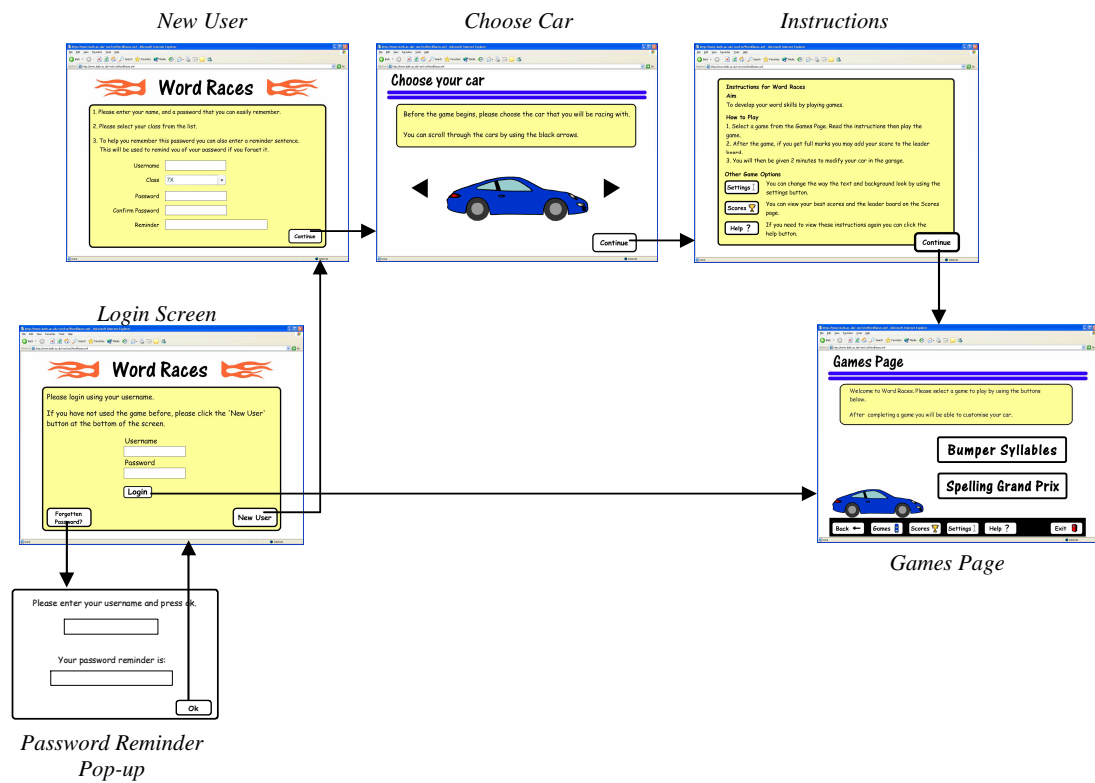


Figure 4.1 Log-in and Introduction

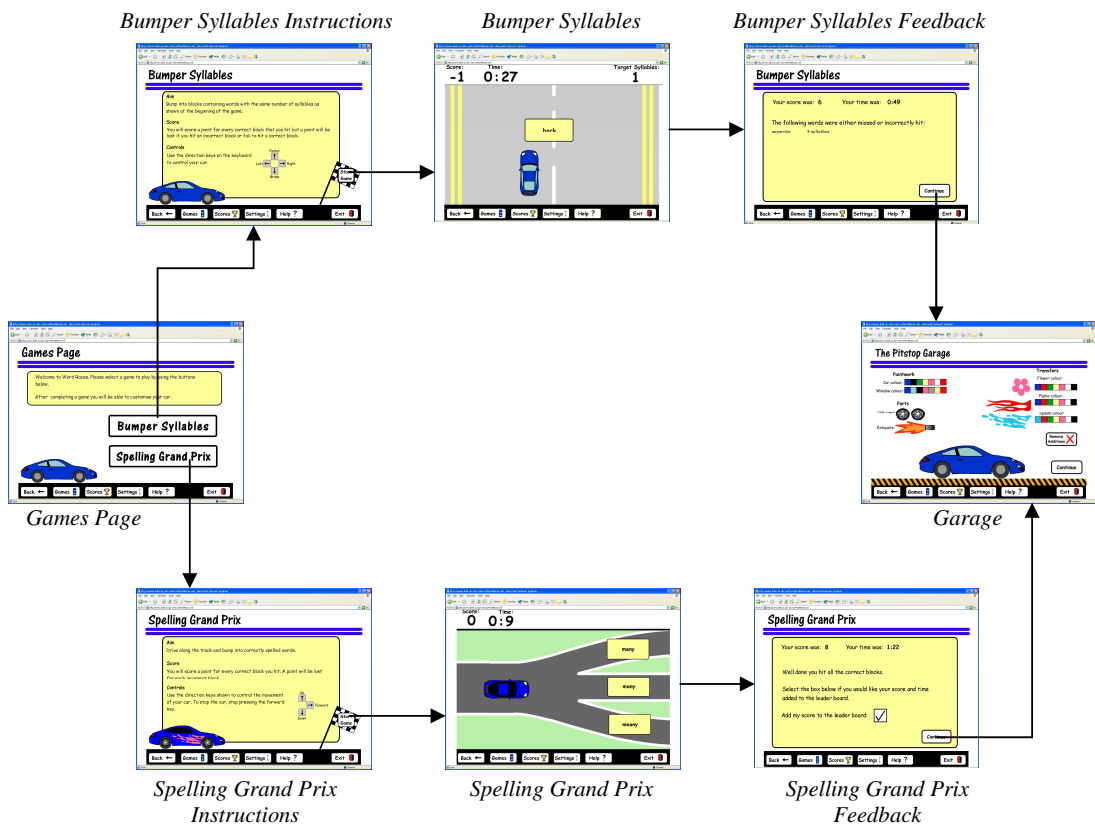


Figure 4.2 Game Navigation

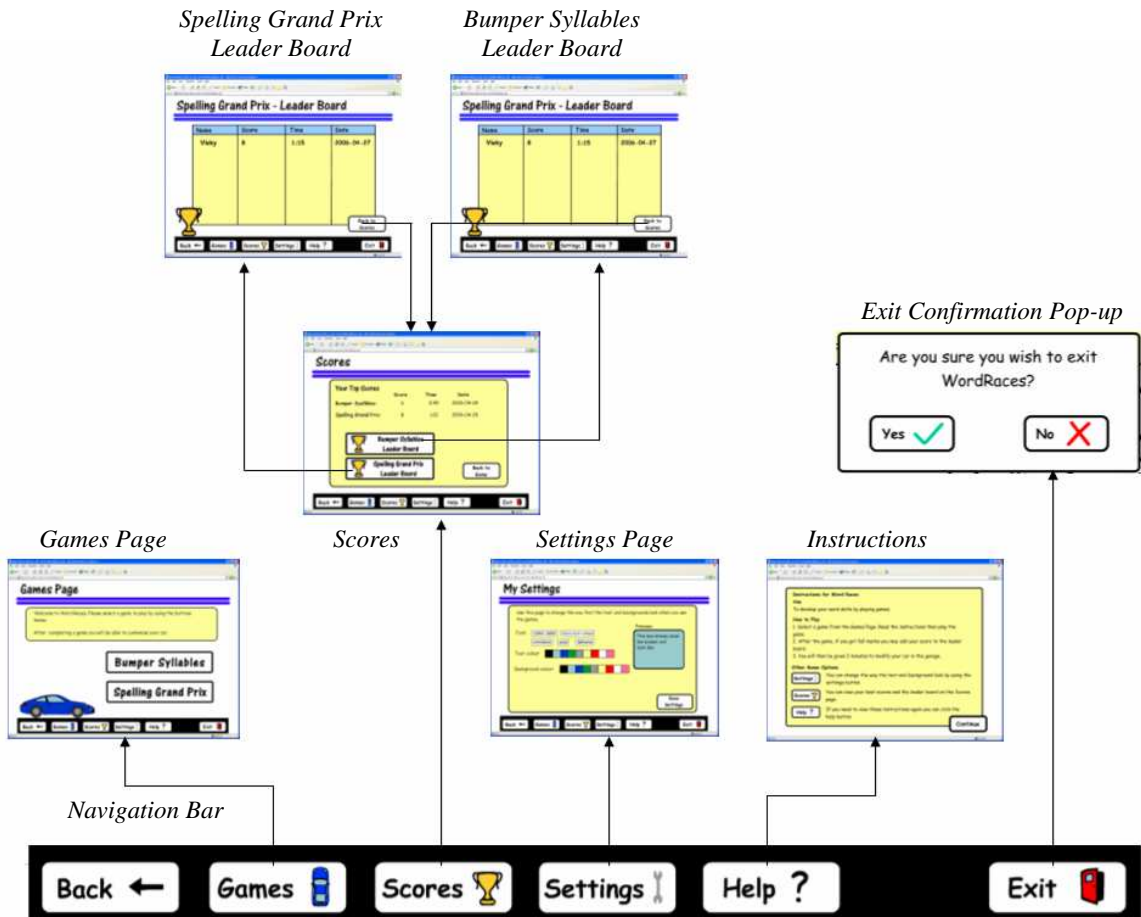
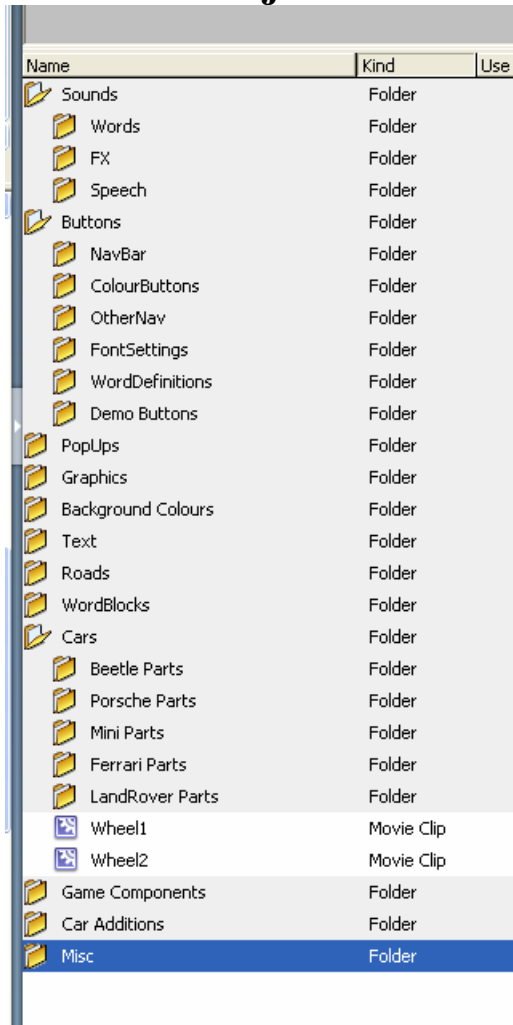


Figure 4.3 Navigation Bar and Related Screens

4.2 Game Objects



Name	Kind	Use
Sounds	Folder	
Words	Folder	
FX	Folder	
Speech	Folder	
Buttons	Folder	
NavBar	Folder	
ColourButtons	Folder	
OtherNav	Folder	
FontSettings	Folder	
WordDefinitions	Folder	
Demo Buttons	Folder	
PopUps	Folder	
Graphics	Folder	
Background Colours	Folder	
Text	Folder	
Roads	Folder	
WordBlocks	Folder	
Cars	Folder	
Beetle Parts	Folder	
Porsche Parts	Folder	
Mini Parts	Folder	
Ferrari Parts	Folder	
LandRover Parts	Folder	
Wheel1	Movie Clip	
Wheel2	Movie Clip	
Game Components	Folder	
Car Additions	Folder	
Misc	Folder	

Figure 4.4 Object Library

4.3 Layers

4.3.1 Intro Scene

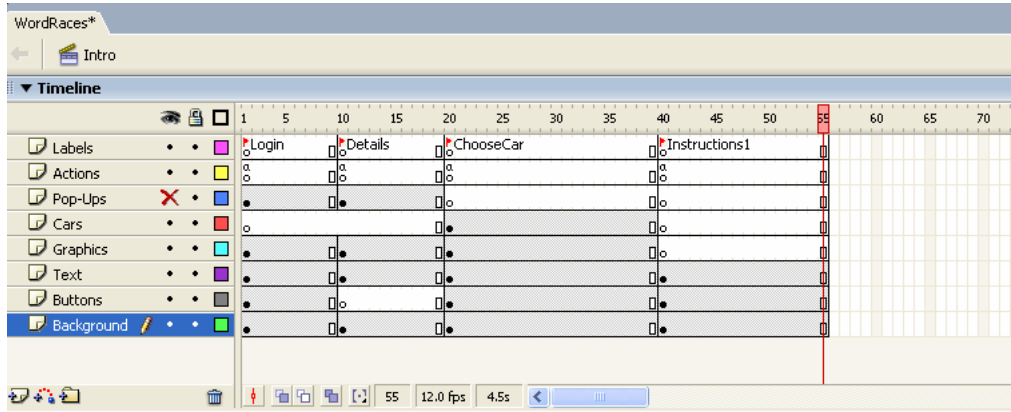


Figure 4.5 'Intro' Layers and Key Frames

4.3.2 Main Scene

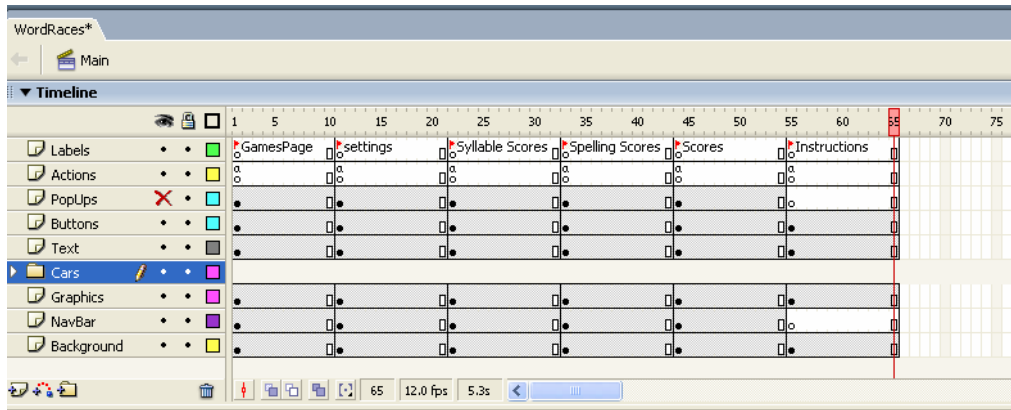


Figure 4.6 'Main' Layers and Key Frames

4.3.3 Bumper Syllables

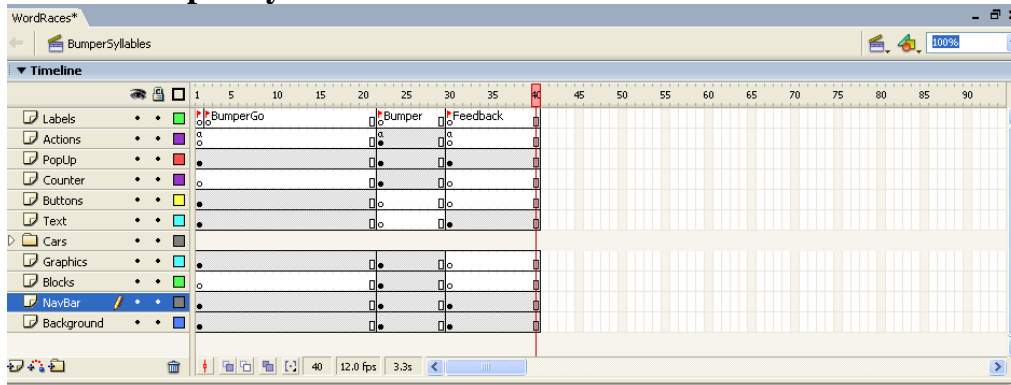


Figure 4.7 'Bumper Syllables' Layers and Key Frames

4.3.4 Spelling Grand Prix

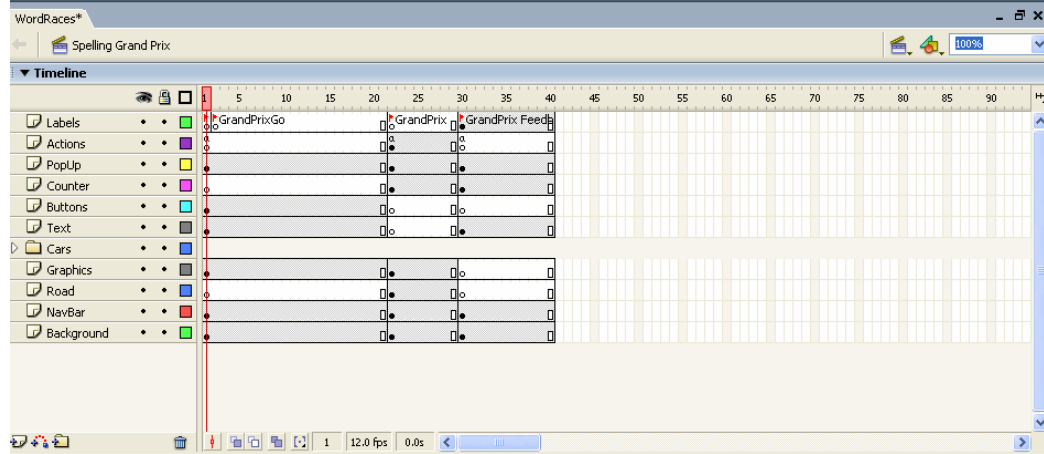


Figure 4.8 'Spelling Grand Prix' Layers and Key Frames

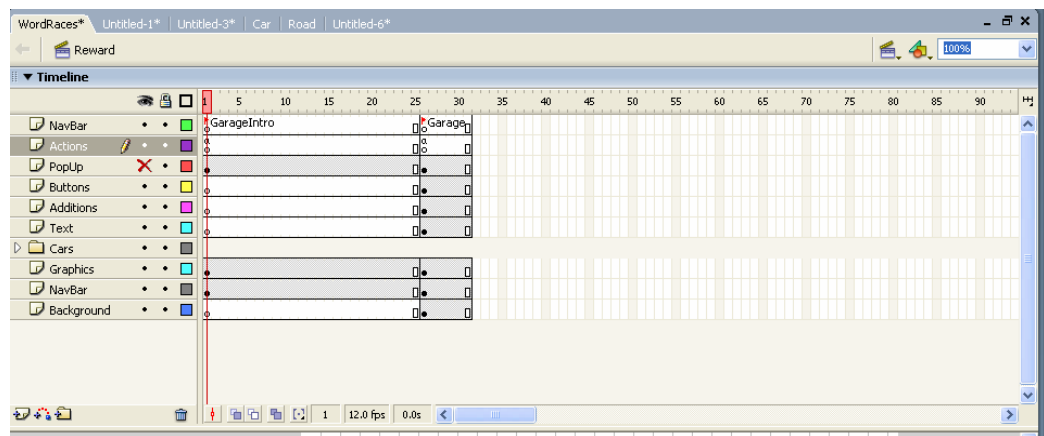


Figure 4.9 'Reward Scheme' Layers and Key Frames

4.4 Database E-R Diagram

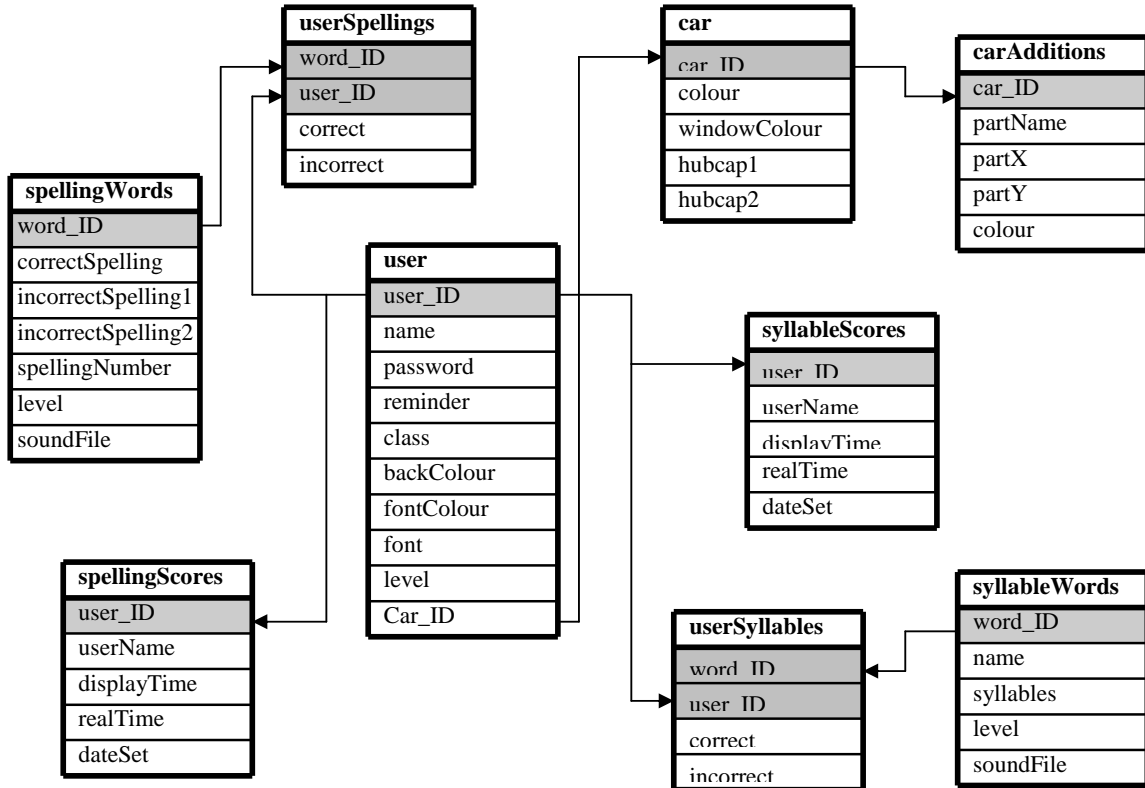


Figure 4.10 Database Entity Relationship Diagram

4.5 Car Graphics

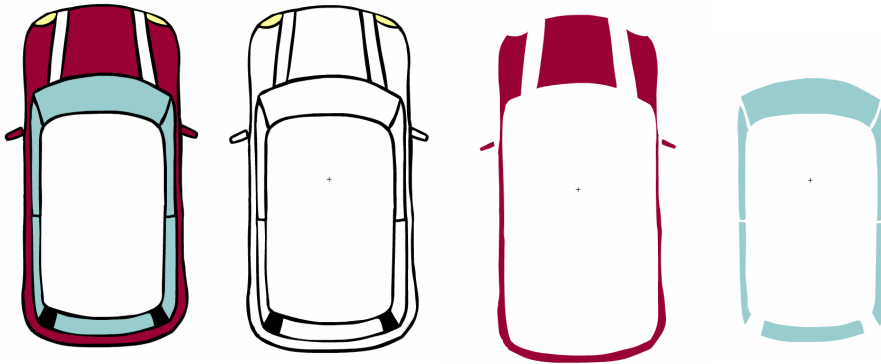


Figure 4.11 Movie Clips within the miniTop

4.6 Classes

4.6.1 Car Class

```
class Car extends MovieClip{
    public var bodyColour: String;
    public var windowColour:String;
    var splash:Array;
    var flames:Array;
    var flower:Array;
    var hubcaps:Array;
    var exhaust:Array;
    var windows:MovieClip;
    var colour:MovieClip;
    var main:MovieClip;
    var wheels:MovieClip;
    var wheel1:MovieClip;
    var wheel2:MovieClip;
    var carID;
    var name;
    var additions;

    function Car(){
    }

    /* Sets look of car from global variables */
    public function setup(user):Void{
        var i = 0; //counter
        eval(_global.car).setWindowColour(_global.windowColour); //
        set window colour
        eval(_global.car).setBodyColour(_global.bodyColour); // set
        body colour
    }
}
```

```

/* If hubcaps exists - add these to the car */
if(_global.hubcap1 == "true"){
    eval(_global.car).addHubcaps("wheel1");
}
if(_global.hubcap2 == "true"){
    eval(_global.car).addHubcaps("wheel2");
}

/* Add additions to the car from global additions array
for(i = 0; i < _global.additions.length; i++){
    /* Attach object to car */
    eval(_global.car).attachMovie(_global.additions[i],
    _global.additions[i+1], _global.additions[i+2]);
    var target = _global.additions[i+1];
    eval(_global.car+"."+target)._x =
    _global.additions[i+3];
    eval(_global.car+"."+target)._y =
    _global.additions[i+4];
    _root.test2 = eval(_global.car);

    /* If addition has a set colour (i.e. if its is not an
    exhasut pipe) */
    if(_global.additions[5] != "null"){
        var colour = new Color
        (eval(_global.car+"."+target));
        // set colour
        colour.setRGB(_global.additions[i+5]);
    }
    i = i + 5; // increment to next set of additions
}
}

/* Setup colours of carTop */
public function setupTop(user):Void{
    var i = 0;

    eval(_global.car+"Top").setWindowColour(_global.windowColour);

    eval(_global.car+"Top").setBodyColour(_global.bodyColour);
}

/ * Loads car data from database */
public function loadData(user):Void{
    var i = 0; //counter

    /* Create LoadVars object and load car colour data */
    var getDetails = new LoadVars();
    getDetails.name = user;
    var Details = new LoadVars();
    getDetails.sendAndLoad("getCarDetails.php", Details);
    Details.onLoad = function() {
        / * set up car colours */
        _root.bodyColour = Details.bodyColour;
        _root.windowColour = Details.windowColour;
        eval(_global.car).setWindowColour(_root.windowColour);
    }
}

```

```

eval(_global.car).setBodyColour(_root.bodyColour);
eval(_global.car).carID = Details.ID;
_global.car_ID = Details.ID;

/* Add hubcaps */
if(Details.hubcap1 == "true"){
    _global.hubcap1 = "true"; // set global variable
    //attach hubcap to car
    eval(_global.car).addHubcaps("wheel1");
}
if(Details.hubcap2 == "true"){
    _global.hubcap2 = "true";
    eval(_global.car).addHubcaps("wheel2");
}
}

/* Create LoadVars and load car additions */
var getAdditions = new LoadVars();
getAdditions.name = user;
var Additions = new LoadVars();
getAdditions.sendAndLoad("getCarAdditions.php", Additions);
Additions.onLoad = function() {
    /* Create global array to store additions */
    eval(_global.car).additions = new Array();

    for(i = 0; i < Additions.rows; i++){
        /* Add additions to global array */
        eval(_global.car).additions.push(eval(
            "Additions.name" + i),eval("Additions.name" +
            i)+i, i+50,eval("Additions.partX" + i),
            eval("Additions.partY" + i));
        /* Attach additions to car */
        eval(_global.car).attachMovie(eval(
            "Additions.name" + i), eval("Additions.name" +
            i)+i, i+50);
        var target = eval("Additions.name" + i)+i;
        /* Set x and y coordinates */
        eval(_global.car+"."+target)._y =
            eval("Additions.partY" + i);

        eval(_global.car+"."+target)._x =
            eval("Additions.partX" + i);

        /* Set up colour if this exists */
        if(eval("Additions.colour" + i) != "null"){
            eval(_global.car).additions.push(
                eval("Additions.colour" + i));

            var colour = new
                Color(eval(_global.car+"."+target));
            colour.setRGB(eval("Additions.colour" +
                i));
        } else {
            eval(_global.car).additions.push("null");
        }
    }
}

```

```

    }
    _global.additions = eval(_global.car).additions;
}
}

/* Clear additions from database */
public function clearDatabase(user){
    /* Create LoadVars and send data */
    var removeAdditions = new LoadVars();
    removeAdditions.user = user;
    removeAdditions.carID = _global.car_ID;
    var test = new LoadVars();
    removeAdditions.sendAndLoad("removeCarAdditions.php", test);
    test.onLoad = function() {
        if(test.removed == "true"){
            _root.test = "true";
        }else{
            _root.test = "false";
        }
    }
}

/* Clear additions from car */
public function clearAdditions(user):Void{
    var i = 0;

    /* Load current additions from database */
    var getAdditions = new LoadVars();
    getAdditions.name = user;
    var Additions = new LoadVars();
    getAdditions.sendAndLoad("getCarAdditions.php", Additions);
    Additions.onLoad = function() {
        /* for each current addition, remove this movie clip
        */
        for(i = 0; i < Additions.rows; i++){
            var newAddition = eval("Additions.name" + i)+i;
            var newAddition =
            eval("_global.car."+newAddition);
            eval(newAddition).removeMovieClip();

        }
    }
    this.clearDatabase(user);
    this.removeHubcaps("wheel1");
    this.removeHubcaps("wheel2");
}

/* Add additions to database */
public function setAdditions(userName, partName, x, y, colour){
    /* Create LoadVars and send addition details to database */
    var setAdditions = new LoadVars();
    setAdditions.carID = _global.car_ID;
    setAdditions.userName = userName;
    setAdditions.partName = partName;
    setAdditions.x = x;

```

```

        setAdditions.y = y;
        setAdditions.colour = colour;
        var test = new LoadVars();
        setAdditions.sendAndLoad("setCarAdditions.php", test);
        test.onLoad = function() {
            if(test.added == true){
                trace("added");
            }else{
                trace("not added");
            }
        }
    }
}

/* Update car colours in database */
public function setColours(user):Void{
    var setColours = new LoadVars();
    setColours.name = user;
    setColours.carColour = bodyColour;
    setColours.windowColour = windowColour;
    var Results = new LoadVars();
    setColours.sendAndLoad("setColours.php", Results);
    Results.onLoad = function() {
        if(Results.added == true){
            trace("added");
        }else{
            trace("not added");
        }
    }
}

/* Alter car body colour */
public function setBodyColour(newColour):Void {
    _global.bodyColour = newColour;
    bodyColour = newColour;
    var carColour = new Color(main.colour);
    carColour.setRGB(newColour);
}

/* Alter car window colour */
public function setWindowColour(newColour):Void {
    _global.windowColour = newColour;
    windowColour = newColour;
    var windowsColour = new Color(main.windows);
    windowsColour.setRGB(newColour);
}

/* Set hubcaps value in database */
public function setHubcaps(user, wheel) {
    var setHubcaps = new LoadVars();
    setHubcaps.name = user;
    if (wheel == "wheel1") {
        setHubcaps.hubcap = "hubcap1";
    }
    if (wheel == "wheel2") {
        setHubcaps.hubcap = "hubcap2";
    }
}

```



```

    }
    var Results = new LoadVars();
    setHubcaps.sendAndLoad("setHubcaps.php", Results);
    Results.onLoad = function() {
        if(Results.added == true){
            trace("added");
        }else{
            trace("not added");
        }
    }
}

/* Attach hubcaps to car */
public function addHubcaps(wheel) {
    if (wheel == "wheel1") {
        eval(_global.car).wheels.wheel1.attachMovie("hubcap",
            "hubcap1", 40);
    }
    if (wheel == "wheel2") {
        eval(_global.car).wheels.wheel2.attachMovie("hubcap",
            "hubcap2", 41);
    }
}

/* Remove hubcaps from car */
public function removeHubcaps(wheel) {
    if (wheel == "wheel1") {

eval(_global.car).wheels.wheel1.hubcap1.removeMovieClip();
    }
    if (wheel == "wheel2") {

eval(_global.car).wheels.wheel2.hubcap2.removeMovieClip();
    }
}

/* Register a users car in the database */
public function register(user, carName):Void{
    var registerCar = new LoadVars();
    registerCar.user = user;
    registerCar.name = carName;
    registerCar.carColour = _global.bodyColour;
    registerCar.windowColour = _global.windowColour;
    var test = new LoadVars();
    registerCar.sendAndLoad("registerCar.php", test);
    test.onLoad = function() {
        if (test.added == true){

        } else {

        }
    }
}
}

```

```

/* Set the visibility of a car when it is first loaded. This calls
loadData */
public function setVisibility1():Boolean{
    /* If this car is the current users car */
    if(eval(_global.car) == eval(this._name)){
        this._visible = true; // make it visible
        this.loadData(_global.user);
        return true;
    }
    else {
        this._visible = false;
        return false;
    }
    _global.visited = true;
}

/* Set the visibility of the car on subsequent screens. This calls
setup */
public function setVisibility():Boolean{
    /* If this car is the current users car */
    if(eval(_global.car) == eval(this._name)){
        this._visible = true; // make it visible
        this.setup(_global.user);
        return true;
    }
    else {
        this._visible = false;
        return false;
    }
}

/* Setup the top visibility of the car */
public function setTopVisibility():Boolean{
    if(eval(_global.car+"Top") == eval(this._name)){
        this._visible = true;
        this.setupTop(_global.user);
        return true;
    }
    else {
        this._visible = false;
        return false;
    }
}
}
}
}

```

4.6.2 Road Class

```

class Road extends MovieClip{
    var started:Boolean;
    var finished:Boolean;
    var orderArray; // Stores an array of road instance names
                    that will be activated during the game
    var count; // counts the number of roads that have been
activated

```

```

function Road(){
    started = false;
    finished = false;
}

/* Activates the road instance */
public function start(orderArray, startNumber):Void{
    // set edge object for keeping the car on the track
    _root.edge1 = orderArray[count];
    // activate the road
    started = true;
    this.orderArray = orderArray;
    count = startNumber;
}

/* De-activates the road and activates the next instance of
road */
public function stop():Void{
    // activate next road instance
    orderArray[count+1].start(orderArray, count+1);
    // set next edge object
    _root.edge2 = orderArray[count];
    /* If this is the last road instance, then activate
the finish line movie clip */
    if(count == orderArray.length-1){
        _root.ended = true;
    }
}

/* Returns the status of the road */
public function getStatus():Boolean{
    return started;
}
}

```

4.6.3 WordBlock Class

```

class WordBlock extends MovieClip{
    private var BlockText:MovieClip;
    private var BlockColour:MovieClip;
    var syllables:Number;
    var soundLoc:String;
    var word:String;
    var speed:Number;
    var sound;
    var started:Boolean;
    var match:Boolean;
    var word_ID;
    var soundPlayed:Boolean;

    function WordBlock(){
    }

    public function fillDetails(data_ID, data_syllables, data_word,
data_sound, data_match):Void{

```

```

        /* Set blocks font and colours to match users display
settings */
        var colourObject = new Color (this.BlockColour);
        colourObject.setRGB(_global.backgroundColour);
        colourObject = new Color (this.BlockText);
        colourObject.setRGB(_global.textColour);
        var textFormat = new TextFormat();
        textFormat.font = _global.fontName;
        this.BlockText.setTextFormat(textFormat);

        /* Set block variables */
        sound = data_sound;
        word = data_word;
        syllables = data_syllables;
        BlockText.text = word;
        soundLoc = data_sound;
        started = false;
        match = data_match;
        word_ID = data_ID;
        _visible = false;
        soundPlayed = false;
    }

    /* Play sound clip of word */
    public function playSound():Void {
        this.sound = new Sound();
        this.sound.loadSound("Words/"+this.soundLoc, true);
        this.sound.play();
    }

    /* Remove block from screen */
    public function removeBlock():Void {
        this.swapDepths(1048000);
        this.removeMovieClip();
    }

    /* Check if syllables match when block hits the bottom of the
screen */
    public function navSyllableCheck():Void{
        _root.blockCounter --; // count block
        if(match == true){
            _root.points--; // decrement points
            this.missed(); // add to array of incorrect hits
            for feedback
                this.removeBlock();
        } else {
            this.removeBlock();
        }
    }

    /* Check if syllables match when block hits a car */
    public function carSyllableCheck():Void {
        _root.blockCounter --; // count block
        if(match == true){
            _root.points++; // increment points

```

```

        this.pushCorrect(); // add to database
        this.removeBlock();
    } else {
        _root.points--; // decrement points
        this.missed(); // add to array of incorrect hits
        for feedback
            this.removeBlock();
    }
}

/* Add a correct hit to the database */
public function pushCorrect():Void {
    var setWords = new LoadVars();
    setWords.word_ID = this.word_ID;
    setWords.user_ID = _global.user_ID;
    setWords.result = "correct";
    var test = new LoadVars();
    setWords.sendAndLoad("setSyllables.php", test);
    test.onLoad = function() {
    }
}

/* Returns if the syllables match the game syllables */
public function syllableCheck():Boolean{
    return match;
}

/* Display block when it reaches the top of the screen and play
sound */
public function appear():Void {
    if(soundPlayed == false){
        this.sound = new Sound();
        this.sound.loadSound("Words/"+this.soundLoc, true);
        this.sound.play();
        soundPlayed = true;
    }
    this._visible = true;
}

/* Adds missed word to array for feedback */
public function missed():Void {
    _root.missed.push(word);
    _root.missedSyllables.push(syllables);
}

/* A series of functions to return the object variables */

public function getSyllables():Number {
    return syllables;
}

public function getWord():String {
    return word;
}

public function getSpeed():Number {

```

```

        return speed;
    }

    public function getSound():String {
        return soundLoc;
    }

    public function hasStarted():Boolean {
        return started;
    }

    public function startBlock():Void {
        speed = 1.5;
        started = true;
    }
}

```

4.6.4 SpellingBlock Class

```

class SpellingBlock extends MovieClip{
    private var BlockText:MovieClip;
    private var BackColour:MovieClip;
    var word:String;
    var correctWord:String;
    var correct:Boolean;
    var started:Boolean;
    var word_ID;
    var soundLoc:String;
    var sound;

    function SpellingBlock(){
    }

    /* Set up the block */
    public function fillDetails(ID, word, correctWord, correct,
    sound):Void{
        /* Set up block colours to match users display settings */
        var colourObject = new Color (this.BackColour);
        colourObject.setRGB(_global.backgroundColour);
        colourObject = new Color (this.BlockText);
        colourObject.setRGB(_global.textColour);
        var textFormat = new TextFormat();
        textFormat.font = _global.fontName;
        this.BlockText.setTextFormat(textFormat);

        /* Set block variables */
        this.word = word;
        this.correct = correct;
        this.correctWord = correctWord;
        this.word_ID = ID;
        this.BlockText.text = word;
        soundLoc = sound;
    }
}

```

```

/* Remove the block from the screen */
public function removeBlock():Void {
    this.swapDepths(1048000);
    this.removeMovieClip();
}

/* Return text for feedback */
public function getText(){
    if(correct == true){
        return "Well done. That is the correct spelling";
    } else {
        return "The correct spelling is ";
    }
}

/* Reacts to a car hit */
public function testHit():Boolean{
    /* If syllables match */
    if(this.correct == true){
        _root.points++; // increment points
        this.pushCorrect(); // add to database
        return true;
    } else if (this.correct == false){
        _root.points--; // decrement points
        _root.incorrectSpelling._visible = true; // show
        feedback pop-up
        _root.incorrectSpelling.gotoAndPlay("start");
        _root.correction = correctWord; // add to database
        return false;
    }
    this.removeBlock();
}

public function pushCorrect():Void {
    /* Update userSpellings in database */
    var setWords = new LoadVars();
    setWords.word_ID = this.word_ID;
    setWords.user_ID = _global.user_ID;
    setWords.result = "correct";
    var test = new LoadVars();
    setWords.sendAndLoad("setWords.php", test);
    test.onLoad = function() {
    }
}

/* Play sound clip of word */
public function playSound():Void {
    this.sound = new Sound();
    this.sound.loadSound("Words/"+this.soundLoc, true);
    this.sound.play();
}
}

```

4.7 Random Number Function

```
_root.generateRandom = function(limit):Array{
  numArray = new Array(limit);
  trace(numArray);
  i = 0;
  j = 0;
  _root.exists = false;
  while(i <= limit-1){
    j = 0;
    _root.exists = false;
    rand = random(limit)+1;
    while(j < limit){
      if(rand != numArray[j]){
        j++;
      }
      else {
        _root.exists = true;
        j = limit;
      }
    }
    if (_root.exists == false){
      numArray[i] = rand;
      i++;
    }
  }
  return numArray;
}
```


High Fidelity Prototype

What follows is an extensive selection of screen shots, displaying sections of the High Fidelity Prototype.

4.7.1 Login and Introduction

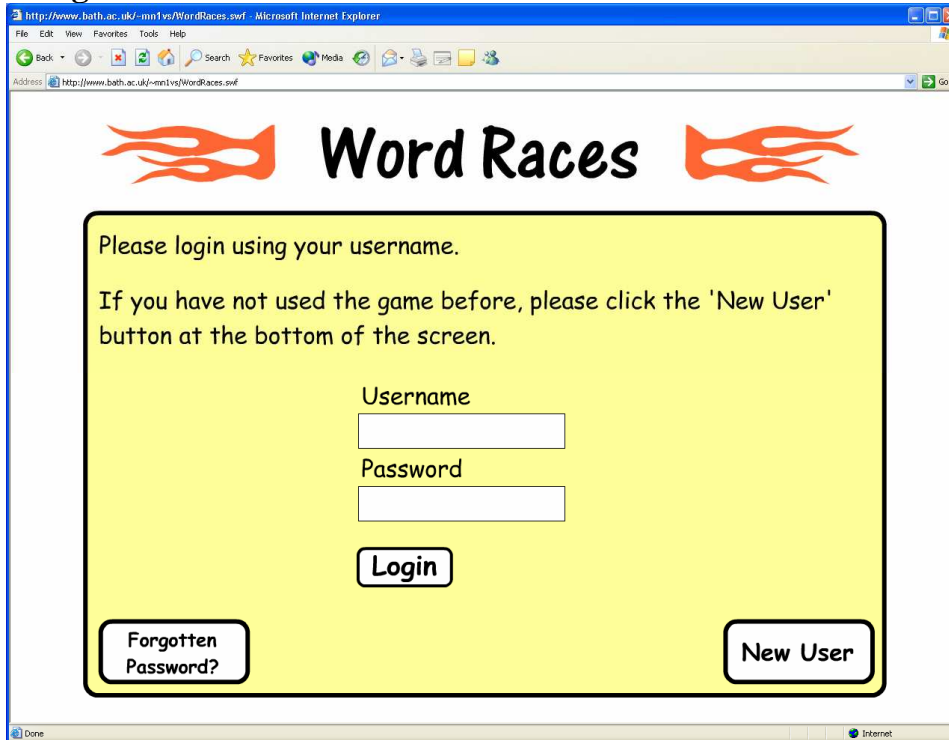


Figure 4.12 Word Races login screen

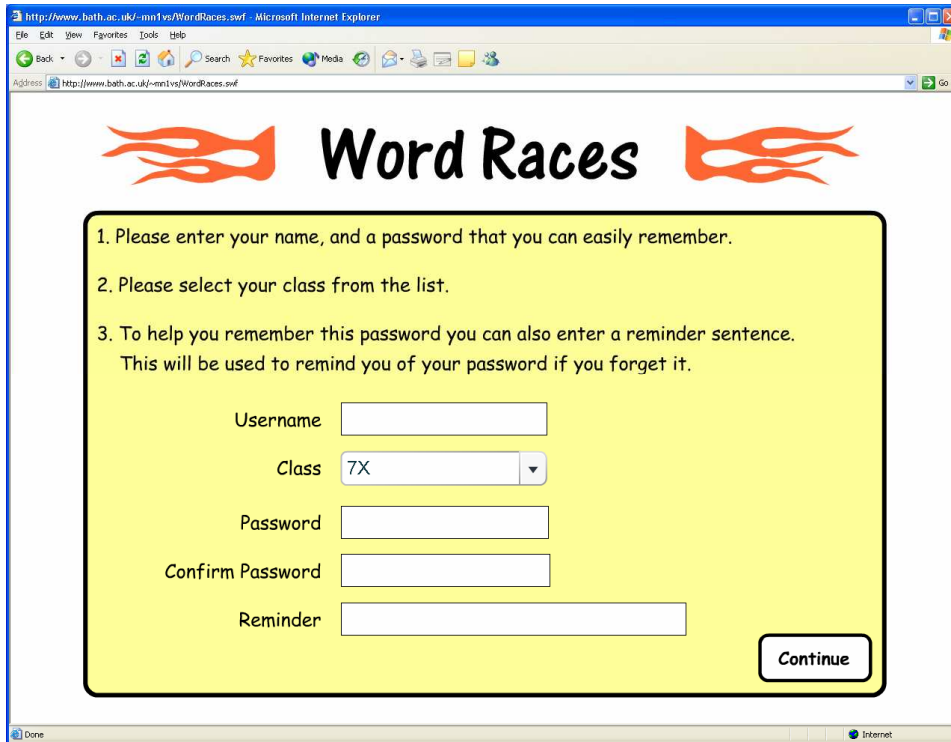


Figure 4.13 New User log in screen

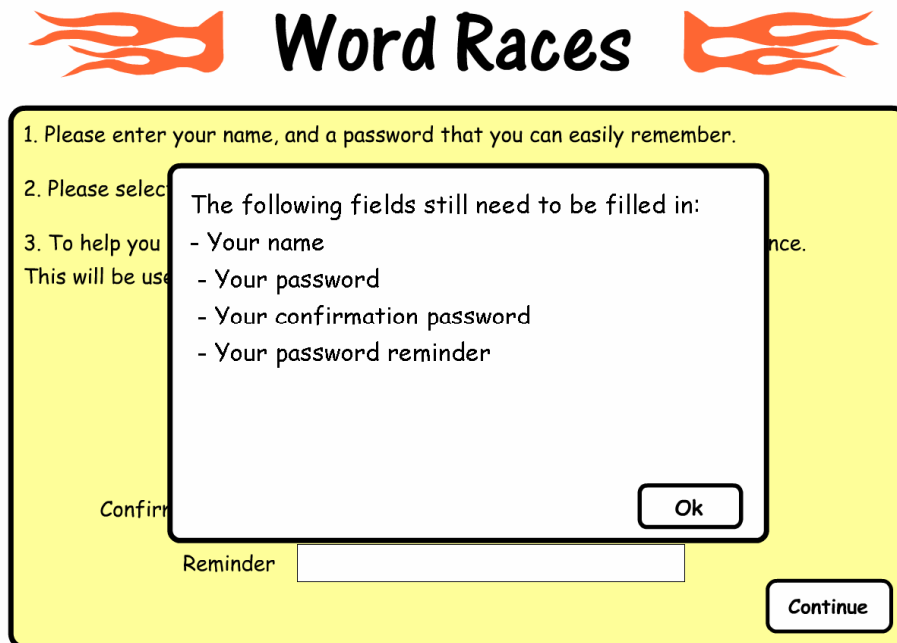


Figure 4.14 New User log in screen incomplete

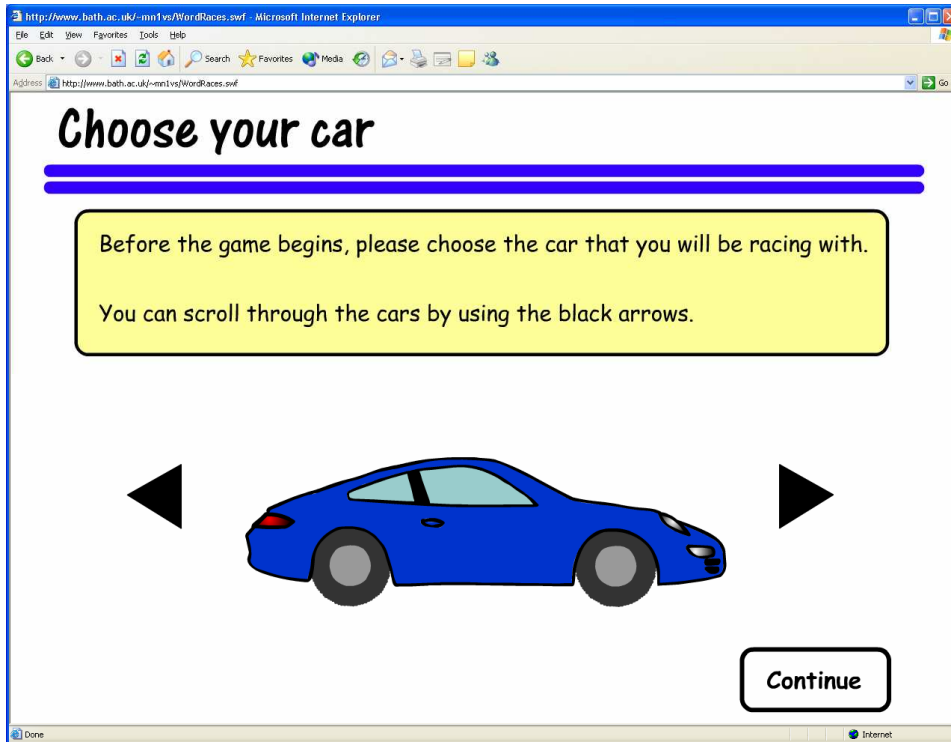


Figure 4.15 New User – Car Selector

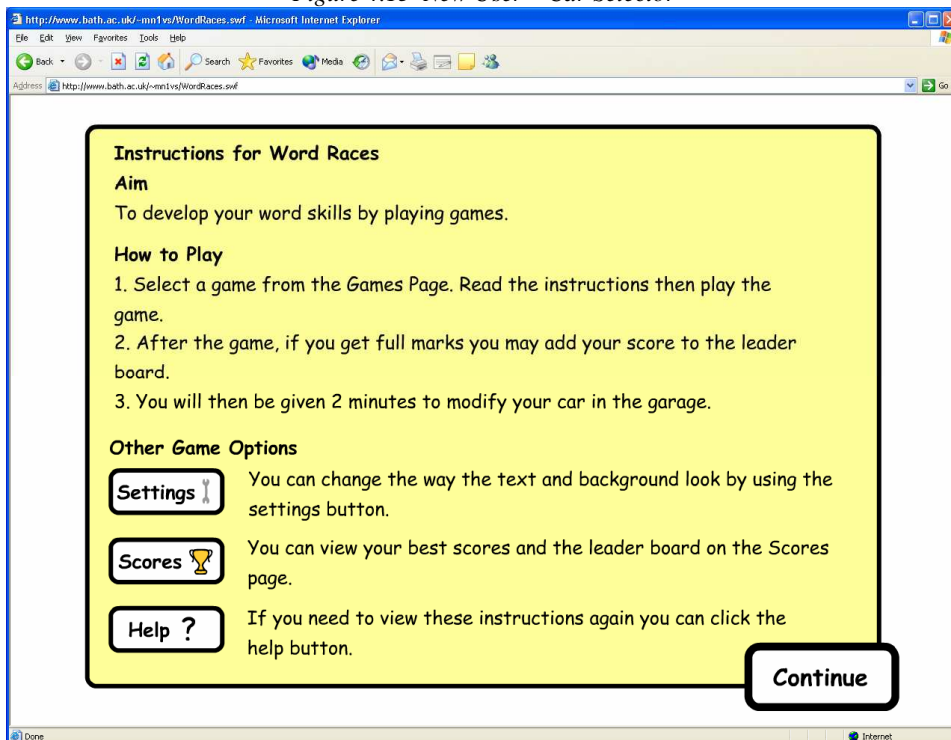


Figure 4.16 New User Instructions Page

4.7.2 Main Game Section

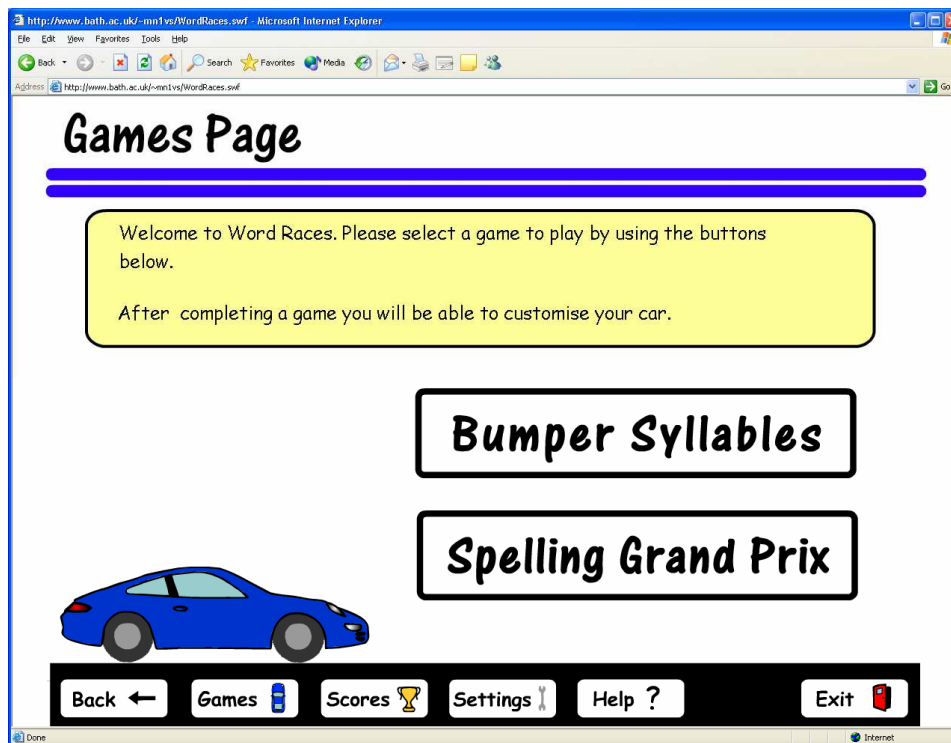


Figure 4.17 Games Page

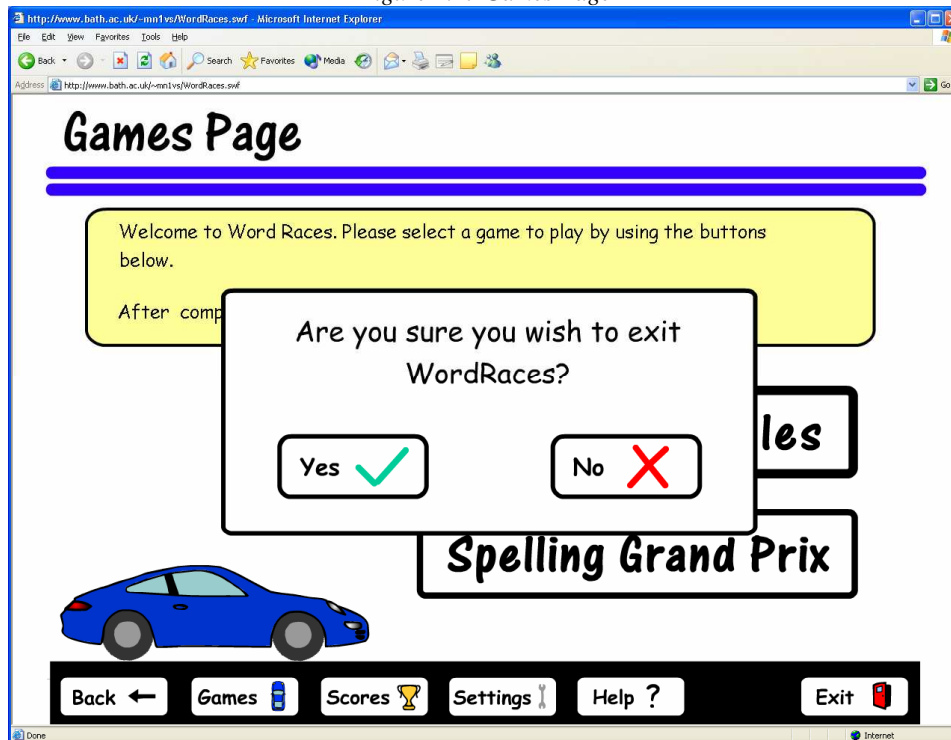


Figure 4.18 Screen Shot of exit confirmation screen

4.7.3 Bumper Syllables

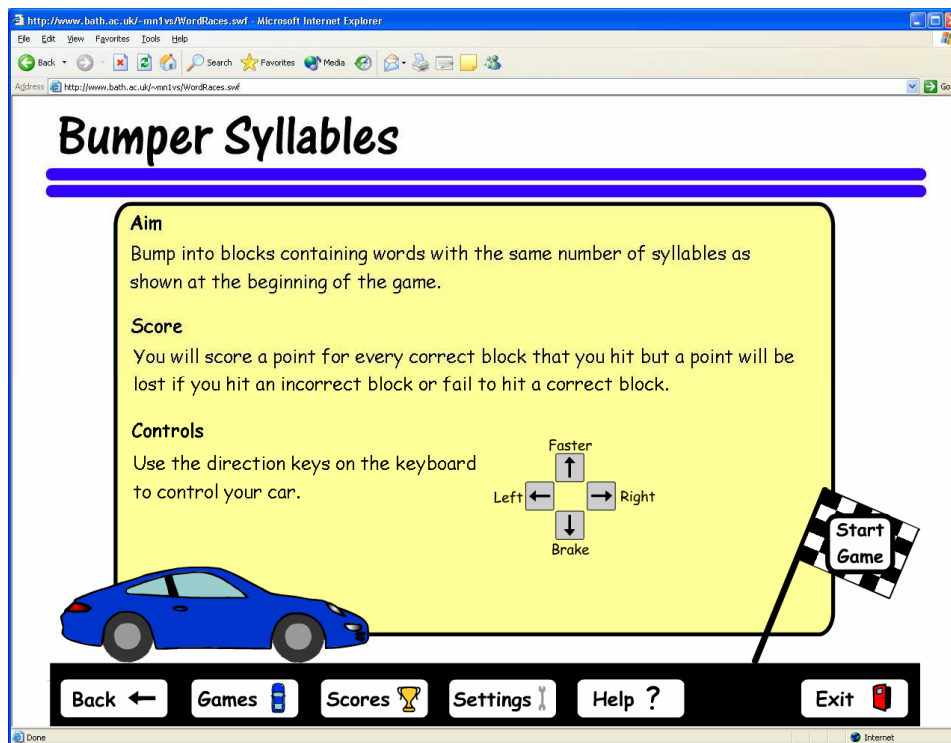


Figure 4.19 Bumper Syllables - Instructions

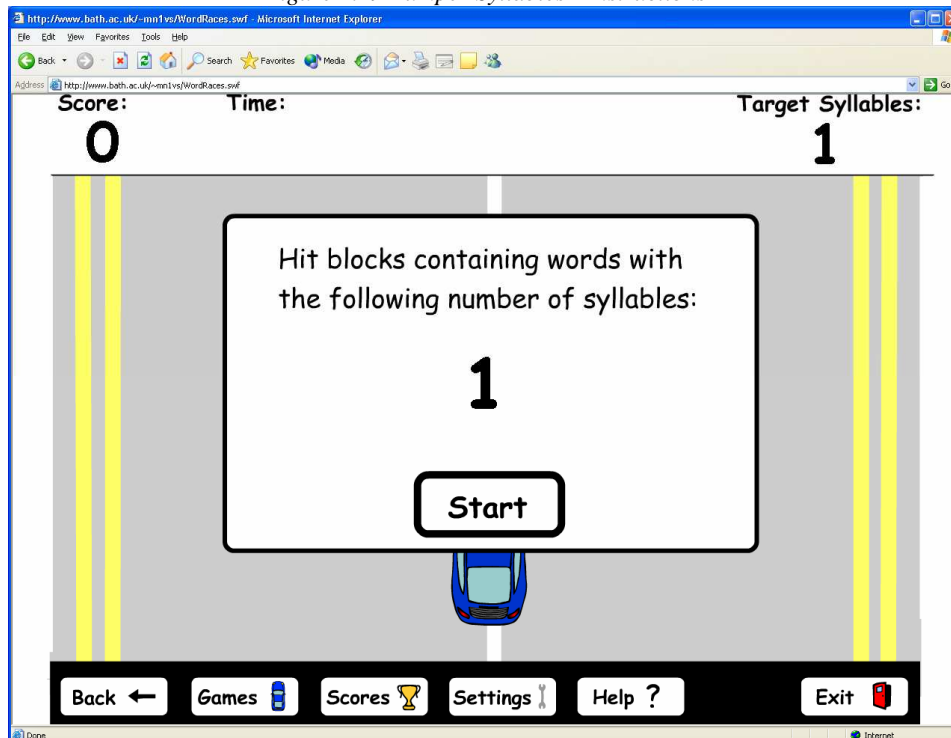


Figure 4.20 Bumper Syllables – Syllable Statement

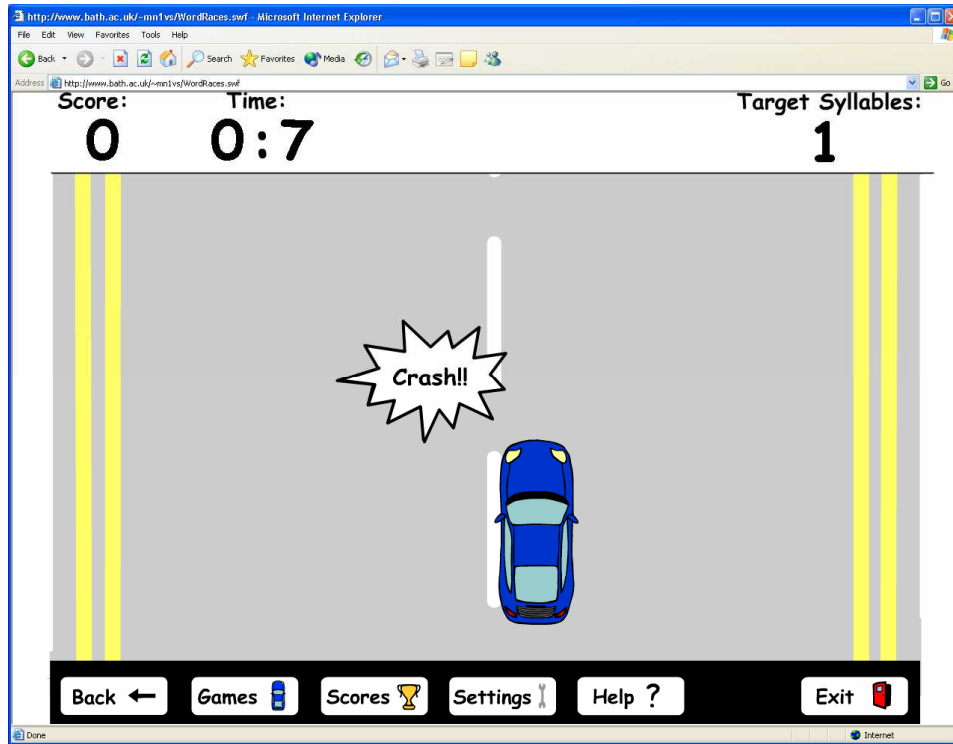


Figure 4.21 Bumper Syllables – Incorrect word hit

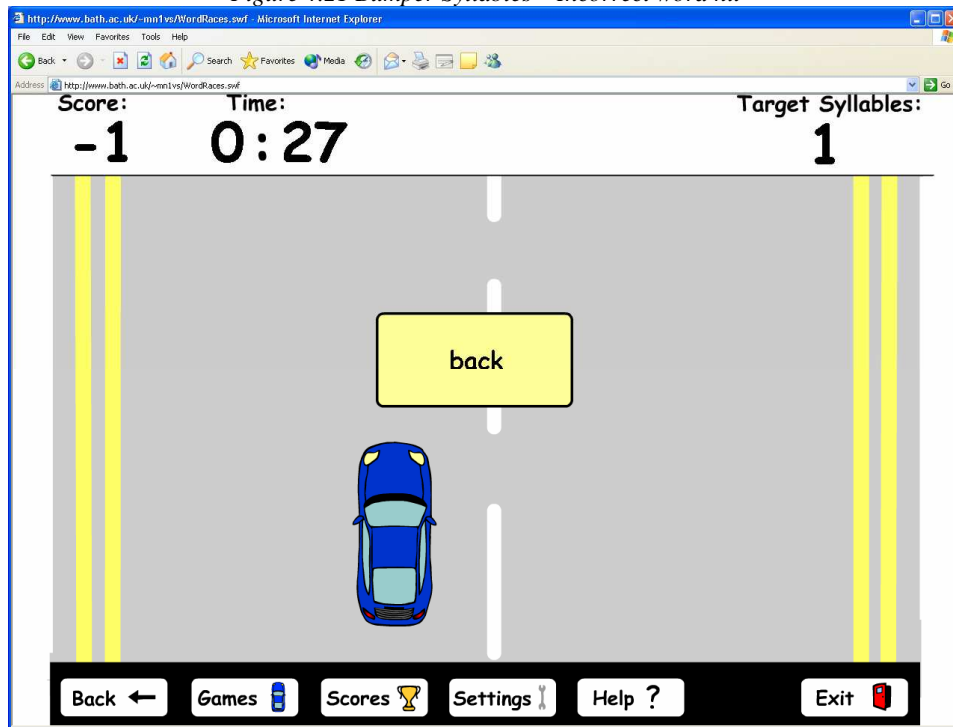


Figure 4.221 Bumper Syllables Screen Shot

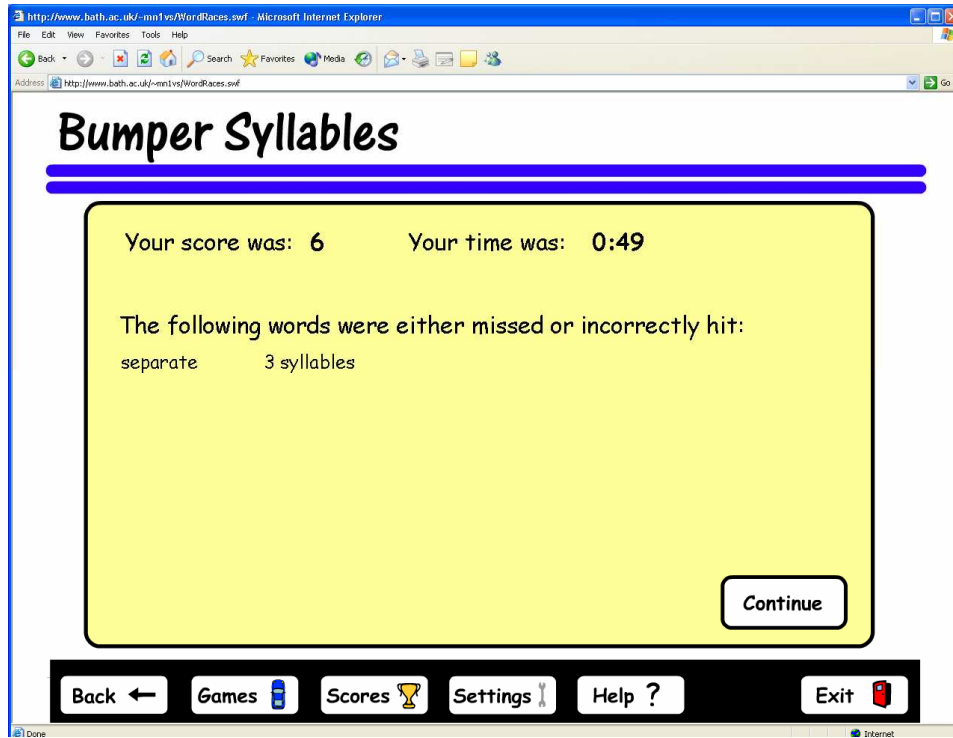


Figure 4.23 Bumper Syllables feedback with corrections

4.7.4 Garage Reward Scheme

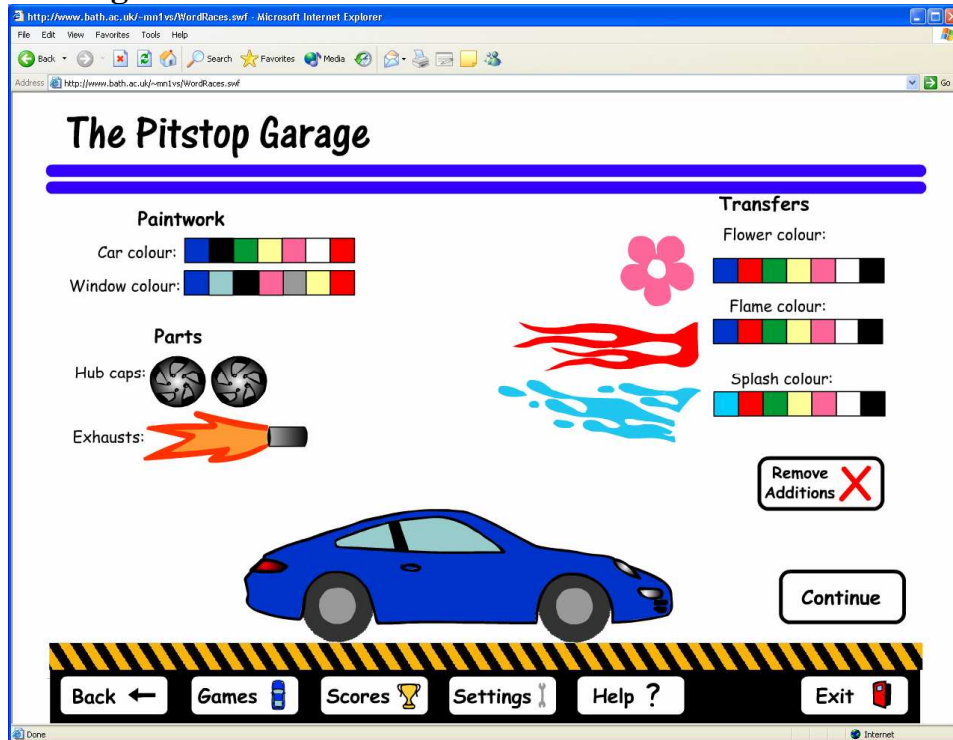


Figure 4.24 Garage before modification

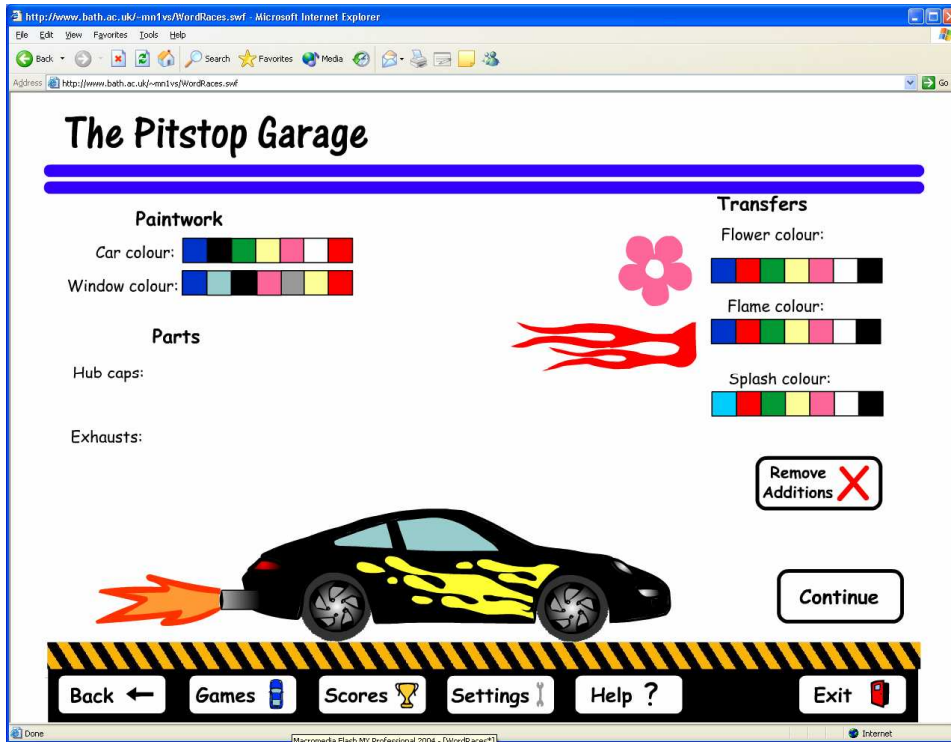


Figure 4.25 Garage after modification

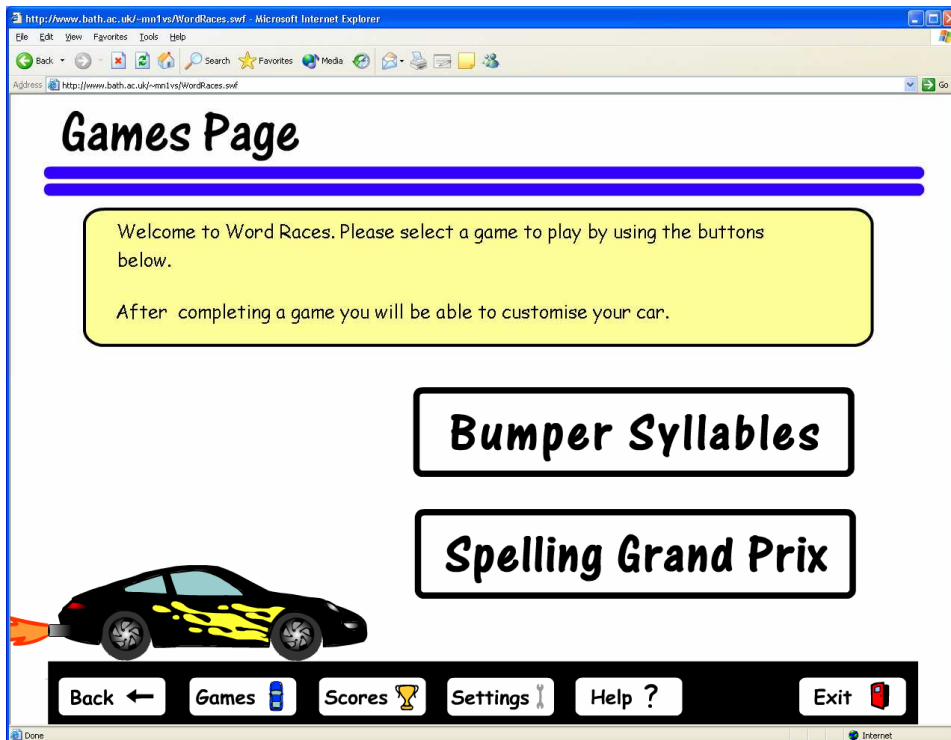


Figure 4.26 Games Page with modified car

4.7.5 Spelling Grand Prix

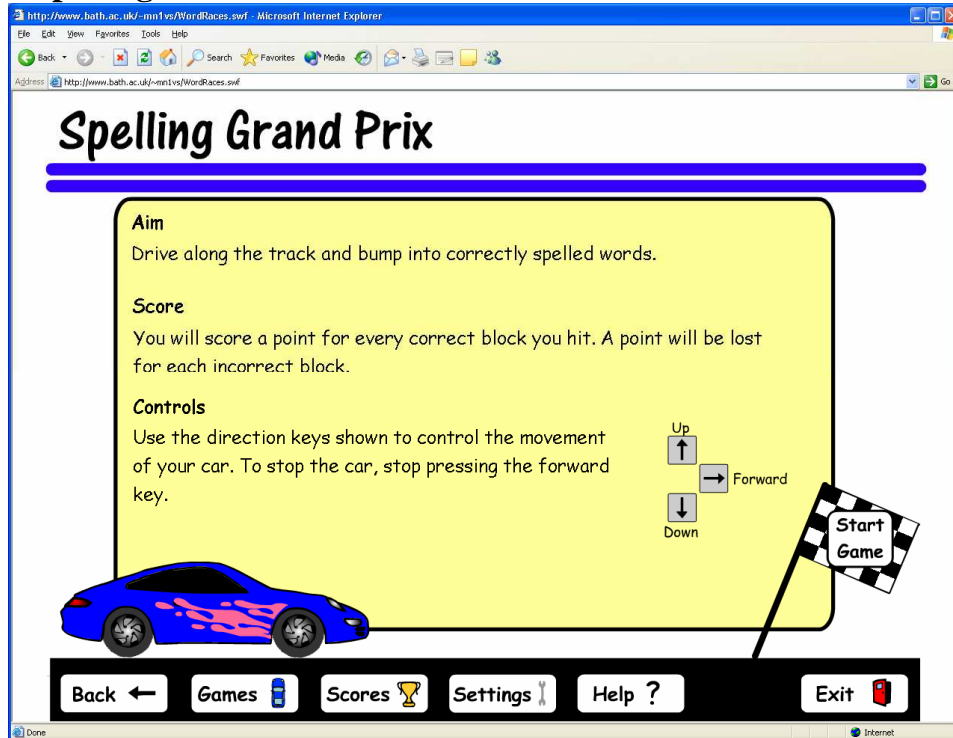


Figure 4.27 Spelling Grand Prix Instructions

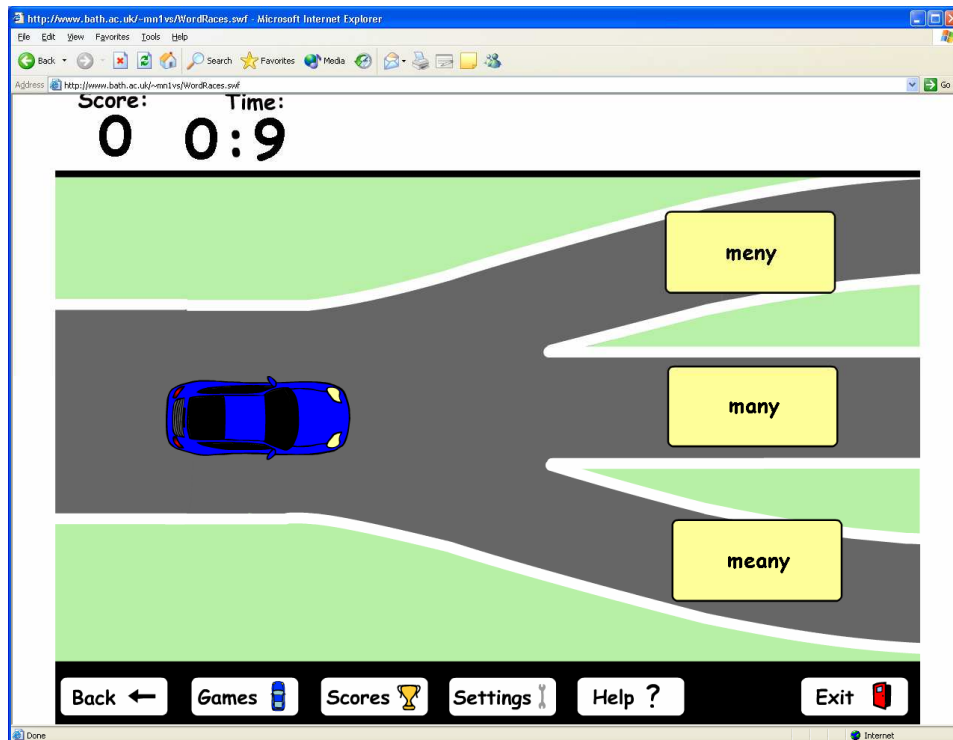


Figure 4.28 Spelling Grand Prix

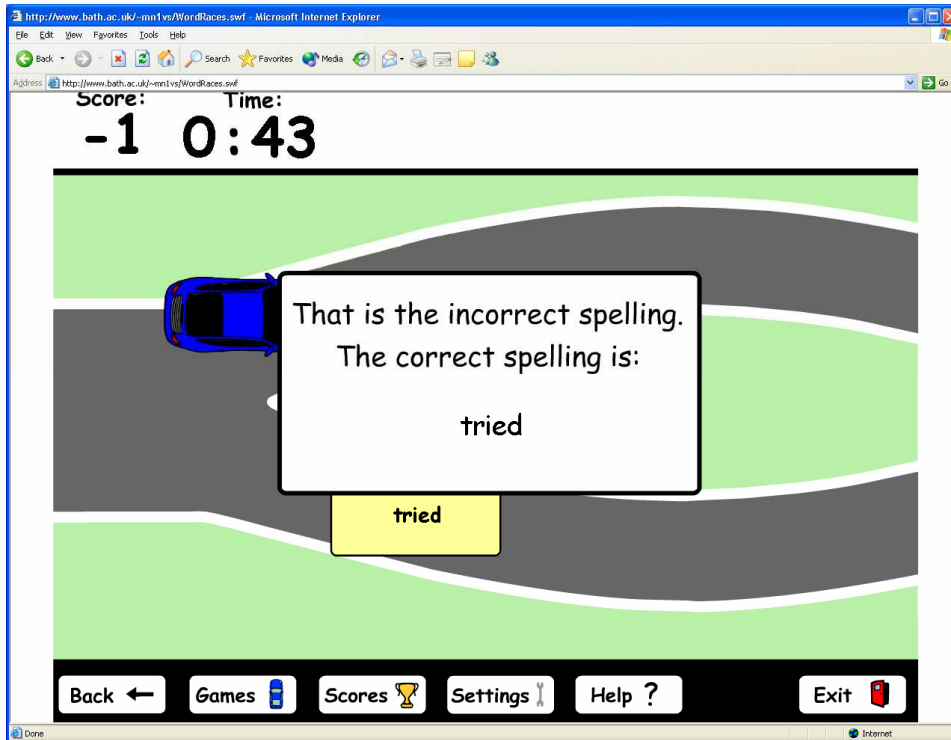


Figure 4.29 Example of instant feedback in Spelling Grand Prix

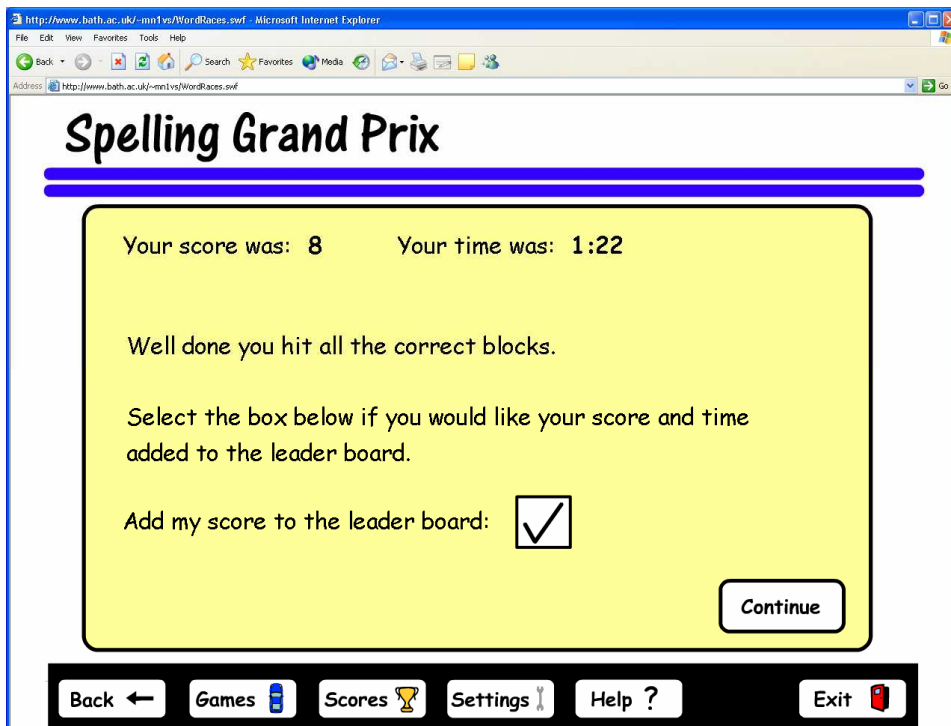


Figure 4.30 Feedback Page with all questions correct

4.7.6 Score Boards

Scores

Your Top Scores

	Score	Time	Date
Bumper Syallbles:	6	0:49	2006-04-29
Spelling Grand Prix:	8	1:22	2006-04-29

Bumper Syllables Leader Board

Spelling Grand Prix Leader Board

[Back to Game](#)

[Back](#) [Games](#) [Scores](#) [Settings](#) [Help ?](#) [Exit](#)

Figure 4.31 Scores Page

Spelling Grand Prix - Leader Board

Name	Score	Time	Date
Vicky	8	1:15	2006-04-27

[Back to Scores](#)

[Back](#) [Games](#) [Scores](#) [Settings](#) [Help ?](#) [Exit](#)

Figure 4.32 Spelling Grand Prix Leader Board

4.7.7 Display Settings

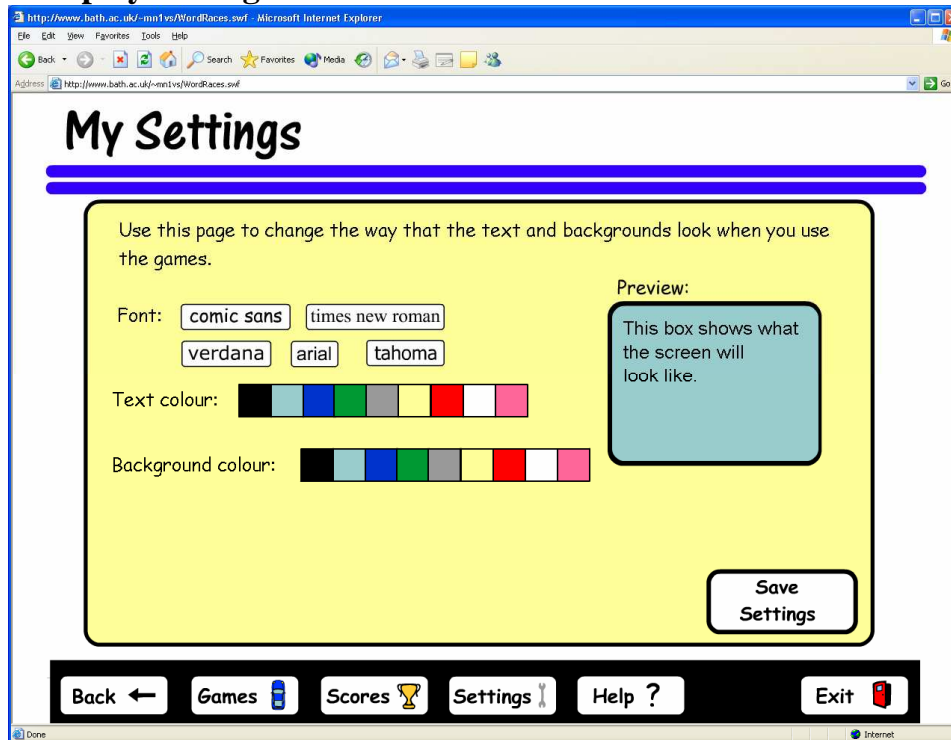


Figure 4.33 Settings Page with preview pane changed

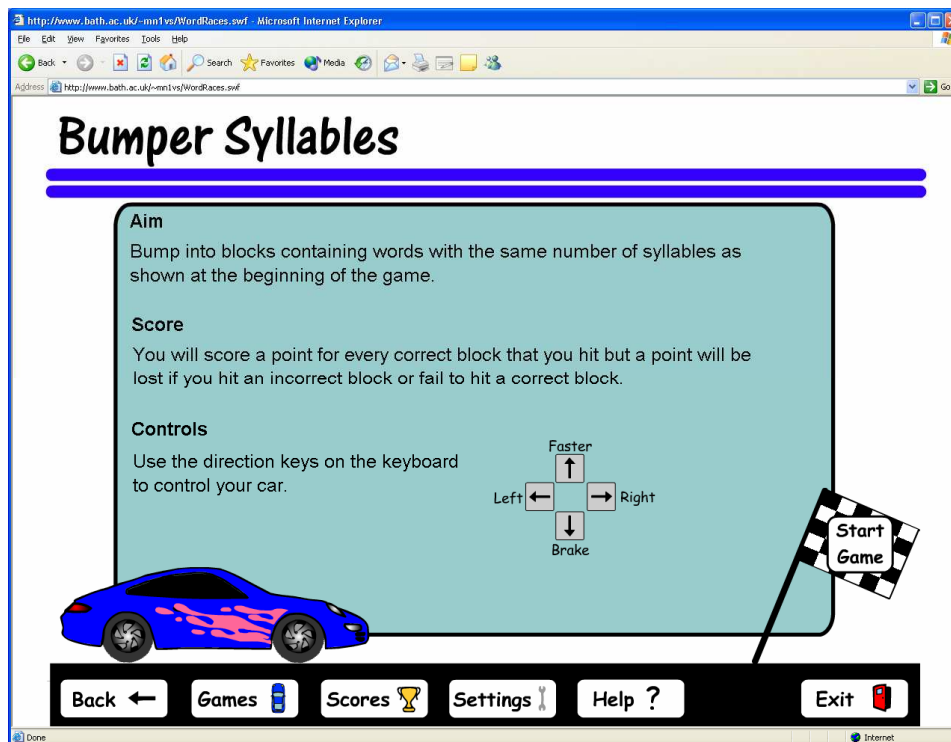


Figure 4.34 Example of the Bumper Syllables Instructions with changed settings

5 Evaluation Appendix

5.1 Learning Content Test

Spell:

- first
- Saturday
- many
- decide
- aeroplane
- sentence

No. of Syllables:

- manager
- terrifying
- sometimes
- accident
- tiger
- elephant

Spell:

- internet
- chocolate
- important

5.2 Test Results

Table 5.1 Test Results

Exp	Test 1	Test 2	Change
P1	9	9	0
P2	6	8	2
P3	6	10	4
P4	6	8	2
P5	11	14	3
P6	14	15	1
P7	7	8	1
P8	13	15	2
		Total	15

Control	Test 1	Test 2	Change
P1	9	8	-1
P2	14	14	0
P3	15	15	0
P4	13	14	1
P5	12	12	0
P6	4	4	0
P7	12	12	0
P8	10	11	1
		Total	1

Counterbalancing Groups

Test - Game	Test	Game - Test	Test
P1	12	P1	15
P2	9	P2	11
P3	7	P3	7
P4	10	P4	12
Total	38	Total	45

Table 5.2 Experimental Group – Detailed Results

		P1	P2	P3	P4	P5	P6	P7	P8	Total Correct	
Q1	T1	first	first	ferst	free	first	first	frest	first	5	
	T2	firsf	first	first	fist	first	first	first	first	6	
Q2	T1	Saturday	Saterday	Satuday	Saterday	Saturday	Saterday	Saturday	Saturday	4	
	T2	Saturday	Saturday	Saturday	Saterday	Saturday	Saturday	Saterday	Saturday	6	
Q3	T1	many	meany	meny	menay	many	many	meny	many	4	
	T2	many	meany	many	meney	many	many	meny	many	5	
Q4	T1	-	disid	diside	deside	deside	decide	disied	decide	2	
	T2	desird	diside	desid	deside	decide	decide	disied	decide	3	
Q5	T1	airplane	airerplane	airoplane	airroplane	airaplane	Aeroplane	arepaem	airoplane	1	
	T2	aeoplane	eoruplane	airoplane	aeroplane	airoplane	Aeroplane	airpam	aeroplane	3	
Q6	T1	seturs	sentes	sentens	sentons	sentecse	sentence	sentons	sentineu	1	
	T2	sentenc	sentence	sentence	sentons	sentence	sentence	satons	sentence	5	
										Total	17
										Test 2	28
										Difference	11

Q7	T1	4	3	3	3	3	3	3	3	7	
	T2	3	3	3	3	3	3	3	3	8	
Q8	T1	4	4	4	4	4	4	4	4	8	
	T2	4	4	4	4	4	4	4	4	8	
Q9	T1	2	2	2	2	2	2	2	2	8	
	T2	2	2	2	2	2	2	2	2	8	
Q10	T1	3	3	3	3	3	3	3	3	8	
	T2	3	3	3	3	3	3	3	3	8	
Q11	T1	2	1	1	2	2	2	2	2	6	
	T2	2	1	1	2	2	2	2	2	6	
Q12	T1	3	1	3	3	4	3	3	3	6	
	T2	3	1	3	3	3	3	3	3	7	
										Total	43
										Test 2	45
										Difference	2

Q13	T1	internet	internet	internet	intertne	internet	internet	iternet	internet	6	
	T2	internet	internet	internet	internet	internet	internet	internet	internet	8	
Q14	T1	choclot	chocolet	choklate	chocolatate	chocolate	chocolate	cocrelet	chocolate	3	
	T2	choclot	cholite	chocolate	chocolatate	chocolate	chocolate	corlet	chocolate	3	
Q15	T1	inport	importent	imp	-	important	important	inpretont	important	3	
	T2	inport	importent	importtant	enttament	important	important	inporte	important	3	
										Total	12
										Test 2	14
										Difference	2

6 Conclusion Appendix

6.1 Condensed Dyslexia Style Guide

General HCI:

- Does the system provide HCI redundancy but avoid multi-sensory overload. Are audio and visual clips user controlled (i.e. do not start automatically)?
- Are sequential presentations slow enough for readers with reading difficulties? Is user control provided to control speed and allow users to go back and re-read information?
- Is it possible for users to change the interface to their own needs?

Fonts:

- Use fonts that are sans serif or similar to natural writing. Unusually shaped 'novelty' fonts should be avoided.
- Does the font have clear ascenders and descenders (the stems on letters)?
- Is there clear space between letters and words?
- Use large fonts (above size 12 or 14).
- Lower case letters should be used where possible (no whole words of capitals).
- Avoid underlining (except for hyperlinks).

Layout:

- Avoid light text on a dark background.
- Avoid using white as a background colour. Pale colours and off-white should be used.
- Is text left aligned?
- Avoid continuous prose - use bullets or numbers instead.
- Use 1.5 to 2 spacing between lines.
- Avoid using moving text as this is difficult to read.
- Are elements that enhance scanning such as headings, bold text, bullets etc. used?
- Is information grouped logically? (To reduce the cognitive burden).

Writing style:

- Use short simple sentences and clear instructions. Be concise.
- Refer to readers as 'you'. Use active verbs.
- Don't hyphenate a word if it is not normally split.

Navigation:

- Navigation should be easy. On websites, a site map should be used.
- It should be possible to see which pages have been accessed.
- Use a shallow navigation system. Keep it simple. Long menu sequences are hard to remember.
- Navigation areas and buttons should clearly stand out from the text.
- Navigation should be obvious and intuitive.
- Try to make no page more than 3 links away from the homepage (or main page).
- Allow user to know where they are.