Math-Puzzle: Equation Tutor for Sighted and Visually Impaired Children

Jarno Jokinen
Alternative Access: Feelings and Games 2005
Department of Computer Sciences
FIN-33014 University of Tampere, Finland
Jarno.Jokinen@uta.fi

ABSTRACT
Teaching mathematics for blind people is a challenging task, since most of the mathematical methods require logical skills and manipulating by abstract notions that are always easy to understand visually. Math puzzle has been designed for supporting the learning of mathematical operators and equations. The simple graphical interface was augmented with short speech cues and earcons, so it can be played in a blind mode. The game allows to develop spatial imagination, memory and logics in blind or visually impaired children.

Author Keywords
Non-visual interaction, access to arithmetic equations, math-puzzle, visual impaired children, sound feedback.

ACM Classification Keywords
K.4.2 Social Issues. Assistive technologies for persons with disabilities

H5.2 User Interfaces. 1.3.6 Methodology and Techniques. Interaction Techniques

INTRODUCTION

Background

Mathematic formulas, diagrams and graphs are presented as visual models of logically connected figures, values and symbols. Visual inspection of math objects is more understandable because of employing an external memory aid that decreases cognitive loading. There have been made many studies on how to present math notions, objects and methods alternatively for blind and visually impaired people. These solutions can be divided into five categories according to feedback signals and senses used for interaction.

- Tactile games, based on pin codes (Braille, VirTouch2.com) and embossed graphics
- Haptic and force feedback tools (PHANTOM Omni)
- Non-speech audio and tonal representation
- Screen reading of the equations
- Multimodal systems (haptic & speech [Patrick Roth, 2000])

The goal of this study was to design a simple game-like non-visual interface for logic skills' training by solution of arithmetic equations. Non-visual interface might be augmented with short speech cues and earcons. In other words, what ways/techniques are possible to compose in designing tutorial for teaching basic mathematics to visually impaired children through game. Besides this, an empirical study should be done in order to estimate the limitations of the game and non-visual interface for interaction with equations.

From the beginning, it was clear that the game was going to be self-made. This decision brought a few other things into consideration. First, we wanted to know how long does it take to solve a puzzle. If the game idea is as simple as we were planning to have, game maker does not want to create a game that takes more than a few minutes to solve. It is quite crucial, for the gameplay experience, how long it takes, because the player may get frustrated if the game is not progressing. Especially when you are struggling against time.

Then we wanted to know how long does it take for the player to get better in this game. In other words, how fast the players progress through the game. Keeping the previous point in mind, this is also important, for the lifespan of the game. A game too easy to master, does not have enough challenge to be played repeatedly. On the other hand, a game too hard to learn is not rewarding enough to keep the player motivated.

Since we were going to do the testing of the game with people able to see, we wanted to find out how dramatic affect does the lack of eyesight have on the game progress. Therefore, we were going to do the tests in both visual and blind mode. In addition, it would have been interesting to see the difference between blind and sighted people.

Together with these questions, we were interested to find out what kind of strategies would the users have when solving these puzzles. And how would they get progress through the game?

MATH PUZZLE

The purpose of this project was to create a game that could be used by visually impaired children. First, the game should be easy to understand and use. Functionality was to be set to minimum so that the child starting to play the game would have no obstacles in getting familiar with the system. As the target audience is children, the game should not have too complex mathematical problems to solve. Therefore, we wanted to keep it simple and use only the most simple equations, with the possibility of addition, subtraction, multiplication and division.

With these operators the game could easily become too hard, so another constraint was to be created. All of the numbers used in the game were set to one integer long. This way the equations were easier to understand and solution would be easier to find. At this point, the appearance of the game board was pretty clear. We were developing a 5 by 5 matrix, which would include 5 simple equations in the form of x + y = z. One true equation in one row would be the way the matrix (or puzzle) would look like when it was solved.

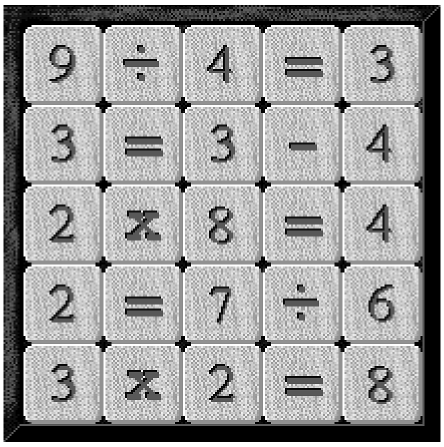


Figure 1. The equation matrix in visual mode.

GAME CONCEPT

As mentioned on the previous section, the equations are shown in a 5 by 5 matrix like in the Figure 1. First, third and fifth column include only numbers, while second and fourth are reserved for operators. At the start, all of the numbers and operators are randomized and the player's task is to change the locations of the numbers so that all of the equations are true. Randomizing all of the numbers and operators would have made the matrix more complex so simplifications were needed. We decided to restrict the randomization so that each of the columns included only either numbers or operators. Then we had to decide which way should be chosen that the equations could be solved.

Two possible solutions

At this point, we had two possible approaches on how the equations could be solved using a static or dynamic matrix. In a static matrix, the equations would be found by clicking all 5 members of the equation in a row like in Figure 2.

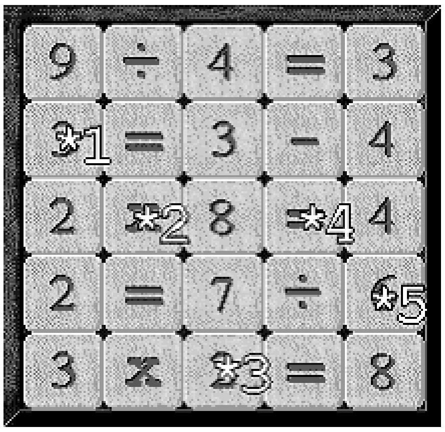


Figure 2. A static matrix approach to solve equations.

In a dynamic matrix each of the equation members are movable and the player can follow the solving progress, when all of the figures are put to their correct locations one by one. The dynamic matrix was found to be a better solution since we could not find any easy way of telling with earcons (short speech cues) to the user what equation members have already been clicked and are therefore reserved. The dynamic matrix would possibly require more memorizing, when the player needs to remember how the matrix has been changed. Even though, we decided to go with the dynamic matrix because it was easier to help the player with the imagination to built a mental model, by reading the equation members every time the user moves over a specific number or operator.

On a dynamic matrix, the squares are swapped so that in the end player would have one true equation on each row.

The idea seemed still pretty hard. The numbers and operators had their own columns already, but we decided to make it so, that the player did not have to even make the changes between different columns. This way the numbers can only be swapped inside the same column, which should make the puzzle a bit simpler to solve. Too many alternatives will increase the number of required moves and the number of moves usually tells the puzzle's difficulty level.

Implementation

The game is made using html and JavaScript. The equations used in the game are not fully randomized, since we wanted to only create a prototype of the actual game. All of the equations are written in the game source code as seen on Figure 3 and randomly selected for each game.

```
var equations = Array();
equations.push ('3x2=6');
equations.push ('4+5=9');
equations.push ('9/3=3');
equations.push ('2x4=8');
equations.push ('7-4=3');
equations.push ('8/4=2');
```

Figure 3. Array of the equations in Math-Puzzle game.

When the matrix has been created, the game can start. For solving the puzzle, the player needs help that is created with sound (short speech cues). All of the sounds are introduced prior to usage so that they are all downloaded to the PC before the game starts (Figure 4). This is crucial point because the game interface allows no delays in sound output.

```
src="0.wav" name="s0" autostart="false" hidden="true"
src="1.wav" name="s1" autostart="false" hidden="true"
src="2.wav" name="s2" autostart="false" hidden="true"
src="3.wav" name="s3" autostart="false" hidden="true"
src="4.wav" name="s4" autostart="false" hidden="true"
src="5.wav" name="s5" autostart="false" hidden="true"
src="6.wav" name="s6" autostart="false" hidden="true"
src="7.wav" name="s7" autostart="false" hidden="true"
src="8.wav" name="s8" autostart="false" hidden="true"
src="9.wav" name="s9" autostart="false" hidden="true"
```

Figure 4. The embedded speech cues.

The sounds are heard whenever the player moves over an equation member. This is done using a JavaScript event handler, called onMouseOver, shown on Figure 5. Also, the onMouseOut - event handler is used to prevent two sounds from playing at the same time.

```
img name="c11" src="s.gif" width="40" height="40"
onMouseOver="on(this);" onMouseOut="off(this);"
onClick="choose(this);"
```

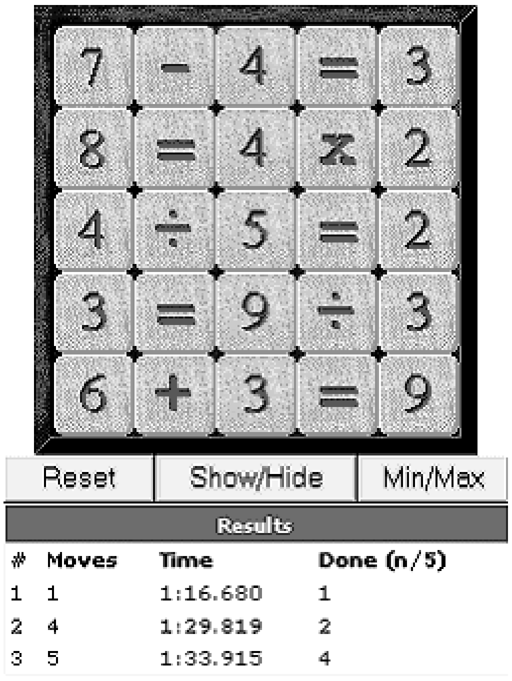
Figure 5. JavaScript MouseOver and MouseOut event handlers.

The event handlers call for functions on and off, which take care of actually playing the sounds. The idea was to use the name of the image for telling what number or operator was behind this location. This was found from the fifth last letter of the image, which was found using the substr function. If this letter is not v, meaning void, the sound is played.

```
function on(obj) {
        sound = obj.src.substr(obj.src.length-5,1);
        if (sound!='v') { document['s'+sound].play(); }
}
function off(obj) {
        sound = obj.src.substr(obj.src.length-5,1);
        if (sound!='v') { document['s'+sound].rewind();
document['s'+sound].stop(); }
}
```

Figure 6. The sound playing functions.

During the game, the player changes the location of the equation members and this way tries to get all of the equations in the matrix true. The game keeps track of how it is progressing by adding one row whenever the number of solved equations changes. This can be seen on the game screen (Figure 7), but a sound is also played telling the player how many moves has s/he used and how many equations are currently solved.



#	Moves	Time	Done (n/5)
1	1	1:16.680	1
2	4	1:29.819	2
3	5	1:33.915	4

Figure 7. The game results

We have provided some shortcuts in the game for the players' aid. They give access to the most common functions and should make the gameplay a bit more fast. For example, for visually impaired people this removes the need of mouse when starting a new game.

- Alt+R starts a new game (reset)
- Alt+M minimizes/maximizes the browser window
- Alt+S shows/hides the numbers and operators.

Game modes

As mentioned before, the game has two playing modes. In *visual* mode, the player can see all of the numbers and operators. In *blind* mode, the game board is still visual, but the numbers and operators are hidden. Both modes support sound cues for navigation.

TESTING PROCEDURE

When design of the game had reached a state in which it was possible to play, we asked five people to test it. These tests were run independently, and the results were gathered via email. The purpose of these tests was to make sure that the game works, and it is possible to play with only a few simple guidelines.

It turned out, that the rules were not clear and the instructions needed to be more detailed. Also, one of the major problems, concerning missing sound in Mozilla browsers, was found. Beta testers found some minor, mainly visual, errors which we had to fix for the final version.

After the beta testing phase, we had to fix the found errors and start to plan the actual testing sessions.

Participants

We recruited another five people (unpaid volunteers) doing the test - all were males. Their age ranged from 26 to 34 and they were all technologically experienced and knew how to use mouse. All of them had also been familiarized with the game and rules.

Apparatus

The tests were done with an IBM T40 laptop computer, 1600 MHz Intel Pentium M processor with 1024MB RAM. During testing, the participants wore earphones to listen to the sound feedbacks and to get focused in the game. We also used an external mouse, because the tools in the laptop itself are inaccurate and hard to use. Since the game is done in HTML and JavaScript, we had to use some browser for playing the game. Internet Explorer 6.02 was used because of its better compatibility with sound effects comparing to Mozilla.

The log files, created throughout the testing phase were not automatically created. They were copy/pasted from the browser window. This was something we had not thought of so much, and caused unwanted delays in the testing sessions. Though, none of the testers complaint about this.

Procedure

The tests were done in usability laboratory which provided a silent and restricted area for running the tests. Usually, these sessions would have started with an introduction, but the testers were all more or less familiar with the game, so we did not need to explain the details of the game.

Each session consisted of 30 games, 10 for each playing mode - visual, blind and blindfolded. We first started with the visual mode which was in a way rehearsing for the actual game. After this, the participants tried to solve the puzzle in blind mode, with sound as the primary guidance. After that, we asked them to put a mask over their eyes and run the last set of 10 games blindfolded, also with sound as the only help. During these games, the game board was visual, so we had a possibility to observe the player actions and evaluate their strategy.

RESULTS AND DISCUSSIONS

Rather than knowing in what time the puzzle was solved or the number of moves it took, we spent time talking with the participants.

The idea of the game was to develop basic mathematical skills and increase understanding of equations. Based on the findings during testing, the game actually evaluates an imagination of the player, especially ability in spatial imagination. The game forces the player to memorize the location of the equation members and make decisions based on his/her memory. Strong logical prowess and a strategy based on it are advantageous when solving the puzzle.

Though have been included only six equations in the source code, there are a lot of matrix variations. Some are very close to being solved and some are more complex. If game maker want to find out how good the player is in this game, he needs to run more games for reducing the meaning of luck. 10 repetitions for each game mode seemed enough for testing. The subjects were tired after the 30 games they had played.

The game includes short speech cues. However, sound cues are long enough that player really need to concentrate attention strong and listen to them. As a result, the speed was significantly slower when playing the game using only non-speech sounds.

Three players complained about the fact that they could not hear the sound when the cursor is not moving. It is heard only when the mouse is moved over the square. Sometimes they forgot what square they were on and wanted a simple way, like a mouse right click, to replay the sound again.

When following game playing of the subjects, we could find some patterns in their puzzle solving methods. Since the game requires a lot of memorizing, it is obvious that the cognitive loading has to be reduced. Completing the equations in some order (for example from top to bottom) lets player forget what rows are already completed and just concentrate on the next row. One of the participants seemed to first solve the equations that include divisions and multiplications. When asked, the tester replied that these equations include the biggest numbers, either as the result of a multiplication or as the number to be divided.

CONCLUSIONS

At this stage, the game is in many ways still, a prototype has to be tested to be efficiently improved. Nevertheless, the Math Puzzle includes all required functionality. The conclusions are based on the feedbacks received on the parts of the game that were complete.

Faced Problems

During the programming phase of the game, we faced a few problems that seemed impossible to fix. The biggest was certainly that Mozilla based browsers could not be made to produce sound. The standard ways of adding sound to web pages using JavaScript simply did not apply to Mozilla browsers.

Some other platforms and browsers had problems to support exceptionally 22kHz sound and images with 256 colors (iPAQ pocket PC). If this game is further developed, these things should be taken into consideration.

Usually, when player completed the fourth equation, the fifth gets solved too, because all of the pieces are on their correct locations. Sometimes, though, there are two possible correct equations but only the other combination can lead player to a correct puzzle. An example of this is shown on Figure 7, where four of the equations are solved, but incorrectly. Therefore, solving the fifth equation requires breaking one of the equations. In this case, the figures 6 and 3 in the last row should be swapped with 4 and 5 in the middle row accordingly.

The first tries of the blindfolded version took several minutes to complete (if they were not totally aborted due to frustration). Later, the gameplay speed increased significantly, as the players started to use the correct logic models and imagination.

REFERENCES

1. Children's math project. Available at: http://www.udel.edu/educ/cmp2/
2. Edwards, A. D. N., Stevens, R. D. and Pitt, I. J. Représentation non visuelle des mathématiques, (translated by A. Assimacopoulos) in A. B. Safran and A. Assimacopoulos (editors) Le Déficit Visuel, Éditions Masson, pp. 169–178 (1995), http://www.cs.york.ac.uk/ftpdir/pub/alistair/publications/ps/geneva.ps
3. Fitzpatrick D. Speaking Technical Documents: Using Prosody to Convey Textual and Mathematical Material, ICCHP 2002, LNCS 2398, p. 494, http://link.springer.de/link/service/series/0558/papers/2398/23980494.pdf
4. Gaura, P. REMathEx - Reader and Editor of the Mathematical Expressions for Blind Students, 2002, LNCS 2398, p. 486, http://link.springer-ny.com/link/service/series/0558/papers/2398/23980486.pdf
5. http://link.springer.de/link/service/series/0558/papers/2398/23980477.pdf
6. http://www.boowakwala.com/kids/math-game-kids.html
7. http://www.computing.dcu.ie/~dfitzpat/publications.htm
8. http://www.educational-software-directory.net/math/
9. http://www.gamealbum.com/keyword/math/
10. http://www.learn4good.com/games/kids/double_digits.htm
11. Karshmer, A.I., Gupta, G., Geiger, S., and Weaver, C.: Reading and Writing Mathematics: The MAVIS Project, BIT (Behaviour & Information Technology), January 1999.
12. Karshmer, A.I., Gupta, G., Gillan, D. Architecting an Auditory Browser for Navigating Mathematical Expressions, ICCHP 2002, LNCS 2398, p. 477
13. Lambda-project: Linear Access to Mathematic for Braille Device and Audio-synthesis http://www.lambdaproject.org/
14. Math project, http://www.cs.york.ac.uk/maths/index.html
15. Mathematical Access for Technology and Science, http://www.papenmeier.de/reha/research/mathe.htm
16. Prosody in Mathtalk http://www.cs.york.ac.uk/maths/robert/prosody.htm