

AudioBattleShip: blind learners cognition through sound

J H Sánchez

Department of Computer Science, University of Chile
Blanco Encalada 2120, Santiago, CHILE

jsanchez@dcc.uchile.cl

ABSTRACT

Recent literature provides initial evidence that sound can be used for cognitive development purposes in blind children. In this paper we present the design, development, and usability testing of AudioBattleShip, a sound-based interactive environment for blind children. AudioBattleShip is an interactive version of the board Battleship game, providing different interfaces for both sighted and blind people. The interface is based on spatialized sound as a way of navigating and exploring through the environment. The application was developed upon a framework that supports the development of distributed heterogeneous applications by synchronizing only some common objects, thus allowing the easy development of interactive applications with very different interfaces. AudioBattleship was tested for cognitive tasks with blind children, evidencing that it can help to develop and rehearse abstract memory through spatial reference, spatial abstraction through concrete representations, haptic perception through constructing mental images of the virtual space, and cognitive integration of both spatial and haptic references.

1. INTRODUCTION

Diverse audio-based applications for blind children have been implemented in the last few years (Baldis, 2001; McCrindle and Symons, 2000; Mereu and Kazman, 1996; Sánchez, 1999, 2001). Many of them focus on the development of 3D audio interfaces to map the entire surrounding space as proof of concept concerning audio-based interfaces. Very few studies centre on the effects and impact of sound on developing cognitive skills of blind people by evaluating the cognitive usability of these applications. No previous work has centred on using spatialized sound to develop abstract memory, haptic perception, and the integration of spatial and haptic references by blind learners. We also have no reference of previous work for engaging blind and sighted people in interactive and collaborative games.

Developing interactive systems for integrating people with different capabilities poses a higher challenge because of the synchronization of different interfaces for each type of user. Interfaces for sighted users normally consist of computer graphical user displays and some sound for the output. The keyboard and mouse are the most common devices for input. Interfaces for blind users rely principally on auditory information for output, keyboards, and other haptic devices for input (Tan, 2000; Sjostrom, 2001). Sometimes joysticks are used for both input (movement) and output, as they pose some resistance in order to simulate the contour objects.

This research study presents the design, development, and usability testing of AudioBattleShip, a sound-based interactive environment for blind children (Sánchez et al., 2003). This system is a similar version of the traditional battleship game for sighted people but including both graphical interfaces for sighted users and audio-based interfaces for blind people. AudioBattleShip provides a gaming environment to enhance group interaction, abstract memory, spatial abstraction, and haptic perception in blind learners. We also introduce a platform which supports programming of distributed applications, especially for synchronizing applications having different interfaces.

For designing and developing information-equivalent interfaces (Veijalainen and Gross, 2001) for sighted and blind people we followed a similar process such as the one described in (Baloian, Luther & Sánchez, 2002) for developing interfaces for people with disabilities.

Finally, a full usability study has been implemented to evaluate the cognitive impact when blind children interact with AudioBattleShip.

2. THE AUDIOBATTLESHIP DESIGN

2.1 *Playing modes and phases*

As the aim of this game is to engage blind people in a interactive environment to help to develop cognitive skills we decided to provide various playing modes. AudioBattleShip can be played blind to blind, blind to sighted, and blind to computer. Blind to blind mode presents the same interfaces to both players. Blind to sighted mode provides a variety of tools to the blind learner to minimize the disadvantages compared to sighted learners that can have in any moment snapshots of the state of actions. Blind to computer mode gives intrinsic advantages to the computer over the blind learner because of computer memory and probability calculations, but this disadvantage can be diminished by limiting the algorithm and creating different levels of complexity.

AudioBattleShip has three phases (see Figure 1):

Ship positioning phase: the player chooses the position of ships taken from a predefined set on the battlefield. A matrix represents a battlefield where ships can be placed over a column and a row by covering different numbers of spaces according to the type of ship.

Creating and joining session phase: After placing the ships, a player can choose between creating a new session of the game, joining an existing one, and playing against the computer. If a new session is created, another player can join it. To join an existing session in the Internet the player has to know the session's name and the host address.

Shooting phase: By taking turns both players can manipulate the matrix with the board's state by identifying a cell within the matrix representing the contender's battlefield for dropping a bomb on that place. The system responds whether a contender's ship was hit or not.



Figure 1. *Phases of the AudioBattleShip game*

2.2 *Interface design*

Baloian, Luther & Sánchez (2002) propose a model for developing educational software for people with different disabilities by using diverse interfaces. This model can be briefly summarized in the following steps: 1. To develop the computational model of the real world; 2. To model the input/output needs for the system (including feedback); and 3. To develop interfaces by “projecting” the model over a range of user's interaction capabilities.

In this case, we modelled AudioBattleShip by mapping the two players, each with a matrix representing the space where their own ships are positioned. A player's matrix will also register the information about where the contender has dropped a bomb. A token represents which player is in turn for dropping a bomb in the contender's field.

We created two different applications, one for blind and one for sighted users. Each application has two interfaces, one for the ship positioning phase and the other for the shooting phase. The ship positioning interface for sighted people consists of a window containing a grid representing the battlefield. Ship positioning takes place by defining a position and a direction (up, down, left, right) with the mouse. The shooting phase interface consists of a window displaying two matrices, one for the battlefield where the player's own ships are positioned and the other representing the contender's field, showing the places where the player has already dropped bombs and the outcome of this operation. A text area was provided to inform about the course of the game.

The application for blind users uses a tablet as input device (see Figure 2a). The tablet can map the entire screen and by using a pen-based pointing device diverse mouse events can be triggered. A grid was built over

the tablet in order to represent the matrix of the battlefield and some additional “help buttons” for triggering actions. During the ship positioning phase the player has to point to a certain place on the grid and indicate the direction by moving the pointing stick to a cell up, down, left, or right. During the shooting phase the player has just to double-click over a certain cell of the grid (see Figure 2b). Sound feedback is provided to inform about a specific spatial location on the board and the occurrence of certain actions, for example, the resulting outcome of dropping a bomb in a cell of the contender’s battlefield. Help buttons assist the player in remembering the places where an enemy’s ship was hit or not.

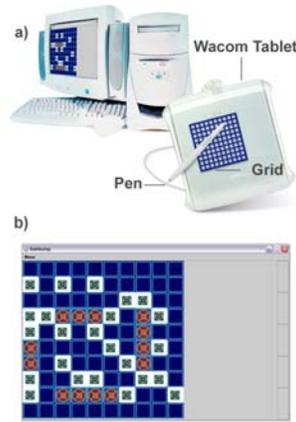


Figure 2. *Input devices (a) and shooting phase (b) for blind players*

3. A PLATFORM FOR REPLICATED COLLABORATIVE APPLICATIONS

Building collaborative software, a developer has to decide a mechanism for making a distributed application. Basically, the developer has to decide between two paradigms. One of them is to select a framework with a centralized server who is capable of serving several clients. In this case, the information needed by all clients is first sent to the server and then somehow broadcasted to the clients. In these scenarios, e.g. NetMeeting (www.microsoft.com/windows/netmeeting), it is not possible to work with the clients any longer if the server is for some reason not accessible any more. The second paradigm is a replicated architecture. Here all the necessary data for clients is replicated in the client applications. The advantage of this kind of framework is that clients are still operable even if they lose their network connection.

The MatchMaker TNG framework (Trewissen et al., 2000), which is based on RMI, combines both paradigms for making collaborative software. First of all it has a centralized server, which manages to send the needed data to the clients, and it also replicates the whole internal data structure of the application. By replicating the data structure it is possible for each application to have a comparative synchronized status. Within this server application the data structure is arranged as a tree called the “synchronization tree”. It is not necessary for every client to listen to the whole tree but clients may also listen to certain sub-trees. Most likely, the synchronization tree will reflect the internal data structure of the application as shown in Figure 3.

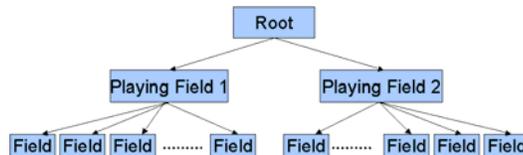


Figure 3. *The synchronization tree for AudioBattleShip*

Several frameworks for synchronization use event-based technologies (Jansen et al., 2001), but MatchMaker TNG replicates according to the Model-View-Controller paradigm, by implementing a so-called MatchMaker TNG model for every object to be replicated. This MatchMaker TNG model consists of the object model but may also have information about the view. By using these models the possibility of interpreting them differently in diverse applications is high. For example a model may be fully interpreted by an application that is running on a PC with high performance, but some pieces of information may not be considered if the same model is interpreted by an application running on a PDA.

Another example is the use of the model in AudoBattleShip. Here it is possible to integrate in the model information needed for people with disabilities. This could be ignored by an application that is running for people without disabilities.

The models are propagated to the clients using the common event architecture of Java. The server has four basic operations to inform the clients about changes in the synchronization tree when a new node is created, deleted, changed, or activated. Every MatchMaker TNG client has to implement a special method for each of these four operations.

3.1 *The implementation of AudioBattleShip*

Since the interfaces of both applications are very different the most convenient way to synchronize them is at the level of the data model. AudioBattleShip has objects that store the information about players, the state of the game (turns), and the coordinates of a given point with the corresponding state (ship, water, destroyed ship, shooting to water). By using these objects a synchronization tree is created with nodes that store information about players that also contains objects corresponding with the size of the board.

According to this, the MatchMaker synchronizing tree has the turn-taking token at the root of the tree. The information about the two players is stored as two child nodes of the root. Each cell representing the battlefield of a player is stored as a separate child node from the player's node (see Figure 3). This has the advantage that only the information of one cell will be transmitted to the coupled application if just the information changes, and not the whole matrix. Synchronizing applications using sound have also an additional problem: the sound playing is usually implemented as a separate thread in most programming languages (and Java is not the exception). In many cases, we must be sure that some auditory information has been completely heard by the user before continuing with the program execution (which may mean the playing of another sound clip). To accomplish this, Java Media Framework was used to synchronize multimedia events. AudioBattleShip contains a directory of sounds that can be easily modified to change language and sounds.



Figure 4. *On site interaction with AudioBattleShip between two blind learners*

4. USABILITY TESTING

AudioBattleShip was usability tested both during and after software implementation. Three usability experiences were designed. Two of them were centred on the usability of the interface provided and the last testing was concerned with the cognitive impact of using AudioBattleShip.

4.1 *Interface usability*

Interfaces of AudioBattleShip were evaluated by blind learners. The first experience consisted of testing different software modules with four blind learners that regularly attend the “Santa Lucia” School in Santiago, Chile. After finishing a software module, it was tested by learners. They interacted with the software, made comments, and answered questions. Thinking aloud methods were also implemented. All of these procedures helped designers to redesign and make it closer to the mental model of the blind children.

The first prototypes were fully tested. The interface design and functionality were analyzed. Four blind learners from the same school participated in the study. They played the game and answered questions during four sessions. Two people from the research team observed them, asked questions, and registered their comments and behaviours in all sessions. Most questions were related to audio interfaces and the mapping and use of input devices such as the tablet. We tested how well the concrete matrix of the traditional board Battleship game was mapped into the interactive environment. The mapping of the screen matrix by the

tablet with a rough grid on it using a synthetic material was also evaluated. We then tested the usability of some interface elements such as sounds (intensity and quality), volume, and feedback to the user actions.

The tasks followed by learners were the following: 1. To explore AudioBattleShip, 2. To understand the structure, rules, and adaptations made, and 3. To understand the differences between the concrete Battleship board and the digital version of the game. Users also learned how to play the game by using both the keyboard and tablet. They fully explored the pros and cons of using a tablet, its functions and analogies with the AudioBattleShip interface. We looked for audio stimuli association to game actions and the understanding of cause-effect action relationships when interacting with the software (see Figure 5).

As a result of this preliminary testing, learners enjoyed interacting with AudioBattleShip. They helped to redesign the software by making insightful and meaningful contributions about sound synchronicity, sound overlapping, sound help, colour contrast, size of the cursor, position identification, and tablet mapping.



Figure 5. *Interface usability testing by blind learners*

4.2 Cognitive impact

We tested the cognitive impact of AudioBattleShip with five learners during six sessions of ninety minutes each. All sessions were captured and registered using a video digital camera. The comments made and answers to specific questions were also registered. Two special education teachers observed blind learners doing tasks with AudioBattleShip using the tablets and filled out observation forms specifically designed for this purpose. We developed a paper form for each cognitive task containing several questions, an answer scale, and open observations.

The sequence of actions followed by learners included: 1. To navigate AudioBattleShip, 2. To identify different audio stimuli, 3. To explore both the concrete board of the traditional Battleship game and the Wacon tablet, 4. To play AudioBattleShip, and finally, 5. To make concrete representation models (clay and Lego) of navigating spaces and position moves (see Figure 7). They interacted with their partners and then represented in a concrete board the moves of the adversary. One player described the coordinates and others localized the pieces and marked them on the board. They also searched the ships of the adversary by using a strategy previously designed (see Figures 4 and 6).

We looked for the impact of using AudioBattleShip on abstract memory through spatial references with sound, spatial abstraction through the concrete representation of the virtual environment, and haptic perception (tactile and kinaesthetic) by generating mental images of the navigated space. They also integrated spatial references through sound and haptic references through the tactile manipulation to construct mental images of the navigated virtual environment.



Figure 6. *Evaluating the cognitive impact of using AudioBattleShip*



Figure 7. Concrete representation models of the navigating space and position moves

Task one was concerned with integrating spatial references through sound and haptic references through tactile manipulation to construct mental images of the navigated virtual environment. We also wanted to check whether or not by using AudioBattleShip learners can perceive the virtual space through haptic perception by generating mental representations. Most learners could clearly differentiate each audio stimulus. Most of them could make an excellent analogy between AudioBattleShip and the concrete board of the traditional Battleship game (80%). The same percentage of learners could make an analogy between the Wacon tablet and the virtual environment of the adversary. All learners did not need a concrete board to orientate their movements and navigation throughout AudioBattleShip (see Figure 8). Some learners performed excellent against an adversary (40%) and others did a good performance (60%). The same result was obtained when played against the computer. This means that learners could design a strategy to solve problems and defeat the adversary.

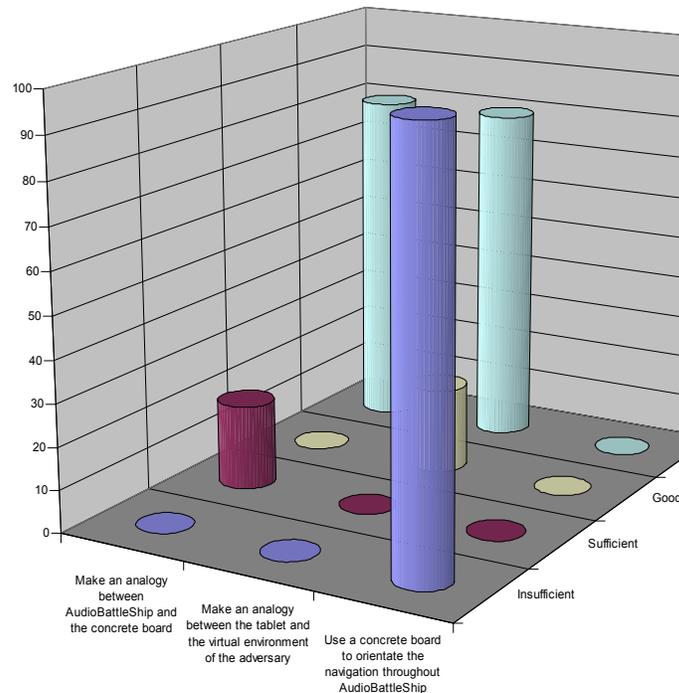


Figure 8. Mental images and haptic perception through using AudioBattleShip

Task two measured abstract memory through spatial references with sound and spatial abstraction through the concrete representation of the virtual environment. Almost 90% of learners could completely represent in a concrete board the moves of the adversary. None of them needed to use a concrete board to orientate their navigation throughout AudioBattleShip (see Figure 9).

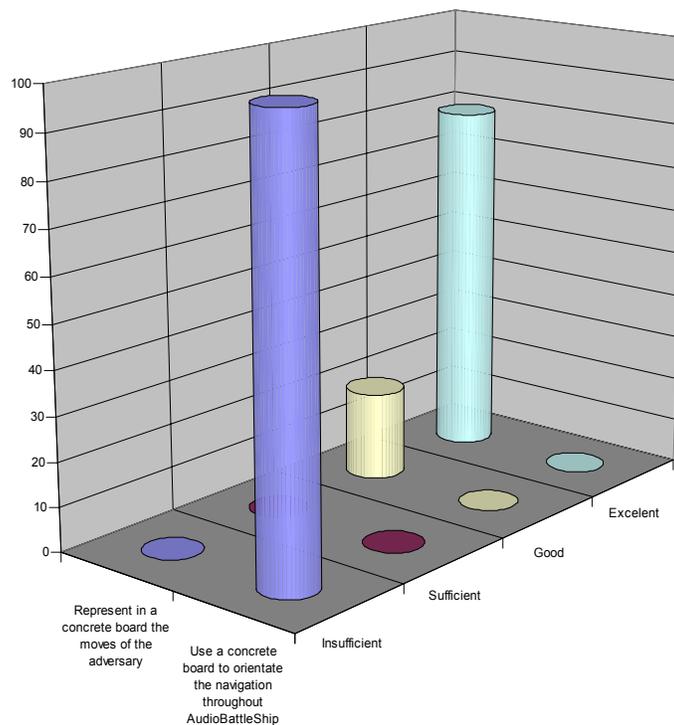


Figure 9. *Abstract memory and spatial abstraction through AudioBattleShip*

Task two also included a specific activity related to detect how well learners could develop a strategy to overcome the adversary. Most of them could design and implement a strategy to win the game (80%). All of them could orientate easily within the matrix implemented on the Wacon tablet.

5. FINAL DISCUSSION

This study was aimed at designing and evaluating a sound-based virtual environment to develop and rehearse abstract memory through spatial reference, spatial abstraction through concrete representations, haptic perception through constructing mental images of the virtual space, and cognitive integration of both spatial and haptic references. Our data indicate that blind learners using AudioBattleShip mediated with cognitive tasks and concrete materials can help to develop and exercise mental images of the space, haptic perception, abstract memory, and spatial abstraction.

There were diverse forms of interaction among blind learners. Some collaborative abilities can be enhanced by using interactive sound-based applications such as AudioBattleShip. We need to know in more detail what type of skills can be stimulated and ways of improving them through the interaction with sound. This is especially important for blind learners since they are accustomed to doing individual work with little social interaction when using digital devices. We envision constructing new ways and scenarios for collaboration between blinds as well as between blinds and sighted learners by using digital audio interfaces such as the one introduced here.

Learners enjoyed the interaction with AudioBattleShip. This can be explained because this interaction allowed them to compete and measure their competences and abilities. This experience also had an affective side. They did improve their self-esteem and self-value to compete defeat other learners with and without vision. Learners also showed more interest to play with a blind partner instead of playing against the computer. They think interacting with computers did not challenge and give the same feedback in comparison to playing with a peer. They were motivated to play and interact with sighted learners because it was a way to show they can make the same performance as sighted if minimal conditions are met at the beginning.

The use of Wacon tablets as interaction devices for AudioBattleShip was a critical strength. They allowed blind learners to interact with different interface haptic devices and demonstrated to be highly appropriate for this type of software.

A recent retest study (Sánchez, Baloian, and Hassler, 2004) is giving us more detailed insights and knowledge about how cognition and collaboration can be enhanced by the interaction with sound. We are contrasting this quantitative data to evaluate more fully the real contribution of audio-based interfaces in blind children cognition and collaboration.

Finally, the observations made by learners, their comments, and meaningful ideas were critical to improve AudioBattleShip. This validates one of our premises in a sense that interactive software environments based on sound should be designed with and for blind learners in order to map their needs, interests, and mental models.

Acknowledgment: This report was funded by the Chilean National Fund of Science and Technology, Fondecyt, Project 1030158.

6. REFERENCES

- C Sjostrom (2001), Using haptics in computer interfaces for blind people. *Proceeding of the ACM CHI '01*, Vol 3, 1, pp. 245-246.
- F Tewissen, N Baloian, H Hoppe, and E Reimberg (2000), "MatchMaker" – Synchronising objects in replicated software architectures. In *Proceedings of CRIWG 2000 (Sixth International Workshop on Groupware)*, Madeira, Portugal, October 2000, Los Alamitos: IEEE Press. pp. 60-67.
- H Tan, (2000), Haptic interfaces. *Communications of the ACM*, 43(3), pp. 40-41.
- J Baldis (2001), Effects of spatial audio on memory, comprehension, and preference during desktop conferences. *Proceeding of the ACM CHI '01*, Vol 3, 1, pp. 166-173.
- J Sánchez (1999), Interactive 3D sound hyperstories for blind children. *Proceedings of ACM-CHI 99*, pp. 318-325.
- J Sánchez (2001), Interactive virtual acoustic environments for blind children. *Proceedings of ACM CHI '2001*, Seattle, Washington, pp. 23-25.
- J Sánchez, N Baloian, & T Hassler (2004). Blind to Sighted Children Interaction through Collaborative Environments, X International Workshop on Groupware, CRIWG 2004, September 5 al 9, San Carlos - Costa Rica (will be also published in an issue of *Lecture Notes in Computer Science, LNCS*, Springer-Verlag).
- J Sánchez, N Baloian, T, Hassler, and U Hoppe (2003). AudioBattleShip: Blind Learners Collaboration Through Sound. *Proceedings of ACM CHI 2003*, pp. 798-799. Also published by *SIGCHI publications, ACM Press*. Fort Lauderdale, Florida, April 5-10.
- J Veijalainen and T Gross (2001), Mobile Wireless Interfaces; in Search for the Limits. In: *Developing an Infrastructure for Mobile and Wireless Systems*; König-Ries, B., Makki, K., Makki, S.A.M., Pissinou, N., Scheuermann, P., (Eds.) NSF Workshop IMWS 2001, Scottsdale, AZ, October 15, 2001, Revised Papers. Springer Verlag LNCS Vol. 2538, Dec. 2002, pp. 153-163.
- M Jansen, N Pinkwart and F Tewissen (2001), MatchMaker - Flexible Synchronisation von Java-Anwendungen. In Klinkenberg, R., Rüping, S., Fick, A., Henze, N., Herzog, C., Molitor, R. & Schröder, O. (eds.): *LLWA 01 - Tagungsband der GI-Workshopwoche "Lernen-Lehren-Wissen-Adaptivität"*, October 2001. Forschungsbericht 763. Universität Dortmund, Germany.
- N Baloian, W Luther and J Sánchez (2002), Modeling Educational Software for People with Disabilities: Theory and Practice. *Proceeding of the ASSETS 2002 Conference*, 8 - 10 July 2002, Edinburgh, pp. 111-117.
- R McCrindle and D Symons (2000), Audio space invaders. *Proceedings of the Third International Conference on Disability, Virtual Reality and Associated Technologies*, pp. 59-65.
- S Mereu and R Kazman (1996), Audio enhanced 3D interfaces for visually impaired users. *Proceedings of CHI '96, ACM Press*.