The commercial market for computer games and other multimedia products is extremely large and young people have a considerable experience of such games. Disabled users have very limited access to this important part of the youth culture. Indeed there are few entertaining computer games which are accessible for them. Research and development in the field of IT and the disabled has focused on education rather than leisure.

During the last five years a few projects have been exploring various ways of rendering game situations for users with special needs. Specific games or specific versions of mainstream games were developed, evaluated and for some of them distributed.

The TiM project [1], funded by the European Commission, created a game development API [2], "blindstation", which helps to design accessible games, together with a set of games which were used to study various game situations with users. One of these games, Mudsplat is currently available for download:

http://www.timgames.org/mudsplat

One can find a good number of audio games. The Swedish Library of Talking Books and Braille (TPB) has published web-based games dedicated to young children with visual impairment [3]. On the other end, Terraformers is the result of three years of practical research in developing a real-time 3D graphic game accessible for blind and low vision gamers as well as full sighted gamers [4]. A quite comprehensive list of audiogames can be found at:

http://www.audiogames.net

Various kind of tactile and haptic devices have been explored too [5,6,7]. [8] proposes a game generator to create simple audio/tactile games dedicated to very young children. [9] reports experiences about virtual reality games involving blind users carrying devices in a backpack and moving on a real football field.

In the paper entitled "Semi automatic generator of tactile video games", Alexis Sepchat will present his work about the representation of a 2 dimensional game space on a linear Braille display, and his models to navigate and play using this kind of devices.

From the experiences collected, we can bring together a set of general rules allowing to improve the accessibility of mainstream games [10,11]. The IGDA (International Game Developer Association) published a white paper about accessibility of mainstream games [12].
Mathew Atkinson’s paper, “Making the Mainstream Accessible: What’s in a Game?”, based on his experience in AudioQuake (first-person shooter games accessible to visually impaired people), discusses some of the low-level accessibility infrastructure employed in this game and compares it to other contemporary research.

Dimitris Grammenos, in the paper “Access Invaders: Developing a Universally Accessible Action Game”, discusses the notion of Universally Accessible Games. The paper is based on a case study (the Space Invaders game).

In the paper “Internet and accessible entertainment”, Morten Tollefsen will present the development of an Internet based game, “HeiPipLerke”, which is used as an example of methodological approach in the framework of the UPS project [13].

The next step is to write guidelines for accessibility of Games, and to make them accepted and used by mainstream game developers. These guidelines will allow:

- To facilitate the development of specific software for disabled users
- Entertainment software publishers to develop new products that respect as well the reality of the market and the needs of all users. Indeed young people with disabilities wish to use the popular computer games used by friends and family.

During the Workshop on Accessible Games held in Linz in November 2005, this task was set up as an international collaboration, leaded by University of Linz and MediaLT. A working group was created and the framework of the accessibility guidelines was outlined. In the paper “Guidelines for the Development of Accessible Computer Games”, Roland Ossmann will report the progresses of the working group. The current state of the guidelines will be presented, together with the system used to work on these guidelines.

References


Semi Automatic Generator of Tactile Video Games for Visually Impaired Children

Alexis Sepchat¹, Nicolas Monmarché¹, Mohamed Slimane¹, and Dominique Archambault²

¹ Laboratoire d'informatique de l'Université François Rabelais de Tours, France
alexis.sepchat@univ-tours.fr, nicolas.monmarche@univ-tours.fr, mohamed.slimane@univ-tours.fr
² Université Pierre et Marie Curie, INSERM U483\INOVA, France
dominique.archambault@upmc.fr

Abstract. Currently, few video games are accessible for visually impaired people. Nevertheless, there are two ways in order to improve video games accessibility: the use of sound or the use of touch. Even if the latter turns out to be the main exploited solution, the use of touch remains substantial. Indeed, touch is the base of Braille learning and Braille knowledge is the only way for visually impaired persons to access written information alone without any technical help such as computer and vocal analysis [1]. This article introduces our works about tactile video games. It shows games like Snake or Maze, which can be played from a Braille display. Finally, these works have led us to think about the way to introduce tactile games as play aspect in Braille learning[2,3].

1 Introduction

Created during the first half of the 19th century by Louis Braille, Braille was an important designing issue for visually impaired people. Nevertheless, the volume of transcribed data is significant. That is the reason why the association of computer science and Braille terminal (in the beginning of the 80's) has been a great revolution in the development of Braille. It allows the association of both an accessible representation technique and a storage one.

But, if at its beginning, computer science was easily accessible because of the simplicity of its interfaces. Its improvement has led to a loss of the numeric accessibility. So, at present, if it is a wonderful learning tool for visually impaired people, it above all remains a working tool and does not have an entertaining aspect as for other people without any visual deficiencies or with other kinds of deficiencies. Yet, computer games are a wonderful tool to learn [2].

In previous works in the bosom of the Computer Science Laboratory (L.I.) of the University of Tours, we have already focused on this entertaining aspect. We have developed games based on sounds and textures for young children and the corresponding game generator [4]. These works were a part of the TiM project (Tactile Interactive Multimedia) [5] which goal is the development of accessible video games for blind and partially sightless children.

© Springer-Verlag Berlin Heidelberg 2006
This new study focuses on games based on tactile modality and more especially the use of Braille terminals [6]. It means that the average age of the target audience is higher than in our previous study. The development of such games has led us to the conception of a semi automatic generator of such games.

This article is organized as follows: in section 2, we will introduce some generalities about accessible video games. Then in section 3 and 4, we will give details about the semi automatic generator of tactile accessible video games for visually impaired children and the remaining works. In section 5 we will show some examples of games and representations developed from this semi automatic generator. In section 6, we will introduce few results of our games tests. The last section will be devoted to the area of improvement that we have identified and conclude this paper.

2 Accessible Video Games

Standard video games are not, or at least not easily, accessible. These difficulties are linked to players’ perception problems but also his/her abilities (mental or motor): speed or complexity of the game. Our works, based on visual impairment, focus essentially on the perception problems.

Visual impairment prevents player from using standard output devices. So the main problem of visually impaired player is to acquire the game information. Two modalities can be used to replace visual modality: tactile modality or sound modality. Till now, sound one keeps being the most important modality, almost all accessible video games or projects about accessible video games use it. Nevertheless, tactile modality is very important. Not only could its use, thanks to Braille terminals (Cf. figure 1), permit to improve the touch sensibility of the visually impaired players but it could also be used as a entertaining aspect in the learning of Braille.

Moreover, such equipment (Braille terminals) is very expensive. Consequently, people can not get several ones and the created games have to run whatever the Braille terminal used. We have thus decided to use the Libbraille library [7].

Besides, a second very important aspect in accessible video games is to take several interfaces into account. Indeed, everybody must be able to play with these video games and not only visually impaired persons. So accessible video games developers must not create a new video games universe but a sharable universe.

Fig. 1. Braille terminal : ECO ONCE 20 - Braille cell details

http://www.audiogames.net/
in which impaired persons and others could play together. Several projects take 
this aspect into account [8,9]. A way to create such a universe is to separate the 
game itself (data, structure, . . . ) and its interfaces (input and output interfaces). 
This separation would facilitate the development of new interfaces, taking into 
account particular impairments or not , and following several guidelines²³⁴.

3 The Semi Automatic Generator

3.1 Introduction

Although most of the accessible video games for visually impaired children are 
based on sounds, the use of the touch modality is very important, in particular as 
play medium as part of Braille learning. But several difficulties are encountered 
during the development of tactile video games using a Braille Terminal. These 
difficulties are linked with the transcription of the game on the Braille terminal:

- the game dimension and the Braille terminal dimension are different. Indeed, 
games chosen are 2D ones whereas the Braille line can be considered as 1D 
environment or as a very restricted 2D environment;
- the limited number of Braille cells. All the game can not be transcribed 
on the Braille line. So the most important pieces of information have to be 
extracted;
- the structure of the Braille representation of the game must be easy to read 
because the player needs to read it quickly.

In order to facilitate the development of new games, as well as, the creation 
of new Braille representations for an existing game, we have decided to develop 
an automatic generator of such games. Its aim is to develop new games without 
programming their main structure, just focusing on the most important points 
like Braille representations. So everyone, with just a few knowledge in computer 
science, can create its own games and especially its own representations: medical 
personal, educators, parents, . . .

Obviously, the developed games must have a similar structure (internal struc-
ture or visual interface) and all the games have to follow the same philosophy. 
The one we have decided to use is the movement of a character inside a grid. 
A grid can be considered as a whole of sectors where each sector is in a given 
state. The different states are limited and they depend on the game. Even if this 
choice can be considered as very restrictive in the development of next games, 
finally most of them can be perceived like this.

For example, we have already developed two games (and several Braille repre-
sentations with each one): a Maze Game and a Snake Game (Cf. section 5). But 
most of games can be considered like these ones: sport ones, platform ones, . . .

² (Roland Ossmann) - http://gameaccess.medialt.no/guide.php
³ (Medialt) - http://www.medialt.no/rapport/entertainment_guidelines/index.htm
⁴ (IGDA) - http://www.igda.org/accessibility/IGDA_Accessibility_WhitePaper.pdf
Unfortunately, in the current version, this tool is still a semi automatic generator. Therefore, the result is reduced to a skeleton of the expected game. It remains necessary to manually finalize this game. But the work is amply facilitated as we will demonstrate in the following section.

3.2 The Several Forms

This semi automatic generator is based on several forms (4) allowing the access to information about the game to build.

First Step: General Information. The first form deals with general information about the game like its name but it essentially focuses on the grid of the game.

Indeed, one game is associated with one grid, which means that each game has to define its own grid i.e. its size (i.e. its size - number of sectors per line and per column) and the state of each sector of the grid. Nevertheless, a project of automatic generation of grids is in progress.

Then, the square size is linked with the representation of the game on the visual interface: each sector is represented by a square in the visual interface.
This information allows to define its size. It can be interesting to regulate this size to give partially sighted children the possibility to play these games.

Second Step: Moving. This second form deals with the moving of the character inside the grid.

This form permits to choose the character way of moving when the player hits a multi-directional control key or a timer expired. Two ways of moving are available:

- Multi-directional: The character moves according to the control key hit.
- Forward (key "up"), Backward (key "down"), Rotation (keys "left" and "right": modify the current character direction in clockwise or anticlockwise): The character moves according to its current direction and the control key hit (forward if a timer expired). In that case, it is necessary to transcribe the direction so this information will appear in the first cell of the Braille representation.

Third Step: Window and Braille Cell Structure. A Braille terminal has a limited number of Braille cells. So, the whole game cannot be transcribed into the Braille representation and some pieces of information have to be selected.

On the one hand, this selection begins with the definition of a window containing all the sectors which have to be transcribed (these sectors are called "direct environment"). This window is centered on the character and can take many patterns like a line, a column (the current line or the current column of the character), a 3x3 window, ...

Moreover, the number of sectors contained in this window depends on the number of splinters associated with each sector. This number of splinters depends itself on the number of states available in the game because each state has to get one only Braille representation. Finally, the more states there are, the more splinters are necessary and the less sectors can be transcribed.

For example, in a game with 9 states, it is possible to transcribe these 9 states with 4 splinters \(2^4 = 16\) and \(16 > 9\). So, only 2 sectors could be transcribed in the 8 splinters of a Braille cell \(2 \times 4 = 8\). Whereas in a game with only 4 states, they can be transcribed with 2 splinters \(2^2 = 4\) so 4 sectors could be transcribed in a Braille cell \(4 \times 2 = 8\).

On the other hand, it can be interesting to keep some splinters free in order to transcribe information (i.e. sectors) not in the initial window. Moreover, if this information is not in the initial window, it is supposed to be far enough from the character to accept a loss of precision. So it is possible to cluster states into classes. Finally, the number of classes is less important than the number of
states. So, it is possible to associate new configurations of splinters containing less splinters to each class.

For example, in a game with 9 states, 4 splinters are associated with each state. If these 9 states are clustered into 4 classes. Each class are associated with 2 splinters. Finally, each Braille cell transcribes one sector from the direct environment (contained in the initial window) and two sectors from the indirect environment ($1 \times 4 + 2 \times 2 = 8$).

Finally, this form allows to define the pattern of the initial window and the number of splinters associated with each environment (direct and indirect one).

**Fourth Step : States Management.** This last form allows to control each state of the game: its name, its color, the corresponding configuration of splinters or the behavior of the game when the character encounters this state during its movement.

4 Games Skeleton

Finally, this semi automatic generator builds a Python skeleton of the game wished. The programmer (or the person with a few notions of programming) needs to finalize the game. The main points of this finalization consist in:

- placing the several elements on the grid (currently just the obstacles and the enemies can be place from the generator);
- configuring the behavior of the character during its movement (when it hits the different elements i.e. states);
- configuring the building of the Braille representation i.e. the way of browsing the initial window and the location of each information on the Braille cell.

5 Snake and Maze Games

Two tactile video games based on the moving of a character inside a grid and several representations have been developed with this tool:

1. a *Maze game*: the player has to find the exit, collecting items and avoiding enemies and obstacles;
2. a *Snake game*: the player is a snake and has to find and eat apples without hitting its tail or wall. When the snake eats an apple the score is up but the length of its tail increases too.

To illustrate the building of a Braille representation, the following example has been extracted from the Maze game: In that game 7 states are available (character, empty, out of game, obstacle, enemy, item and exit) and have been associated with 4-splinters configurations ($2^4 > 7$ - Cf figure 3). Then, these 7 states have been clustered into 4 classes (Empty, Jamming : obstacle or out of game, Interesting : exit or item, Danger : enemy) each one represented with 2-splinters configurations ($2^2 = 4$ - Cf figure 3).

In that example, the initial window is the current character line (Cf figure 4). Finally, a Braille representation example could be the following (Cf figure 5).
6 Evaluation

In order to validate our games, few tests have already been realized with several visually impaired adults who are advanced Braille reader but also with people without any visual impairment (playing just from the Braille representation of the graphic interface as in figure 5).

On one hand, people without any visual impairment understood very quickly the game because we have used the whole game grid interface simultaneously with the Braille representation one to explain them the functioning of the game. They have met no difficulty in understanding the Braille representation building.

On the other hand, this explanation was more difficult with visually impaired player. But after few minutes of explanations (between 5 and 10 minutes), most of them played without any difficulty and several ones even suggest new Braille representations. Few players did not manage to play because of a lack of time for explanations or because of additional motor impairment.

7 Conclusion

This tool is a semi automatic generator, a programmer's intervention remains necessary. We have made this choice in order to develop a friendly-use tool to use but especially to keep a large range of personalization. Indeed, a full generator would be more complex to use because it would have to take into consideration many particular cases in order to create full games. Consequently, forms would have to be more complete. Moreover, more complete forms would reduce the generation of new games because it would prevent the programmer from creating new scenarios.
Additionally, these games are still exclusively based on the tactile modality. It would be interesting to develop games using simultaneously tactile and sound modalities. This combination would permit to develop new representations offering a new element to transcribe information.

This study proves that it is possible to create a new kind of games based on Braille displays. Even if Braille representations have to be simplified to focus on Braille reader beginner or easier games have to be used (as game based on words, ...), it would be very interesting to insert them as entertaining aspect into Braille learning.

To conclude, the tactile modality is substantial in visual impairment universe but there are currently very few video games using this modality. Our works have already led to the development of two tactile video games, but especially to the development of a semi automatic generator. The aim of this tool is to facilitate the creation of such games and to help visually impaired children to develop their touch, to familiarize themselves with Braille terminals and to help them in Braille learning through games.

References

1. CNIB: The importance of braille literacy. (The Canadian National Institute for the Blind)
Making the Mainstream Accessible:
What's in a Game?

Matthew T. Atkinson¹, Sabahattin Gucukoglu²,
Colin H. C. Machin¹, and Adrian E. Lawrence¹

¹ Loughborough University, Loughborough, Leicestershire, LE11 3TU, England
² The AGRIP Project

Abstract. Though accessible gaming is a well-established phenomenon,
what mainstream applications of it exist. We present some of the work of
the AGRIP project—an effort to develop techniques to render modern
first-person shooter games accessible to the blind and vision-impaired.
We discuss some of the low-level accessibility infrastructure employed in
the game AudioQuake and compare it to other contemporary research.
The project's ultimate goals of generalisation and use of the technology
in educational settings are also introduced.

1 Introduction

AudioQuake is the first adaption of an existing mainstream game designed specif-
ically for sighted people that has been made playable by blind gamers. It is unique
in terms of the range of Internet-enabled gameplay modes it provides. At one
level, it could be termed an "accessibility layer" for Quake¹.

This paper describes the work of the AGRIP project—an effort to develop
techniques for making mainstream games, tools and their communities accessible
to blind and vision-impaired gamers that has been active since May 2003. The
approach taken by this project contrasts with other contemporary research [1,2,3]
in the following ways.

- Adaptation - Whilst other projects often develop engine and game platforms
  anew, this project modifies existing well-designed mainstream technology to
  improve its accessibility and usability for all. The goal is not to retrofit acces-
sibility, but to show how properly-designed systems may be made accessible
  through the processing and rendering of information at separate levels.

- Generalisation - An important aim is to use the project to develop a deeper
  understanding of accessibility barriers that can be used in the development
  of general techniques to deal with those issues. We hope to create "portable"
solutions to accessibility problems, that can be used in other (academic and
leisure) settings.

¹ The seminal first-person shooter from id Software.

© Springer-Verlag Berlin Heidelberg 2006
Throughout development, the AGRIP project has been shaped by community feedback. This incorporates that given by users via e-mail and using project mailing lists. Many comments from users and suggestions for improvements have been made. Using the mailing list approach has enabled a number of interesting discussions to take place between users of the software, thus giving us a greater insight into how effective the work has been. AudioQuake has been used as the basis of a number of educational workshops at the 2005 International Computer Camp for Vision-Impaired People.

The rest of this paper expands on the user requirements, design, technical and other issues encountered in providing low-level access to the game (essentially local navigation).

2 Local Navigation

There are two main strands of navigation present in almost any computer game, and in any problem that involves searching a solution space. These are global navigation towards one's ultimate goal — the silver key; the red team's flag; the shortest path between nodes in a tree — and local navigation — “How do I get out of this room?”. At the low-level game accessibility stage of the project's development, the primary concern was to develop the mechanisms to support effective local navigation.

In this style of game, the player controls their character directly, and is presented with the character's world view. A number of “devices” which act as navigation aids are provided to facilitate accessibility. These devices attempt to resemble real-world mobility aids where possible and encourage the user to navigate the virtual world in a similar way that they do in the real world. For example, to allow players to get a very fine-grained sense of what obstacles/structures surround them, a “sweep” can be performed. This gives similar feedback to that which a mobility cane might (albeit in sound), though with a longer range.

This method of increasing accessibility has enabled users of different abilities to use only the devices they need. This approach can be found in some other fast-paced accessible games such as GMA’s “Shades of Doom”.

3 Structure Adaption and Filtering

Many existing accessibility systems are tightly coupled to an underlying mainstream technology used by non-disabled people. They provide accessibility by processing the output created by a given mainstream system and adapt it for users with certain disabilities (this is the way that screenreaders and accessible PDF viewers work, for example). This can cause a number of problems such as:

- Much work is required to interpret the meaning of the original output, based on visual markup such as layout and colouring [4,5].
- It may not be possible to directly extract or infer the types of information that specialist/disabled users require from output targeted at the “normal” user.
When the underlying technology becomes obsolete, so does the accessibility system built on top of it\(^2\).

The examples above fit in with many contemporary approaches, which emphasise the idea that to truly cater for the varying needs of users, the information behind any final output must be adapted for those needs. Rendering should be decoupled from information processing so that “accessibility” systems can utilise data at the same level that mainstream systems currently do.

### 3.1 Domain-Specific Solutions

In the case of the current work, this was achieved by utilising the architecture of modern computer games. Principles for how the information must be adapted were developed. Some domain-specific principles of structure simplification we have developed are described below. It would be very useful if the principles could be made more general; work towards achieving this is being carried out.

- **Necessity-Based Rendering** - It is not required, or even useful, to bombard the user with an audio interpretation of all graphical output from the game. In the context of navigation, obstacles such as walls need only be explicitly rendered\(^3\) when they become obstacles for the user. Doors or ramps, however, should always be rendered, even if they are not directly in front of the player.

- **User-Centred Flexibility** - Even when considering only blind gamers, we found significant diversity in terms of preferences for rendering style. Two main schools of thought existed\(^4\). Within these, users had differing opinions on factors such as how quickly object indicator sounds should fall off with distance, how often scans for objects should be carried out and so on.

Note that through adhering to these principles, no information has been explicitly added – this does not appear to be necessary to help disabled people overcome barriers to accessibility. Our and others’ work [6,7] has shown that simply adapting the underlying information and transforming it into the most appropriate format for rendering is the most important factor in increasing accessibility.

In fact, feedback given by AudioQuake users\(^5\) would as far as to suggest there is no such thing as a “fair advantage” – trying to overcompensate for an inaccessible system by creating supposed advantages for disabled users may simply serve to confuse them.

---

\(^2\) A discussion of alternative, more loosely coupled architectures is out of the scope of this paper but is currently being investigated.

\(^3\) This does not preclude effects such as echos from being used to give the user an impression of their immediate environment.

\(^4\) These differences, and their implications, will be discussed in a separate paper.

\(^5\) There is a core set of roughly 15 blind and vision-impaired gamers that take part in our electronic discussion group.
3.2 Case Study: The EtherScan Radar

An example of the above phenomenon can be found in the behaviour of a navigation aid present in AudioQuake. The “ESR” warns players of nearby enemies and team mates using a RADAR-like metaphor: sounds emanate from the position of the enemy and have a gain and repetition speed proportional to the players’ distance from them.

Originally it was conceived that allowing the ESR to “see” beyond walls and doors would help give blind players more chance of survival in the game. In reality, this feature—intended to help users—interfered with their sense of global navigation and caused them to track enemies behind walls, getting both stuck on obstacles and frustrated in the process.

Alternative indicators for enemies obscured by walls (such as muffled sound effects) could be used—and their effects may be investigated. However, the principle of conveying the required information in the simplest possible way and the avoidance of information overload [8] has been a prevalent (and successful) theme during the development and use of AudioQuake.

This result implies that mixing local and global navigation in an ad-hoc manner, within the same modality, can be confusing for users and detrimental to their ability in and enjoyment of games (and, thus, similar other activities).

3.3 Generalisation

A greater understanding of why certain pieces of information can be omitted from the auditory representation of the visual scene could be useful. This would enable us to construct a model of how people with certain disabilities (sight loss in this case) navigate, highlighting the types of information they require to do so effectively. From such a model, it would be possible to determine how this required information might be provided in this and other settings. Currently the implementation of navigation aids in AudioQuake serve as an empirical model. However, creating a more abstract and general version is an important goal and further work is being carried out in this area.

4 Serialisation and Prioritisation

A common generic approach to increasing the accessibility of a system is to (a) create a method for serialising its output and (b) put measures in place to ensure that the linearised output remains understandable (this may involve prioritising the rendering of parts of the output based on user needs). Examples of this are the way in which screenreaders interpret HTML pages and the work of the LAMBDA project [9]. This approach is popular and has a number of advantages, some of which are described below.

- It reduces multidimensional problems into single-dimensional problems, which may be easier to understand or at least display (the two most popular accessible output formats are speech and 1-dimensional Braille displays—both of which are linear).
In some cases it is the more cost-effective and achievable approach, especially in areas such as web accessibility where retrofitting is more attractive to companies than redesigning their web presence. Sometimes it is the only known effective way of conveying information in an accessible form.

However, there are also some significant disadvantages to serialisation, which are highlighted below.

- Multiplexing nominally multidimensional data so as to render it via a 1-dimensional output medium may be cognitively demanding for users.
- Though it could be the most economically attractive approach, requiring less re-working of existing systems to implement, it may well not provide as sophisticated a level of accessibility as other methods.
- On its own it simply puts a (potentially large) amount of information into a format that can be "read". This does not necessarily make the information easier to understand (problems with navigation through the data may well occur). Other techniques are needed to complement serialisation to prevent information overload.

4.1 Domain-Specific Factors and Solutions

The above points mention some systems that the user can interact with largely at their own pace. However, many 3D computer games (and similar systems) pose a number of additional challenges:

- They are often fast-paced and time-critical.
- They generate a large volume of (mostly visual) information.
- This information spans multiple domains (spatial, strategy, storytelling and is presented in parallel.

AudioQuake does have some basic characteristics in common with hyper-stories such as AudioChilile [11], however it is much more like other audio/accessible first-person-shooter games such as Terrafomers [2] and Demor [1], though unlike the latter it is targeted firmly at users who can only access commodity computing hardware.

---

6 e.g. Screenreaders can linearise HTML tables with little effort required on behalf of the content producer. The planning agent approach [10] requires the author to undertake some extra work in order to work most autonomously (the adoption of newer mark-up standards would negate the need for a lot of this work, but that alone incurs significant effort).

7 e.g. The planning agent approach [10] is capable of providing significantly more power and flexibility to the user than the serialisations of tables produced by a screenreader, but to work most autonomously it requires that the site be coded in XML (partly due to the semantic reinforcement afforded by meta-data).

8 The LAMBDA project, for example, makes use of a hierarchical structure-exploration mode, for example.
One major difference, however, is that it is based on technology (Quake) not originally designed for non-sighted or otherwise disabled players\(^9\). Though this technology's architecture provides a solid base for building accessible games, the challenges listed above are potentially much more pronounced due to the fast-paced and unforgiving nature of the gameplay. Our approach to managing these issues is centred around the following key ideas:

- We imagine that there is a certain amount of bandwidth available for sending data from the computer system to the user (this may vary based on user capabilities).
- Information is streamed from the computer to the player. Some information is more urgent than other information and must therefore be rendered in an appropriate order.
- Different domains of information may be sent on different "channels"; these roughly correspond to different output devices/types of output – i.e. the system is multimodal.

We present a number of complementary techniques for effective serialisation.

- **Periodic Rendering by Priority** - The main tasks carried out by modern games and 3D applications run continuously. In one "tick" the screen is updated and physics rules, AI and gamecode are executed to update the user on the current state of the virtual world. Though it is appropriate to render visual, and some auditory, information at a high rate, this is not always necessary. For example, the player doesn’t need to be constantly reminded of the locations of powerups (weapons, health and other such items). Indication of their position can be given periodically, as opposed to continuously.

- **Sub-Domain Prioritisation** - Information in a given domain may have varying importance. For example, enemies further from the player, or out of weapon range could be rendered with a lower priority than those which are within range. Similarly for powerups.

Different schemes for prioritisation within each domain/stream were developed and found to be of use to AudioQuake players\(^10\).

- **Multimodality across Domains** - The benefits of multimodal interfaces have been discussed extensively [12,13]. Rendering each stream to a different output device may not be possible, as there are a limited number of commodity output devices available. However, rendering all navigational information using non-speech audio and presenting communication between users using text-to-speech seems to have enabled AudioQuake players to understand the two sound-based streams separately, as intended.

The use of haptic feedback (from Braille Displays to force-feedback input devices) could further separate out streams and enable users to interact more comfortably with the system, as demonstrated by other research [14].

\(^9\) Fortunately, it was designed well, which enables us to implement accessibility – alternative rendering – at the correct level.

\(^10\) This should not be confused with the filtering techniques described above – even after filtering out surplus information, some prioritisation may be necessary.
Cross-Domain Prioritisation - When the pace of events in the game increases, the player could require more information to complete certain tasks. For example, when interacting with a number of enemies, they should ideally be informed of the position of all enemies (as opposed to just the closest) and perhaps any nearby powerups which could assist them. This has not yet been implemented but feedback from users indicates it could be worthwhile. The general idea of load-balancing in computer networks or the grid [15] could well be applied in this situation - with the goal of sharing output bandwidth amongst modalities and devices in such a way that the user does not become overloaded with information.

More implicit, techniques for improving bandwidth usage and immersing the player more in the game are out of the scope of this paper and will be discussed separately.

5 Links to Education

There is a growing interest in the potential usefulness of game-like technology for education [16,17]. This work aims to promote inclusion in a number of ways.

- Application to Other 3D Engines - The techniques developed are quite generic and could be employed elsewhere (for games or other applications).
- Collaboration - Though out of the scope of this paper, AudioQuake promotes accessible online collaboration, promoting integration with sighted students who may be using the mainstream version of the same technology.
- Uses in Other forms of Navigation - Mobility training and the navigation of complex data structures are similar tasks in many respects and could make direct use of the navigation aids described here.
- Direct Educational Uses - As well as the potential applications listed above, some more direct benefits of accessible 3D engines and games exist. They can be used as practical material in the teaching of programming, networking and even the development of important algorithms related to AI/machine-learning and searching [18].

6 Conclusions

We have discussed a number of issues that are key to providing accessibility to 3D environments such as computer games. Many of these challenges - relating structure/space to the user; presenting information that the user needs, quickly - are common to a number of accessibility and usability problems. We hope to generalise the currently domain-specific solutions to other areas.

References


Access Invaders: Developing a Universally Accessible Action Game

Dimitris Grammenos¹, Anthony Savidis¹, Yannis Georgalisl, and Constantine Stephanidis¹²

¹ Foundation for Research and Technology - Hellas (FORTH), Institute of Computer Science (ICS), Human-Computer Interaction Laboratory, Heraklion, Crete, GR - 70013 Greece
{grammenos, as, jgeorgal, cs}@ics.forth.gr
² Department of Computer Science
University of Crete

Abstract. This paper depicts the notion of Universally Accessible Games and presents the development of a related action game entitled Access Invaders. The design of the game's user interface which accommodates concurrently the needs of people with diverse abilities is described, along with the approach followed to adapt the game logic and content to achieve accessibility. In this context, the concept of Parallel Game Universes is introduced and suggested as a solution for the creation of multiplayer universally accessible action games.

1 Introduction and Problem Description

In the past few years, the accessibility of electronic applications and services by disabled people has become a topic of paramount importance at an international level. Based on the fundamental right of all people for access to information and services, and equal opportunities for employment and independent living, several governments and international political bodies have adopted legislative and policy measures for application software and Web accessibility (e.g., [1], [2]). One of the basic needs of most people, beyond working and independent living, is entertainment. Presently, computer games constitute indisputably one of the major related sources.

Unfortunately, computer games are usually quite demanding in terms of motor, sensor and mental skills needed for interaction control, while they often require mastering inflexible and complex input devices and techniques. These facts often render games inaccessible to a large percentage of people with physical (or situational) disabilities. So far, little attention has been paid to the development of computer games that can be played by all players, independently of their personal characteristics, requirements, or (dis)abilities. Furthermore, concerning human-computer interaction issues, computer games have fundamental differences from all the other types of software applications, for which accessibility guidelines and solutions are already becoming widely available.
2 State of the Art in R&D and Application

In contrast to Web accessibility, up to now, relatively few efforts have been devoted to game accessibility. Currently, there are no related official guidelines or standards, nor any world-wide initiatives promoting game accessibility, and no related governmental or legislative actions. At present, the only support and design knowledge available to developers for creating accessible games is limited to some indicative approaches outlined in a White Paper of the Game Accessibility SIG of IGDA [3], as well as general-purpose guidelines for developing accessible software (e.g., [4,5,1]). From a technical point of view, two main general approaches have been adopted to address the issue of computer games accessibility:

1. Mainstream games are developed to be compatible with the use of assistive technologies, such as screen readers, mouse emulators or virtual keyboards.
2. Special-purpose games are created, optimally designed for people with disabilities, like audio-based games for the blind and switch-based games for the motor-impaired.

The first approach typically achieves very limited accessibility and suffers from low quality in use. The second approach, though being the most promising from a quality point of view, has two key drawbacks: (a) the cost of developing high quality games is prohibitive when the potential target group is limited; and (b) there is an evident hazard of segregation between able-bodied and disabled gamers, leading to potential social exclusion.

Currently, there are no computer games that can be concurrently played by people with different disabilities sharing the same computer. Recently, a few games for both visually-impaired and fully sighted players have been developed (e.g., All inPlay\(^1\) card games; 3D shooter Terraformers\(^2\)). Several audio-only games adopting the Space Invaders theme exist for the blind (e.g., a.Shooter\(^3\), Sonic Invaders\(^4\), Alien Outback\(^5\)). All are single-player sound-only games, with no configuration capabilities. A game with both audio (in Japanese) and visual output is Space Invaders for the Blind\(^6\) by Taito. A 3D Space Invaders game has been developed\(^6\) which combines audio and visual interfaces with force. Regarding people with motor impairments and low vision, only Alien Invasion\(^7\) is available. The game can be played with a range of standard and adaptive technology controls, and allows players to adjust the parameters of the major game elements, also offering large character sizes for low vision players.

\(^1\) http://allinplay.com/
\(^2\) http://www.terraformers.nu
\(^3\) http://www.agame.org/en/aShooter/index.html
\(^4\) http://www.mytaras.com/sonicinvaders.html
\(^5\) http://www.espsoftworks.com/
\(^6\) http://homepage2.nifty.com/JHS/spi.html
\(^7\) http://www.areness.com/aliens.htm
3 Research and Methodological Approach

In order to overcome the limitations of previous approaches to game accessibility, the concept ofof Universally Accessible Games (UA-Games) has been proposed, primarily emphasizing game accessibility, but also putting forward the objective of creating games that are concurrently accessible to people with diverse abilities. UA-Games are interactive computer games that:

1. follow the principles of User Interfaces for All [7], being proactively designed to optimally fit and adapt to different individual gamer characteristics without the need of further adjustments or developments;
2. can be concurrently played among people with different abilities, ideally also when sharing the same computer;
3. can be played on alternative technological platforms and contexts of use using a large variety of devices, including assistive technology add-ons.

The potential impact of UA-Games in the upcoming Information Society is threefold: (a) they open up and enhance an entertaining social experience that would otherwise be unavailable to a significant percentage of people; (b) they allow for social interaction among people who may never have (or could have) interacted with each other; and (c) they considerably expand the size and composition of the potential market of the computer games industry.

3.1 From Space Invaders to Access Invaders

Action games constitute a real challenge for Universal Access. They have highly dynamic content, since they comprise many different moving objects with alternative characteristics, they usually require complex controls and are based on reflex-based reacting, a fact that can render them particularly hard for those who have difficulty or cannot use their hands or eyes. Also, devising ways for two players with different disabilities to play, cooperatively or against each other, the same action game, is a very challenging research issue.

In this context, a universally accessible version of the classic action game Space Invaders by TAITO has been designed and developed. The new game, named Access Invaders, achieves Universal Access by supporting alternative input / output modalities and interaction techniques that can co-exist and cooperate in its user interface, combined with tailorable player profiles and game content. The game is highly customizable and supports the creation and use of unlimited user profiles. Each game parameter can be adapted both based on the player’s profile and the current game level. Non-visual gameplay is also supported. In this case, full acoustic rendering of game information is provided through spatial audio and a built-in screen reader. Multi-player games are available, where people with different (dis)abilities can play cooperatively, sharing the same computer. In this case, the game’s interaction parameters can be independently adjusted for each player. An unlimited number of concurrent players is supported. Access Invaders is available for download as freeware at http://www.ics.forth.gr/hci/ua-games/access-invaders

http://www.ics.forth.gr/hci/ua-games
3.2 Game Design

A prerequisite for creating UA-Games is *inclusive design*, which implies that the requirements of the broadest possible population should be taken into account during the design phase. For designing Access Invaders, the following user categories and respective requirements were considered: (a) people with hand-motor impairments; (b) blind people; (c) people with deteriorated vision; (d) people with mild memory / cognitive impairments and novice players; (e) people belonging in more than one of the previous groups. To elicit user requirements, three basic sources were used: (a) relevant bibliography; (b) interviews with experts and representatives of target user groups; and (c) observation of potential users while playing related computer games.

In order to be able to effectively accommodate the particularly broad spectrum of diverse interaction requirements, the game was firstly designed in a context-independent way, i.e., at an abstract level, minimizing references to specific interaction modalities, metaphors, techniques or devices. Additionally, it was necessary to appropriately map the abstract design elements to low-level physical, interaction styles, meeting the requirements of each target user group. In this context, accessibility barriers related to the game’s interface, but also barriers stemming from the game’s content and rules were identified, as along with possible design strategies for overcoming them. For these purposes, the Unified Design method was adopted, reflecting a procedural design discipline of abstract task definition with incremental polymorphic physical specialization.

Based on the high-level design, indicative electronic prototypes of the game were developed, showcasing alternative interactive properties of its user interface for the different target user groups. These prototypes were informally evaluated with the stakeholders that participated in the design process using the *thinking aloud method* [9]. During the evaluation, participants were prompted to express their thoughts, comments and feelings. The outcomes of this process aided in validating, correcting and updating design decisions, as well as in developing new ideas for improving the accessibility of the final game.

3.3 Adapting the Game to the Player’s Needs

The main mechanism of game adaptation are *profiles*. Once a profile is selected the game’s interaction and content adapt to its characteristics. Thus, in practice, the game is viewed by players as a collection of different games integrated into one, from which they can select the one that is most suitable for them. On the other hand, from a developer’s point of view, it is a single application with unlimited alternative manifestations. Access Invaders currently offers seven predefined profiles but new ones can also be created. Some indicative examples of the game’s adaptation capabilities are provided in Fig. 1.

**Adapting the User Interface.** The game’s interface can be controlled using any, or all, of the following devices: keyboard, mouse, joystick, game pad and binary switches. The interface includes two basic interaction objects, i.e., menu and text entry. Both can be manipulated with any of the aforementioned devices, while the font size and family is configurable. Furthermore, automatic scanning is provided for
people who can use only a single switch. For supporting non-visual interaction, the objects are augmented with speech output capabilities. The player's spaceship is controlled through three commands ("move left", "move right" and "fire"), which can be issued using any device. In order to achieve accessibility for people with limited motor functions, the game can also be played using only two, or even a single command. Furthermore, two more features are supported (combined into one): (a) pausing the game, and (b) accessing an "in-game" menu with functions for resuming, quitting or changing parameters of the current session. Typically, these features are accessed through additional dedicated controls (e.g., Escape key, right mouse button). Nevertheless, in the case of limited motor control this is not possible. In this case, the menu is activated if all the available commands are concurrently issued for a certain amount of time (e.g., if the player keeps the "change direction" switch pressed for more than 3 seconds).

Adapting the game content and logic. As a result of an analysis of the currently available accessibility guidelines for applications and games, and after several hours of play testing with end-users, a number of adaptation categories were identified which
are outlined in Table 1. Alternative combinations of these adaptations can serve the accessibility needs of each of the addressed target user groups.

<table>
<thead>
<tr>
<th>Category</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Game Content</strong></td>
<td></td>
</tr>
<tr>
<td>Speed</td>
<td>Make individual game elements, or the gameplay, faster or slower.</td>
</tr>
<tr>
<td>Quantity</td>
<td>Change the number of aliens and shields that are present.</td>
</tr>
<tr>
<td>Size</td>
<td>Make the game elements bigger / smaller.</td>
</tr>
<tr>
<td>Layout</td>
<td>Alter the absolute / relative position of game elements.</td>
</tr>
<tr>
<td>Firepower</td>
<td>Supply the aliens / player with faster / slower and more/less powerful weapons or stop them from firing.</td>
</tr>
<tr>
<td>Visual complexity</td>
<td>Use of backgrounds and graphics with large solid colour areas.</td>
</tr>
<tr>
<td>Contrast</td>
<td>Use of colour combinations that provide high contrast.</td>
</tr>
<tr>
<td>Sound</td>
<td>Association of spatial feedback to game elements through 3D sound.</td>
</tr>
<tr>
<td><strong>Game Rules</strong></td>
<td></td>
</tr>
<tr>
<td>Interaction</td>
<td>Set which type of aliens can destroy the player’s spaceship and vice versa, and whether the spaceship’s bullets can pass through the shields or collide with them.</td>
</tr>
<tr>
<td>Analogue vs. digital control</td>
<td>Select whether the aliens / spaceship move in discrete or continuous positions.</td>
</tr>
<tr>
<td>Hints</td>
<td>Make the player’s task easier by providing added-value information, e.g., audio radar, visualisation of the path of objects, oral descriptions.</td>
</tr>
<tr>
<td>Stamina</td>
<td>Provide the player with additional “lives”, make the spaceship more resistant to incoming fire and the aliens less.</td>
</tr>
</tbody>
</table>

### 3.4 The Concept of Parallel Game Universes

Multiplayer action games constitute an open research challenge, since it is not only the interface that changes, but also the game’s content and rules. This practically means that two (or more) people should be able to play the same game, being aware of each other, while at the same time each one follows different rules and perceives distinct content. A possible solution to this problem is to allow each player to play in a different “game universe” and then somehow project each universe to the other(s). The term “game universe” is used to denote an instance of the game after it has been adapted to suit the requirements and needs of a particular player. For example, the alternative profiles of Access Invaders could be considered as different game universes.

As an illustrative example, consider the following situation. Two friends want to play the game together. One of them (Player X), due to severe motor-impairments, can use only a single switch. A manageable difficulty level includes a small group of aliens that move slowly and fire very scarcely, while the player’s bullets do not collide with shields. The second Player (Y) does not have any impairment and in order to for the game to be challenging enough, he wants to confront numerous fast, fire blazing aliens. So, in case the two players attempt to share the very same game, if this
is adapted to the first player, then it will be rather boring for the second, while if it is adapted to the second player it will be extremely difficult - if not impossible - for the first. A novel approach, following the idea of Parallel Game Universes, is to merge the two distinct game versions into one. Thus, in this new version of the game, two groups of aliens will exist: a big, fast and powerful which can destroy and be destroyed only by Player Y, and a small, slow and quite harmless that plays only against Player X. The bullets of each player will not affect the aliens fighting against the other, while Player Y’s bullets will collide with the shields, and Player X’s will not.

Fig. 2. Example of universally accessible multiplayer gaming through Parallel Game Universes

4 Future Work

In the short term, future work includes the support of tactile output through a Braille display and a force feedback joystick and stylus. Furthermore, an interactive application for editing user profiles and related game levels is under development. In the medium term, planned research and development work can be divided into two different, but highly interrelated, areas: (a) interactive profile selection, through which the game will help players in selecting a profile and game parameters that best suit their needs; and (b) dynamic gameplay adaptation, which includes monitoring the player’s
actions and dynamically adjusting the gameplay to better match the player’s skills. Longer term work entails further research and elaboration of the concept of Parallel Game Universes, since the current version of Access Invaders supports only parallel universes that can be instantiated in the same computer. In this context, the main objective is, on the one hand, to further experiment and study the potential and the limits of this approach, while, on the other hand, to implement and evaluate prototypes of distributed game universes.

References

Internet and Accessible Entertainment

Morten Tollefsen and Are Flyen

Media Lunde Tollefsen AS, Tevlingveien 15, 1081 Oslo, Norway
{morten, are}@medialt.no

Abstract. Young people with disabilities should be entitled to use technology for entertainment, but research and development in IT and the disabled has not typically focused on pleasurable activities. It is therefore important to set up guidelines that facilitate accessibility, develop good examples of accessible computer games, and not least, utilize the great potential of technology to include persons with different requirements.

1 Background

The UPS-project [1] started in October 2003 and runs for a period of three years. The project is aimed at adapting and developing entertaining software for young people with disabilities. The target group includes those with mental disabilities, or a combination of severe motor handicaps and perceptual losses. The work can be divided into three main areas:

- Adaptation of standard software
- Development of software designed to meet the needs of the target group
- Design of a website with information, game tests and free, digital materials

The project has focused on standard software, and how this can be inclusive, allowing disabled young people to use computer games popular among friends and family. However, we found during the course of our work, that it would be necessary to develop software for the most disabled users. This software should contain the best aspects of standard computer games and entertaining products. With this perspective in mind, software development for the target group, exemplified by an internet based game, is described below.

2 State of the Art

2.1 Entertainment and Simple User Interfaces

Research and development in the field of IT and the disabled has focused on education rather than leisure. Computer games, media players and other entertaining software are important for many people, and these products are typically developed without an educational motive. Entertainment is important, and of course just as important for disabled persons as for others (in some cases perhaps even more
important). Educational software is of course useful and in some cases even fun! The opposite is equally true: entertaining software may provide good motivation e.g. for learning assistive devices and standard computer skills. An example of this is found in [3], and Mr. Phillips writes: “Whenever I look at a computer, I think of all the fun to be had, not how I’ll be able to write essays or operate the lights in my room.” And “Yes, games like Warcraft III and Age of Mythology can be fast paced, but it was my desire to play such games that made me the highly proficient scanning user I am today. Gaming taught me speed”.

Several standard games can be used by the disabled, but there is also a need for simple games developed for fun only, and for the most disabled users. “Easy Games” (http://www.leripa.se/easygames.asp) and games found at http://www.arcess.com/ are examples of entertaining software for our target group. Other products, more or less educational, do exist but our experiences indicate that these products are typically either too childish or difficult to operate for young persons with the largest disabilities. Another important fact is that games are not typically localized into small languages like Norwegian.

There are few accessible net based games. Some research and development has been done for the visually impaired [4]. As described below, a net based game was desirable in the UPS-project. We have not found such games with built-in scanning etc.

2.2 Standards and Guidelines

The development of guidelines for developing accessible games [2] has been an important activity in the UPS-project, and this work was organized as an international co-operation in 2005. Guidelines are available at http://gameaccess.medialt.no/guide.php. We have implemented the most important guidelines for our target group in the product, and even tried to allow for users with other needs, e.g. that blind persons are able to use the game.

The game is developed using Macromedia Flash, and Macromedia Flash guidelines “Best Practices for Accessible Flash Design” [5] have been helpful. Supplementary product information like frequently asked questions is found on standard web pages, and WCAG 1.0 (http://www.w3.org/wai/) are emphasized.

3 Research and Methodological Approach

3.1 What Do the Users Want?

We did not want to spend a great deal of effort developing a product which was not appropriate for the target group. This is quite possible when developing a product for users with limited communication possibilities. Our approach to discovering what the users actually wanted was to arrange two brainstorming sessions/seminars with several experts. We also mapped the preferences of pupils at a special school (Haukåsen School). Although interests vary greatly (some prefer guns, others dolls) the conclusion from this work was: “internet and music”.

3.2 Best Guess Design

A participatory design approach has been used when developing the product. However, experts (teachers, pedagogues etc.) have in most cases represented the young disabled players. When designing for persons with mental disabilities, or a combination of severe motor handicaps and perceptual losses, it is necessary to build on earlier experiences, both when designing tasks and user interface. Best guess design is therefore an adequate description of the design process used when developing HeiPipLerke.

Previous work and experiences (e.g. from developing electronic picture books) suggested four important keywords for developing a new entertaining product:
1. Entertaining
2. Simple (not childish)
3. Accessible
4. Flexible

3.3 User Testing

Best guess design must be succeeded by user testing. In our case user tests should answer two important questions:
1. Is HeiPipLerke entertaining?
2. Is the product accessible and easy to use?

User testing has been performed at three Norwegian centres of competence and a school for the disabled. An interview scheme was used; however most of the findings are based on observations from teachers/helpers. In addition to direct user testing it has also been possible to send feedback from the web.

3.4 Development Tools and Accessibility Support

Although the conclusion from brainstorming sessions, expert views and the mapping of user preferences indicated an internet product, we still had to consider distribution media and which operating systems to support. It is probably easier to allow for accessibility when using standard programming languages or tool kits for disabled than to use the most popular web development tools. New internet technologies do also focus on accessibility and universal web accessibility seems to be constantly more important. We believe that in the future even more accessibility features will become available, e.g. in Macromedia Flash.

Using internet has several advantages: support for different platforms, no client installation, only one source to maintain, and less time spent on support. Three different technologies were considered: Macromedia Flash, Java (Applets) and traditional HTML/JavaScript with server-side interaction. The final choice was Macromedia Flash (Macromedia is now known as Adobe). The reasons for choosing Macromedia Flash were:
- Flash is suitable for built-in tools for creating graphics and animations
- Using mp3-files for music, speech and sound-effects is straightforward
- Accessibility support is fairly good
- Easy distribution to different medias like internet and CD-ROM (offline edition)
- Visual appeal and short development time

4 HeiPipLerke (Meadow Pipit)

The game can in short be described as a music composer adjusted to meet the requirements of the target group. When the game is started a short introductory screen is displayed while the game loads and brings up the main menu. The main menu contains three choices: start the game, setup and introduction.

When starting the game for the first time both automatic scanning and speech (audible scanning) are enabled by default. In the setup screen it is possible to enable or disable scanning and speech, set scanning interval, change the colour-scheme for the entire game, select touch screen support etc. Choices are saved and automatically used when the game is started again.

![Melodi]

Fig. 1. This screenshot shows one of three possible instruments for the melody with focus

A set of boxes is shown when the game is started. Each box can be described as a music collection. When a music box is selected, the next three screens display choices for three different categories: melody, beat and bass. In each category there are three different instruments to choose between which gives $3 \times 3 \times 3 = 27$ different songs that can be created from each initial box. More music boxes can easily be added at a later time to extend the game and give more variations.

When the “composition” is complete a simple animated show is displayed while the song is played. Each song is 40 seconds long. After playing the song there are three choices: repeat playback, return to the main menu, or send the song to a friend. The latter is a form where the players name and a friend’s e-mail can be filled in. An
e-mail is then posted to the address with a link to a website where the song can be played using an accessible media player.

5 Findings, Challenges and Solutions

Macromedia Flash was the most suitable development tool for our purpose but we faced some challenges with respect to accessibility. Some problems are described in [6]. Macromedia has however added accessibility features in MX and version 8. The main features are naming and description of all objects and tab-indexing. These features are not appropriate when developing an accessible game, and our most important findings are discussed below.

5.1 Scanning and Built-In Speech/Sound

In HeiPipLerke scanning and speech are required functionality. This functionality is not part of Flash, and therefore had to be implemented. The result of this work is techniques to add visual- and audible scanning and speech with or without scanning enabled. When using built-in speech, scanning functionality has to allow for the application messages before focusing on specific selectable objects. This is the case e.g. when asking the user to select a melody. The first melody cannot be selected/played before the guidance is spoken.

5.2 Mouse over and Touch Screens

One important feedback from the expert testers was the “mouse over” function (or object focus) e.g. an instrument starts to play when pointing on it with the mouse. This functionality is important when learning assistive devices, and is not typical in standard games.

A somewhat different challenge was reported for persons using a touch screen. When pointing to an object the object was focused. But when pressing a finger on the object, it was immediately selected i.e. the instrument did not play an example before being selected. The solution we chose was to add an option in the touch screen setup. When pressing the object for the first time it will be focused i.e. start to play and when pressing a second time the object will be selected.

5.3 Screen Readers, Color Schemes and Focus Rectangle

Although HeiPipLerke is not primarily designed for blind and visually impaired users, our goal was to make the product as accessible as possible. Flash has built in options for describing elements (name, description) and defining tab-index. These accessibility options make it possible to use the product, although the user may have to adjust the standard settings in e.g. a screen reader. One example is that it is an advantage to disable the virtual PC cursor in Jaws.

Text can be read as text (not rendered as graphics). However screen readers like Jaws will typically focus on the text elements when pressing Tab. This is not desirable in a game, and user testing indicates that a better solution is graphics with appropriate tagging.
At least two different high/low-contrast colour-schemes are needed to support visually impaired users (black on white, and white on black). Most of the objects could easily be given a new colour in runtime but a few objects involved much more work. The standard focus rectangle in Flash was reported to be a major problem. By default this rectangle is too indistinct (thin, yellow frame). On some objects e.g. form fields the default color of the rectangle is green. The color as well as the size should be user definable. A distinct focus rectangle is important for persons with a visual impairment. It is also a requirement for other disabled users. To solve this problem the standard style sheets needed to be redefined. This resulted in a lot of work and research because the colour scheme/stylesheet was not appropriately documented by Macromedia.

5.4 Browser Compatibility

HeiPipLerke has been tested with the most recent versions of Internet Explorer, Opera and Mozilla Firefox. Getting focus on the Flash application is a problem in all browsers. The Flash application has to be focused before the user is able to use the keyboard. Microsoft Internet Explorer proved to be the best browser (even if an extra Tab has to be pressed).

5.5 Forms and Built-In Scanning

From the very beginning user scanning was implemented in all screens. These screens include two forms: Setup and Send mail. User testing resulted in disabling scanning in these forms. An on screen keyboard can be used to enter text (e.g. mail address); however many in the target group are not typically able to do this without a helper. An option could have been to enable scanning with an interruption when text is entered.

5.6 List of Mail Recipients

Send the music to a friend is a popular function. Originally we did not plan a “My friends mailing list”. This is a very reasonable thing to add, and will make it much easier to send music for the target group.

6 Future Plans

HeiPipLerke is designed to work as an example of an accessible net based game and as a free entertaining product for young disabled persons. Feedback from experts and users indicates a need for products with similar user interfaces, both in applications as simple as HeiPipLerke and in more complex applications.

Localization: Music is an international language. It is therefore easy to translate HeiPipLerke. In addition to translating some text pages it is necessary to record corresponding sound files.

More contents: Music can easily be added and improved.

Hearing impaired: Because some persons in the target group of this product use sign language it is appropriate to include sign language in addition to speech. This can
be done using a free Norwegian dictionary. Free sign language videos are available in other languages as well, and even this functionality should be straight forward to translate.

New games: The techniques and programming tools developed for this game can be used in other internet applications. The need for such applications is evident. The challenge is to raise funds, establish free programming resources (e.g. students) or develop commercial business.

References

Guidelines for the Development of Accessible Computer Games

Roland Ossmann and Klaus Miesenberger

University of Linz, Institute Integriert Studieren
Altenbergerstr. 69, 4040 Linz, Austria
roland.ossmann@students.jku.at, klaus.miesenberger@jku.at

Abstract. Games are very important for learning, teaching, entertainment, inclusion. But they are of the most challenging applications concerning accessibility, and usability for people with disabilities. Especially in the context of playing together or in groups equal access is critical. In this paper we will present first attempts to define games accessibility guidelines helping game developers to design their products in a way that assistive technologies can interact with the game interface and that the parameters of usage can be adapted to the needs of people with disabilities.

1 Introduction

Computer games have become a major part in child and youth culture, but they are also played by adults. To give people with disabilities the chance to have access to these games, new methods and tools have to be developed. One possibility is to give game developers guidelines, with rules and hints how to develop accessible games.

This paper will present the work done in a working group which aims at starting an international co-operation to develop guidelines for games accessibility. The paper (in its final form) will a) the state of the art in research towards guidelines in games accessibility, b) present the state of the guidelines developed in the group c) present the system used to work on these guidelines at international level d) present other activities which should support developers in making their games more accessible (“Active Games Accessibility”).

2 State of the Art

There are two main research projects in the area of guidelines for games accessibility. Project one is from IGDA, the International Games Developer Association. Inside IGDA is the Games Accessibility Special Interest Group (GA-SIG), who has published a whitepaper [1] about games accessibility, and one chapter is about rules and hints for game developers.

The other main project is done by the Norwegian IT company MediaLT. MediaLT has developed a set of guidelines [2], which were the basis for further development of our guidelines. Furthermore MediaLT is partner in our project.
3 Research

With the guidelines from MediaLT, the rules and hints form GA-SIG and our own ideas, we developed our own guidelines and published them as a web page [3]. A web page was decided to make the GL accessible to everyone who want to bring in new ideas or want to help making the existing GL better.

These guidelines have five main categories:
- level/progression
- input
- graphics
- sound
- installation and settings

The guidelines have, beside the rules itself, a categorisation in three classes of priorities:
- Priority 1 – Must have
  Must have means, that it is absolutely necessary for the listed group of gamers. Otherwise the game is not accessible for them.
- Priority 2 – Should have
  Should have means, that it is a big help for the listed group of gamers. The game is accessible without that point, but with that point, the game is easier to learn or the fun factor is higher.
- P3 – May have
  May have means, that it is a help of feature for the listed group of gamers. The game is accessible without that point.

Furthermore there are four groups of disabilities: visual, auditory, mobility and cognitive disabilities. These disabilities are allocated to the priorities, e.g. one rule can have priority 1 for visually impaired people and priority 3 for auditory impaired people.

### 2.8 Inline tutorials

Inline tutorials and automatic help should be available in the game.

Priorities:
P1: none
P2: all
P3: none

<table>
<thead>
<tr>
<th>Author</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matthew T. Atkinson</td>
<td>Though a lot of time can be spent on producing accessible documentation (<a href="http://www.3gdev.org/InterestGroup/Documentation">http://www.3gdev.org/InterestGroup/Documentation</a>), it is also helpful to create tutorials like this, as they reinforce what the gamer (should have) learnt by reading the manual. Providing a series of levels with increasing difficulty and/or focusing on specific topics can be useful. Also, allowing the game to give feedback to the player on how they are doing or what the goals for a given tutorials are, is very important.</td>
</tr>
</tbody>
</table>

Enter a new comment to this rule

![Fig. 1. Screenshot of the Guideline Tool](image)
In November 2005, a workshop about games accessibility was held, and in this workshop the further approach was decided. The future development of the GL is split up in two groups, a main working group and a board group for decisions and quality control. The communication works over mailing lists and the possibility to post comments to each rule of the guideline. Figure 1 shows one rule of the GL (about inline tutorials), the categorisation in the priorities and a comment given to this rule.

4 Future Plans and Development

The future plan is to have a useful and usable set of GL for game development like the W3C/WAI guidelines [4] for web pages. These GL are a standard for developing accessible web pages. One of the reasons for the great success of these GL is the possibility to test web pages if they are fulfilling the GL with programs like Bobby [5].

The future development of the games GL has two main phases, in phase one are short range goals for the GL itself, in phase two are goals for the adjustment of the GL and the distribution.

To get more in the detail, the main issues in phase one are:

- Start cooperation or to intensify cooperation with other organisations and companies working in the area of games accessibility.
- More explanation and, if possible, add best practice examples and code samples to the rules.
- Implement life cycles and visioning in the GL.
- Stronger integration of AT in the GL, specially adding an input device section in the GL.
- Ideas for dissemination and marketing.

Not all of the issues of phase two will practical, but we will try to implement as much as possible. Here are the main issues of phase two:

- Introduce a certification for games fulfilling the GL (or parts of it).
- Implement a testing tool for games, testing the observe of the GL.
- Split the GL up for different platforms, e.g. PC, Xbox, mobile devices, ...
- Split the GL up for different user groups, e.g. GL for small children, GL for medical testing games, ...
- Add a search function to the GL, so that a game developer can e.g. search for all relevant rules for blind people on PCs or deaf people on mobile devices.
- Add the cultural aspect to the GL.

A very important issue is, as already mentioned in the listing above, the integration of AT in the guidelines. We decide to make an additionally document to the guidelines, in which the different kinds of AT's are listed with the possibility to add devices and descriptions to every kind of AT. The main idea of this additionally document is to give game developers an overview of all the AT's being available and to give them hints and ideas how to integrate these technologies in there games.

The future goal is to have dynamic GL, which are continuously upgraded with new ideas and to be adapt to new requirements. If these GL are used for developing games,
not only disabled people benefit, also the "normal" gamer benefit form an easier to understand and more configurable computer game.

Furthermore these GL will be part of an AGA (active games accessibility) framework, where games developers will have, beside these GL, a set of tools, code samples and functions for an easier development of accessible games.

The other parts of this AGA framework is still in the research and development phase, the next step will be a descriptive language. This language will make the connection between the game engine and all input and output devices, used in a game, with the option of a kind of free configuration between the different devices and the possibility to use more devices simultaneously. In addition, this language can be used to adapt already developed games to make them accessible.

References