AN INTERFACE USABILITY TEST FOR THE EDITOR MUSICAL

Irene K. Ficheman, Andréia R. Pereira, Diana F. Adamatti, Ivan C. A. de Oliveira, Roseli D. Lopes, Jaime S. Sichman, José R. de Almeida Amazonas, Lucia V. L. Filgueiras
Escola Politécnica - Universidade de São Paulo

Keywords: Music Composition Interface, Music Education, Usability Test, Computer Supported Collaborative Learning.

Abstract: This paper presents a usability test conducted for a music composition edutainment software called Editor Musical. The software, which offers creative virtual learning environments, has been developed in collaboration between the University of São Paulo, Laboratório de Sistemas Integráveis (LSI) da Escola Politécnica da Universidade de São Paulo (USP) and the São Paulo State Symphony Orchestra, Coordenadoria de Programas Educacionais da Orquestra Sinfônica do Estado de São Paulo (OSESP). This paper focuses on the description of a usability test applied to children between 8 and 9 years old. The goal of the test was to verify the easiness of its use and to elaborate a final report that will guide the development of new improved versions of the software.

1 INTRODUCTION

This paper presents a usability test we conducted on the Editor Musical interface, a learning environment developed for music education. The main goal of the test was to verify the Editor Musical interface and its adequacy to the target users, to check if it attends recommended usability requirements and to elaborate a report that will be used to guide the implementation of future improved versions.

The software interface is based on a simplified music notation, in opposition to standard notation so that users can experience music composition without going through the process of music literacy. The software, which has been implemented by the LSI and is part of a set of technological projects of the OSESP educational programs, offers highly interactive composition environments for individual users as well as interactive collaborative environments for groups of users interconnected through a local area network or a wide area network.

Research on technological resources applied to music education is very challenging, since this area requires attention to sound reproduction, timbre authenticity, coordination between audio and the corresponding text (musical notation or other) and the information transmission speed for real time execution and appreciation. In other areas sound tracks or sound effects are additional elements, in music education they are primary material.

A music education software interface requires special attention to the intersection between graphics and sounds. Therefore, according to Nielsen and Mayhew (Nielsen, 1993; Mayhew, 1999), it is important to check if the interface attends recommended usability requirements.

This paper is structured as follows. Section 2 presents the basic structure of usability tests. Section 3 details the necessary aspects to be considered when developing interfaces for children. In section 4 we describe the Editor Musical interface. The usability test we applied for the Editor Musical is presented in section 5 and section 6 contains our conclusions.

2 USABILITY TESTS

Even though usability tests are essential steps of software development, companies do not understand the long-term implications of not conducting these tests before proper distribution, and frequently usability is far too easily forgotten. Also, very often no funds are allocated to conduct any usability tests at all, even though they are key components to any project development. According to Mayhew (1999), the usability engineers argue that to invest in usability increases the costs of the software development industry. Nielsen (1993) defends that users do not tolerate difficult designs or slow

Romary L. (2005).
IMPLEMENTING MULTILINGUAL INFORMATION FRAMEWORK IN APPLICATIONS USING TEXTUAL DISPLAY.
In *Proceedings of the Seventh International Conference on Enterprise Information Systems*, pages 122-127
DOI: 10.5220/0002545601220127

systems (users do not want to wait), and they do not want to learn how to use them (users have to be able to grasp the functioning of the system on the fly).

Some steps are necessary to develop systems that adopt usability criteria (Mayhew, 1999; Nielsen, 1993):

1) *Planning the System*: the developer needs to understand what the system objectives are (why the system is being developed and who the users will be) and what usability objectives must be considered (efficiency, easy to remember how to use, satisfying with a minimum number of errors).

2) *Collecting Data from Users*: because the design should be based on user needs, data about these needs must be collected and developers should verify how well an existing system (if there is one) meets these needs.

3) *Developing prototypes*: it is easier for a user to react to an existing example than to theorize what would work best. Useful results can be obtained by building a prototype system, with a minimum of text content and no graphics, for a first round of usability test. The prototype can then be used to elicit users' comments and observe the prototype's ability to lead the users through the tasks they need to perform.

4) *Collecting, writing, or revising content*: based on what users need, the developer must put content into the system. As developers consider information users already have, they can think about how useful and understandable it is. Most people want to quickly scan information and read only small sections. If the information is organized in long paragraphs, it definitely needs revising and should be broken into small chunks with many headings. Unnecessary words must be cut out. Lists and tables help people find information quickly.

5) *Conducting usability tests*: usability tests are an iterative process. The goal of usability tests is to ascertain what helps users accomplish their tasks and what prevents them from completing their tasks. Using the prototype as a starting point, the usability testers build a set of scenario tasks they will ask users to execute. As detailed information about user success is gathered and reported, the prototype can be modified and additional aspects tested.

The focus of a usability test is the user's experience with a system. During a usability test, specialists working with the designers and system developers watch users working through tasks with the system and gather users' feedback. The purpose is always to see what is working well and what is not working well – keeping in mind the main goal, which is to improve the system. Usability specialists plan the test, work directly with the users, and take notes; designers, developers, and others also observe and take notes. The result of usability testing is a report that includes a set of recommendations to improve the system.

3 WORKING WITH CHILDREN

Educational software should trigger children's curiosity, guiding and stimulating them to seek knowledge. It should create environments where children can develop initiative and self-confidence, as well as language, thought and concentration. The interface of educational software programs should be simple, intuitive and interactive, providing learning while playing environments.

An educational software developing team should be formed by programmers and educators, which must take part in the conception, specification and development of the software. The composition of this team is extremely important, so that technical and pedagogical aspects can be considered and good learning tools can be developed. However, the team must also be concerned with the final users, mostly when these are children.

As any other user-interface design process, educational software projects should start with the analysis of the user profile and the tasks the user will need to perform. It is practically impossible to design for children of all ages (Shneiderman, 2000).

According to Druin (2002), children can take part in the software development process in four different ways: as technology users, as testers, as informers and as project colleagues. The most common way children participate is as technology users. In this case their role is to use the software while the development team observes them so they can understand their behavior and the learning experience they demonstrate. These observations can be used to guide future projects.

Children, in the role of testers, use the prototypes during the software development process, so that the development team can correct technical and pedagogical inconsistencies found by the users. The development team can observe children using existing products before the beginning of the development process, which can start on the basis of these observations. In this case children play the role of informers and can participate further during the development process. The role of children as project colleagues is very similar to the role of informers, but as colleagues they also participate in the research and decision-making during the whole development process.

In our project we chose to include children in the role of testers. We involved children of the target age group and with their help we tested prototypes of the software program during the development process. The team studied all the observations made

during these tests and implemented changes to improve the final product. Children observation helps the development team discover positive aspects and negative aspects of the educational software interface. This is an opportunity to correct mistakes and to identify and implement changes and improvements.

Children will not normally write down their impressions at the end of the test, therefore it is essential to observe the users and pay attention to any comment they make while using the software program and to conduct a semi structured interview during which they can express their opinion verbally. The children are neither programmers nor engineers, but are experts in knowing what they want, and in being children (Guha *et.al,* 2004). Therefore, it is important to conduct usability tests with children during the development of an educational software program.

4 THE EDITOR MUSICAL SOFTWARE

Computational power and especially multimedia resources enabled the development of educational software programs with which abstract concepts can be graphically represented and users can control the interaction rhythm, can manipulate and construct their knowledge by the means of experience. The way concepts are presented to the users becomes more important than the concepts themselves (Druin, 1999).

Figure 1: Editor Musical Operation Modes

According to Lopes and Krüger (2001), developing educational software programs that stimulate students' creativity and innovation capacity, is extremely important. These programs increase student potentials and productivity supporting them in the learning process.

The C(L)A(S)P Model developed by Swanwick (1979) identifies five parameters of musical experience, five ways in which people relate to music: Composition, Literature studies, Audition (or Audience—listening), Skill acquisition and Performance. According to the author, Composition, Audition and Performance are primary activities since they allow a direct involvement with the music and, Skill acquisition and Literature studies provide knowledge about music. Composition includes all forms of musical invention, and is the best way to acquire musical knowledge since the individual is able to make decisions and transform the created musical object (Hentschke, 2004). The Editor Musical is an edutainment software inspired by the constructivism learning theory that allows the students to actively participate in the learning process manipulating timbre, sound and rhythm in highly interactive environments where they can experience music composition.

4.1 Description

The Editor Musical is a music composition software which includes different interactive environments or operation modes. These include composition modes in which the students can use their creativity and experiment composing melodies in a wide range of musical styles – like homophonic (a single melody), polyphonic and harmonic, tonal and atonal music, of different historical periods and musical cultures. Challenge modes present composition suggestions previously prepared by the teachers. Its main feature is that it uses a simplified music notation, as opposed to standard notation. Therefore, users can experience music composition without going through the process of music literacy.

Figure 1 shows the different operation modes supported by the software and they are: individual composition, individual challenge, collaborative composition and collaborative challenge. The individual composition mode is detailed in section 4.2 below since we conducted a usability test (presented in section 5) regarding only this mode. The individual challenge mode allows users to solve composition challenges prepared by a teacher, who can develop his/her own challenges using the Challenge Editor, an additional software developed to support teachers class preparation tasks. The collaborative modes support collaborative learning. Users interact in virtual classrooms to compose together a melody (collaborative composition mode) or to collaborate and solve a challenge (collaborative challenge mode) (Ficheman, 2002).

4.2 Individual Composition Interface

This interface allows the student to interact with the software and freely experiment music composition in a direct contact with the music. It includes a tool bar on the left side of the screen, a grid on the right side and a command menu bar on the top. Users can experiment making music with up to three musical

instruments, can listen to the composed melody, appreciate and edit the result.

Instruments are represented by colors and the user composes a melody by painting the grid, which represents the staff. This graphical notation allows non-music literate users to interact with sounds of musical instruments and experiment making music without neither having previous knowledge of the standard notation nor learning to play a real instrument. The note names can optionally be shown or hidden.

The main feature of this interface is that it is intuitive and highly interactive. When a note is added to the grid, the computer automatically reproduces the sound of the chosen instrument playing that note. Long notes are represented by long bars and short notes by short bars. When the 'play' tool is activated the software plays the composed melody highlighting the notes that are being played at the time they are played.

5 THE USABILITY TEST

When developing software programs for children, it is very important to involve some children of the target age group during the development process (Druin, 1999). We accomplished a practical usability test with 8 children, working in couples in each computer and for each couple there was one observer. We chose to run the test with 8 children because our research group is small and we wanted one observer for each computer and couple of children. The observer's tasks were to guide the children, to time how long the activities took and to write down additional information about the way the children interacted with the interface. Another observer took photographs with a digital camera for future analyzes.

5.1 Usability Criteria

Before developing the usability test, we chose the usability criteria we considered most relevant to the application we were testing. It is important to define usability criteria, because they help focus the attention and resources on the user and their related issues. Also, usability criteria challenge the design team to innovate and provide the basis for design tradeoffs. In addition to guiding the design, usability criteria are useful in customer interactions, in evaluation, and in testing. The usability criteria considered for the Editor Musical were:

- to develop fun interfaces: software should create a learning by playing environment;
- to develop intuitive and easy to learn interfaces: the interface should be simple enough and it should be easy for the user to associate commands and icon symbols to real life activities.;
- to stimulate children's creativity: children should be able to express themselves using the software and should be able to be the authors or creators of new objects (text, drawing or sound, for example);
- to use simple vocabulary in Human-Computer dialogs and maintain nomenclature consistency: children of the target age group should understand the dialogs that should be, in our case, in Portuguese and communication should not use technical words. The same object or operation should be always named the same way for nomenclature consistency.

This way, the main objective of this test was to determine the software's adequacy to the usability criteria described above.

5.2 Usability Test Organization

The usability test was organized as follows:

1. Presentation of the team and the children: so everybody felt comfortable during the test;
2. Explanation that the goal was to test the software and not the children, to avoid pressure and frustration;
3. Explanation that some activities needed to be executed and that the observer would indicate what the activity was, but could not explain how it was done. Children would have to discover how to do so.
4. Pre-test Questionnaire application (shown in Table 1): to identify the user's profile.
5. Usability test application (activities shown in Table 2).
6. Pos-test Questionnaire application (shown in Table 3): to verify what the users thought of the system.
7. Semi-structured interview with the group of children: at the end of the test we asked the children to give their opinion on the software and the test. We interviewed them together to emphasize that the questions we asked were addressed to the group and this way they felt free to interact with each other.

Table 1: Pre-test Questionnaire

1. Do you play a musical instrument?
2. What grade are you in?
3. How old are you?
4. How often do you use a computer?
5. Have you already used a computer program to learn music?
6. What do you expect from the computer program you will use now?

Table 2: Usability test activities list

1. Run the EDITOR MUSICAL.
2. Go to the Individual Composition mode.
3. Compose a music:
4. Choose the Piano instrument.
5. Make a music.
6. Play this music.
7. Choose another instrument and continue your composition with two instruments.
8. Play this music.
9. Continue this music with the first instrument only.
10. Play this music
11. Change the Piano to the Guitar.
12. Play this music.
13. Save the music and name it.
14. Erase the Guitar part.
15. Play this music.
16. Select part of this music and put it in another place of the screen.
17. Play this music.
18. Use 'the metronome' to modify the music speed.
19. Play this music.
20. Save the music again.

Table 3: Pos-test Questionnaire

1. Did you like to use this computer program?
2. Is the computer program similar to what you imagined?
3. Would you use this computer program again?
4. Do you think this computer program is easy to use?
5. Do you think this computer program is fun?

5.3 Usability Test Analyses

The objective of the pre-test questionnaire was to identify the children profiles (see Table 1). The answers of the pre-test questionnaire were:

- Question 1: 100% of the children did not play any musical instrument;
- Question 2: 75% of the children studied in 3rd grade and 25% of the children studied in 2nd grade;
- Question 3: 62% of the children were 9 years old and 38% were 8 and 10 years old;
- Question 4: 87% of the children used a computer once every two weeks and 13% used a computer once a week;
- Question 5: 62% never used a computer program to learn music 38% used such a program once;
- Question 6: this question was open and we were mostly surprised to find out that the children expected the computer program to be fun. Some answers were: "Fun music", "Music and many fun things", "I hope the software is cool and fun".

According to the children's answers in the pre-test questionnaire, we can say that most children studied in 3rd grade, were 9 years old, used a computer once every two weeks and never used a computer program to learn music. The last question was very important to identify the children motivation to use the software, and we can deduce that they were very motivated. We should remember that these children come from low-income families, and this explains why they have little access to technology in general and computers in particular. These children use computers mainly at school. We developed this software program to be used specifically in public schools where children usually come from low-income families. Therefore we can deduce that the children involved in the usability test were representatives of the target users group.

Table 2 details the activities the children executed during the usability test. We analysed the observers' notes as well as the photographs and could identify the following usability problems:

- children do not perceive intuitively that they can compose with more than one music instrument;
- it is not clear when instruments are active or not, and some commands are only executed on active instruments;
- children are not used to common standard commands like: copy, paste, cut, select and save;
- the icons "play" and "erase" were very intuitive.

The pos-test questionnaire helped us understand what the children thought about the software (see Table 3). The answers of the pos-test questionnaire were:

- Question 1: 87% of the children liked the software very much and 13% answered that they liked it a little;
- Question 2: 74% of the children thought the software was very similar to what they expected and 26% thought it was similar;
- Question 3: 100% of the children would use the software again;
- Question 4: 87% of the children thought the software was very easy to use;
- Question 5: 100% of children thought the software was fun.

The main objective of the pos-test questionnaire was to evaluate what the children's opinions about the software were. All the pos-test questionnaire answers were very positive, because the children enjoyed themselves when they used the software. We believe that the test is adequate to children, and they did not feel pressured by it.

All the activities developed during the usability test helped us verify if our usability criteria were satisfied. Of course, some interface problems were discovered when the children used the software, especially an important one we did not consider beforehand: when the children were asked to change the instrument, they did not know which instrument was selected, and only after playing the music, they

discovered that they had chosen the wrong instrument. We concluded that the interface must be changed to reflect and show visually the instruments that were chosen. However, we could observe that they enjoyed themselves and understood most of the icons, menus and buttons.

The fact that each instrument is associated to a colour made intuitive that changing the instrument will change the colour of the notes and therefore the resulting sound, although it is not obvious which instrument is chosen just by looking at the interface. The play icon that was used during the usability test in the activities number 6, 8, 10, 12, 15, 17 and 19, as well as the erase icon used in the activity number 14, were very intuitive, since the children identified automatically the icons and their corresponding function.

Also, software analyses allowed us to identify some usability criteria that were not satisfied like for example, the goal to use simple vocabulary in dialog boxes that should be in Portuguese. We used program screenshots to illustrate the problem as shown in Figure 2.

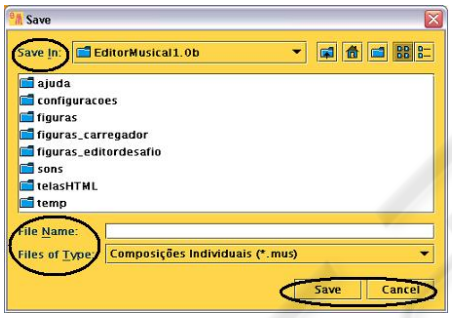

Figure 2: Usability Criteria not satisfied: dialogs in Portuguese

After analyzing the questionnaires, the notes and the observations, we compiled the data in a spreadsheet and elaborated a final report that included statistical information about the test as described above, software interface analyses and screenshots, as well as recommendations related to the software interface. This report has already been used to correct some software usability problems like command names inconsistency and dialog boxes that communicated with the user in English and not in Portuguese.

6 CONCLUSIONS

The interface of an educational software must be simple, intuitive and interactive, providing learning while playing environments. Involving children in the design process and in usability tests may be the key to success and certainly guaranties the development of more adequate interfaces.

We have presented a usability test we conducted on the Editor Musical, a new interface for music composition that can be used for music education supported by computers for individual interactions or for group learning in collaborative virtual environments.

The usability test helped us identify some interface problems that will be corrected in the new version of the Editor Musical. Usability tests are essential steps in any system development, especially when working with children, since the result of these tests can guide the development so that the system will be adapted to the users. An important aspect when usability tests are conducted with children is that they must feel confortable and enjoy themselves, because their reactions are indications about the software interface usability.

REFERENCES

Druin, A. The Design of Children's Technology. Morgan Kaufmann Publishers, 1999.

Druin, A. The Role of Children in the Design of New Technology. Behaviour and Information Technology, 21(1) 1-25, 2002.

Ficheman, I.K.; Lopes, R.D.; Krüger, S.E. A Virtual Collaborative Learning Environment. SIACG - Simpósio Ibero-American de Computação Gráfica. I. Julho 2002. Portugal.

Guha, M. L., Druin, A., Chipman, G., Fails, J., Simms, S., & Farber, A. Mixing Ideas: A new technique for working with young children as design partners. In Proceedings of Interaction Design and Children (IDC'2004). College Park, MD, 2004, pp. 35-42.

Hentschke, L., Martínez, I. Mapping music education research in Brazil and Argentina: the britsh impact. Psychology Of Music. Grã - Bretanha: , v.32, n.3, p.357 - 359, 2004.

Lopes, R.D.; Krüger, S.E. O Estímulo à Criatividade e as Novas Tecnologias. IV Congresso Arte e Ciência – Mito e Razão. Centro Mario Schenberg, São Paulo. 2001. p.188-194.

Mayhew, D. J. The usability engineering lifecycle: a practitioner's handbook for user interface design. San Francisco, Calif. : Morgan Kaufmann Publishers, 1999.

Nielsen, J. Usability Engineering. Academic Press, 1993.

Shneiderman, B. Universal Usability: Pushing Human-Computer Interaction Research to Empower Every Citizen. Communication of the ACM, vol. 43, n. 5, 2000, p. 84-91.

Swanwick, K. A Basis for Music Education. London: Routledge, 1979.