MASTER THESIS

ACADEMIC YEAR 2009 – 2010

Introducing Accessibility Features in Authoring Tools for Creating Accessible Educational Games: <e-Adventure>

Author: **Elisa Sanz-Troyano**

Advisors: **Baltasar Fernández-Manjón, Ph.D.**
Pablo Moreno-Ger, Ph.D.

June 2010

Research Master in Computer Science in Intelligent Systems

Computer Science School

Dpt. Software Engineering and Artificial Intelligence

Complutense University of Madrid

PROYECTO FIN DE MÁSTER

CURSO 2009 – 2010

Accesibilidad en Herramientas de Autoría para la Creación de Juegos Educativos Accesibles: <e-Adventure>

Autora: **Elisa Sanz-Troyano**

Dirigido por: **Baltasar Fernández-Manjón**
Pablo Moreno-Ger

Junio 2010

Máster en Investigación Informática en Sistemas Inteligentes

Facultad de Informática

Dpt. Ingeniería del Software e Inteligencia Artificial

Universidad Complutense de Madrid

El/la abajo firmante, matriculado/a en el Máster en Investigación en Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Máster: “Introducing Accessibility Features in Authoring Tools for Creating Accessible Educational Games: <e-Adventure>”, realizado durante el curso académico 2009-2010 bajo la dirección del Dr. Baltasar Fernández Manjón y del Dr. Pablo Moreno Ger en el Departamento de Ingeniería del Software e Inteligencia Artificial (DISIA), y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Fdo.: Elisa Sanz Troyano

Agradecimientos

Este proyecto culmina el trabajo llevado a cabo en este curso académico y puesto que es un estudio extenso y exhaustivo, he recibido ayuda y apoyo de varias personas y grupos. Por ello agradezco su apoyo y colaboración a todo el que ha dedicado algo de su tiempo a orientarme.

A la hora de realizar este proyecto, agradezco a mis dos directores de proyecto, a Baltasar Fernández Manjón y a Pablo Moreno Ger, las oportunidades que me han brindado y su orientación a la hora de realizar mi trabajo y asimismo, la confianza depositada en mí para llevar a cabo este proyecto.

También, me gustaría resaltar la ayuda que he recibido de mi grupo de investigación <e-ucm>, gracias al cual he aprendido a trabajar en grupo, proponiendo ideas y aceptando consejos, y en definitiva, cooperando unos con otros.

Un agradecimiento especial a Javier Torrente Vigil, compañero de grupo de trabajo, que me ha orientado desde que empecé a formar parte de este grupo, y que me ha supervisado en todos los trabajos realizados.

En cuanto a la empresa Technosite, junto con la que hemos trabajado, me gustaría agracederles su orientación, consejo y guías en un domino del que ellos son especialistas.

Por último, me gustaría resaltar la ayuda que me han brindado todos mis compañeros y familiares que han participado de alguna manera en este trabajo, tanto de forma logística como la psicológica.

Acknowledgments

This project completes the work carried out during this academic year and since it is an extensive and exhaustive study, I have received help and support from various groups and people. Hence, I thank to everyone who has spent some time to orient me, for their support and collaboration.

When realizing this project, I thank my two project managers, Baltasar Fernández Manjón and Pablo Moreno Ger, the opportunities they have given to me and his guidance to develop my work and also the trust they have posed on me to carry out this project.

Also, I would like to highlight the help I have received from my research group <e-ucm>, thanks to which I have learned to work together, proposing ideas and accepting advice, and ultimately, to cooperate with each other.

Special thanks to Javier Torrente Vigil, co-working mate, which has guided me since I joined this group, and who has supervised me in all the work realized.

As the company Technosite, with which we have co-operated, I would like to thank them their orientations, advice and guidance in a domain where they are specialists.

Finally, I would like to highlight the support that all my friends and family members have given to me, who have participated in any way in this work, both logistical and psychological form.

Palabras clave

- Accesibilidad
- <E-Adventure>
- E-Learning
- Educación
- Herramientas de autoría de juegos
- Aprendizaje basado en juegos
- Aprendizaje personalizado
- Metodología
- Videojuegos

Keywords

- Accessibility
- <E-Adventure>
- E-learning
- Education
- Game authoring tools
- Game-based learning
- Personal adapted learning
- Methodology
- Videogames

Resumen

En la última década, la generalización de las nuevas tecnologías está cambiando todos los aspectos de nuestra vida, ya sea en el campo del trabajo, de las relaciones sociales, del comercio, de la educación, etc.

Debido a la introducción de estas nuevas tecnologías, surgen problemas de accesibilidad que antes no existían o que ya estaban lo suficientemente abordados como para que no presentasen una barrera.

Centrándonos en el campo educativo, estas barreras pueden poner en peligro el derecho universal a la educación para estudiantes con discapacidades. Los principios del denominado "diseño para todos" o diseño universal, tienen como objetivo el diseño de productos y entornos de fácil uso para el mayor número posible de personas, sin la necesidad de adaptarlos o rediseñarlos de forma especial. El diseño universal, así pues, beneficia a todas las personas, sean cuales sean su edad y/o habilidades.

Por otra parte, en el ámbito educativo, es recomendable ofrecer a los alumnos contenidos más dinámicos e interactivos y es por ello que los videojuegos se han propuesto para promover el desarrollo de nuevas habilidades, aunque esto suponga un gran reto desde el punto de vista de la accesibilidad.

En este trabajo se discuten los principales retos en la introducción de la accesibilidad en videojuegos educativos y cómo se están abordando desde la herramienta de autoría de juegos educativos <e-Adventure>. Este tipo de herramientas son equiparables a herramientas de diseño y retoque fotográfico, en el sentido de que son herramientas que requieren una alta interacción visual por parte del usuario para conseguir el cometido final. Por esta razón, es necesario que el desarrollo de herramientas en este campo tenga en cuenta, aún más si cabe, características de accesibilidad que minimicen los problemas de tratar con contenido multimedia o imágenes, por parte de personas con discapacidad, ya sea visual, auditiva o motora.

Nuestro objetivo primordial es desarrollar una metodología general para el desarrollo accesible y el buen diseño de herramientas de autoría de juegos. Para ello se estudian las características de accesibilidad que deberían introducir estas herramientas, se estudian metodologías existentes y se presentan varias pautas para conseguir hacer accesible cualquier herramienta de autoría genérica.

Posteriormente, nos centramos específicamente en la plataforma <e-Adventure> con el fin de contrastar la metodología propuesta sobre ella y ejemplificar posibles soluciones. Nuestro propósito es convertir <e-Adventure> en una solución integral para el desarrollo de juegos educativos accesibles teniendo en cuenta tanto la accesibilidad de los juegos desarrollados como la herramienta de autoría en sí.

Abstract

In the last decade, the introduction of new technologies has been changing all aspects of our lives, changing professional activities, social relations, trade, education, etc. The introduction of most these new technologies poses new accessibility issues that increase the barriers for disabled people.

Focusing on the educational field, these barriers may put the universal right to education for students with disabilities at risk. The principles of the universal design, also called “design for everyone”, propose the design of user-friendly products and environments for as many people as possible, without the need to adapt or redesign them. Universal design, therefore, benefits all people, whatever their age and / or their skills.

Moreover, in the educational field, there is a growing trend towards giving students more dynamic and interactive content. That is the reason why video games have been proposed to promote the development of new skills, although this represents a major challenge from an accessibility perspective.

This work discusses the main challenges in the introduction of accessibility in educational games and how they are being tackled in the <e-Adventure> authoring tool for educational games.

<e-Adventure> is similar to design and photo editing tools, in the sense that they are tools that require high visual interaction from the user to achieve the final task. For this reason, it is necessary that these tools keep in mind even more the challenge of accessibility for visual disabled when dealing with media or images.

Our main goal is to develop a general methodology that includes a series of guidelines to follow for developing accessible authoring tools for games.

To achieve this goal, we have studied the accessibility features these tools should introduce and compared them with existing methodologies. Finally, we present several guidelines to evaluate and improve the accessibility of a generic authoring tool.

We focus especially on the <e-Adventure> platform to check the proposed methodology and to illustrate possible solutions to the identified issues. Our purpose is to turn <e-Adventure> into a comprehensive solution for the development of accessible educational games taking into account not only the accessibility of developed games but also the authoring tool itself.

Table of Contents

PREFACE

CHAPTERS

Index of Tables

Index of Figures

1. INTRODUCTION

In the 16th and 17th centuries, education was addressed primarily to the wealthy classes. Later in the period known as the Illustration[1], it was when education was directed to everyone. It means that Illustration was the period where society started to be aware of universal education rights.

National education systems came in Europe in the early 19th century following the French Revolution. In Spain, the 1812 Constitution incorporated the idea of education as a network in whose organization, financing and control, the state had to take part. So, these were the foundation for the establishment of the Spanish educational system.

This foundation includes the defense of universal primary education for all people without exception and uniformity of teaching plans for the entire State. As our illustrated antecessors, we share with them the faith in public education as the main instrument of renewal and reform, and also their beliefs of democracy and the education needs for freedom.

Since we are in the 21st century and society has evolved since the current educational systems were developed, there is a growing demand to open new lines of instruction in the educational field, in addition to those presents, which are gradually being implemented. During the last years new technologies and media supporting new pedagogical strategies have been proposed for education in order to prepare students for the challenges that our changing society demands (Schrier, 2005). These new teaching paradigms, that are taking shape as part of the information society, provide more dynamic and interactive content for students.

For instance, the progressive introduction of ICT (Information and Communication Technology) provides access to major kinds of information, always available everywhere, through the use of the new features of computers, telephones, the mass media and especially the Internet. The use of ICT favors particularly the development of continuing education (a.k.a. further education), providing tools that enable the creation of virtual learning environments, free from the restrictions of classroom instruction, and adaptable to individual characteristics of each student.

Therefore, educators need to prepare not only themselves for their new role but also their students for this new scenario. Both, need to enhance in an early stage, their skills in using technological tools (Fernández-Manjón et al., 2009).

The starting point is undoubtedly the incorporation of ICT to education. This incorporation happens, firstly, by the allocation of equipment and the necessary infrastructure in schools, and secondly, by including these new technologies in the subject calendar so as to provide the needed time to use them.

Therefore, once the installation and use have been accepted, the approach of using video games arises as a possibility, following the trend of including more dynamic and interactive content for students (Van Eck, 2006). Videogames have been pointed out because of their potential as learning tools (P. Moreno-Ger *et al.*, 2009; Squire, 2003) leading the students to deep states of concentration usually called “Flow states” in Psychology (Csikszentmihalyi, 1991).

We also highlight that educational video games allow users to take new roles and to learn new skills that the characters they are controlling are carrying out. Some of these

[1] http://www.historiasiglo20.org/HE/8f.htm (Retrived June 21, 2010)

skills they would not be able to develop otherwise. Students have new requirements and work expectations, which is why a greater student participation in learning through active investigation and experimentation methods is advisable.

The other part of this set is composed by teachers who can free themselves from their more traditional tasks, through the use of these new technologies. New teaching standards, which take into account new technologies and media to enhance learning, mean new roles for teachers: It is not the teacher who must provide all the information.

With these new standards, teachers are liberated from routine work and can make truly professional and creative guidance of pupils. This work involves counseling, guidance and wide participation in planning and production of audiovisual materials to suit the needs of groups or individuals to whom they are intended.

Nonetheless, the application of video games in educational settings is very broad and diverse, including experiences where students interact with the video games but also other experiences where not only teachers but also students create their own video games (Overmars, 2004).

As the academic interest in using video games as educational materials grows, several proposals have emerged to facilitate the creation of educational video games and simplify their application in educational settings. One of the proposals is use of authoring tools, since they reduce the production costs of video games and facilitate the game development by people who do not have a technical background (J. Torrente et al., 2008). In this manner video games are brought closer to the educational system as teachers and students can develop their own games.

In the next sub-sections, we discuss the different accessibility issues and their current state of the art. Then, we present the main goals of our work and how the identified issues have been addressed.

1.1 Accessibility issues

Our aim is to achieve that video games and authoring tools can be used both by teachers and students with no exception. To achieve this, it is necessary to be aware of requirements that it would be essential to develop in order to consider accessible approaches that can be used by everyone. Only in this way, we will fulfill the requirement of universal education.

However, the accessibility of video games is an open issue, not only for commercial video games (where the industry does not favor large investments to incorporate accessibility), but also for educational games where the budgets are more limited and it is necessary to adjust them.

Whereas accessibility in video games is slowly starting to be considered within game authoring tools, there are not widely accepted guidelines that standardize how to develop accessible highly visual software.

This problem becomes more evident when games are introduced in e-Learning environments, where the accessibility of traditional Web content is reasonably

covered by initiatives that facilitate access to the Web, such as WAI, (Magerko, 2008) or the use of assistive technology.

On the other hand, focusing on the authoring tool field, the challenge is easier in text-based authoring tools, such as Microsoft Word ™ or PowerPoint ™. These tools are not as visuals as the ones which address the creation and edition of video games or design and photo editing tools such as Adobe Photoshop™.

As a conclusion, we have depicted the open research lines that compose our wide scope to give an idea of the current accessibility issues. In the next section we address the goals in which we focus our research work to clarify our specific approach within the wide field we have defined.

1.2 Objectives

Once we have argued what is already covered and which are the fields that are necessary to go on developing, we clarify the scope of this work and the approach we follow.

The approach followed in this work is closer to the "gaming" point of view than to any other. Thus, our aim is to provide a holistic approach, creating an accessible video game authoring tool that produces accessible games without requiring a significant increase in the development cost.

Therefore, the purposes of this thesis study are:

- Analyze the challenges posed by the introduction of accessibility in game authoring environments and educational video games.
- Develop a general methodology to establish a series of guidelines that test whether the authoring tool that we are checking can be used by a disabled person.
- From this analysis we present how these problems are being addressed in the <e-Adventure> authoring tool.

1.3 Structure

This master thesis covers different topics, and in order to make clearer we depict its structure in this section. It has been organized in the following chapters:

Chapter 1, titled as Introduction, describes the context in which this master thesis is developed, and also sets the main goals for this work.

In chapter 2 we make a review of the current state of the art starting by studying the types of disabilities, regulations and guidelines, current technologies, accessible games and accessible authoring tools.

Next, in chapter 3, we propose our evaluation model by studying the needed requirements and their applicability in various tools.

Chapter 4 is an introduction to the <e-Adventure> platform, describing both the video games and the video game authoring tools, and also a section addressed to study the accessibility in the video games developed with this platform.

Chapter 5 describes the solutions we are carrying out in the <e-Adventure> editor and exemplifies several of them.

To finalize, in chapter 6 we talk about the contributions and future work.

1.4 Acknowledgements

This work is promoted within the GAMETEL project (Spanish Ministry of Industry, Tourism & Trade, "Avanza" Program I+D: TSI-020110-2009-170), which aims to develop a software system that allows to design, develop, implement and evaluate highly interactive educational content such as video games and simulation environments, to adapt and customize for specific needs of each user and execution context.

Within this project the <e-UCM>[2] research group, where this work has been developed, is focused on the development of adaptive and accessible educational point-and-click adventure games (Javier Torrente et al., 2009) that can be integrated into e-Learning systems.

[2] http://www.e-ucm.es/

2. STATE OF THE ART

In this first section we present a state of the art in accessibility focused especially in video games and video game authoring tools.

We focus our work in addressing accessibility in video game authoring tools as a special case of accessibility in visual desktop authoring tools.

Our study mainly addresses the software solutions for accessibility. A review of the hardware and special devices for accessibility is outside the scope of this work and this kind of review can be found at (Argue et al., 2004; Kearney, 2005; Sánchez et al., 2009; Wang et al., 2009).

The implementation of accessibility requirements, in addition to allowing and improving access for people with disabilities to content, also carries additional advantages that are presented below.

Based on section 2.1, the group with mobility issues is the largest, followed by auditory, visual and learning disabilities, as we can observe in Table 2 (section 2.1), so, focusing on this kind of disabilities more commons, we introduce our motivations to improve the accessibility of games and advantages of developing it (K Bierre et al., 2004).

Some are related to the user's quality of life, while others are more important to the publishers and developers of games. Next, there are shown some of them[3]:

- User Satisfaction: Gamers play games for entertainment, not to experience a sense of frustration.
- Simplifies development: Certain conditions and technical requirements, which accessibility recommends, should result improvements in development processes, what will save costs.
- Increases the target public: As improving access to content in general, we are facilitating the use to disabled and non-disabled people, which increases the possible users.

Therefore accessibility should be viewed not as an isolated set of requirements for a particular group, but as options for improving the quality, what will bring benefits and better background for future technologies.

Once we have seen the motivations to carry out this work, we present the structure followed in this second chapter.

Therefore, in section 2.1 we introduce both, a study of the types of gaming disabilities and a statistical survey to clarify the percentage of population that present any kind of disability.

Section 2.2 presents the existing regulations in terms of laws and the existing guidelines that tackle these topics.

[3] http://archives.igda.org/accessibility/IGDA_Accessibility_WhitePaper.pdf

In section 2.3, we address the most common assistive tools as well as several developed libraries which attempt to make easier the job carried out by these assistive tools.

Section 2.4, addresses a review of the current state of accessibility in video games, whilst section 2.5 focuses on the accessibility in desktop authoring tools.

Finally, we present our conclusions and we introduce chapter 3.

2.1 Gaming Disabilities

As accessibility is a very broad term that comprises very different aspects, systems and situations, we consider necessary to focus on a specific definition.

According to the World Health Organization WHO/OMS in the "International Classification of Impairments, Disabilities and Handicaps - ICIDH[4]", the definition of disability is the next one: "In the context of health experience, a disability is any restriction or lack (resulting from an impairment) of ability to perform an activity in the manner or within the range considered normal for a human being".
Thus, a disability is concerned with functional performance or activity, affecting the whole person.
There are a wide range of disabilities that limit a person who is attempting to play a game and also various ways to sort them.

According to the WHO and following its ICIDH (International Classification of Impairments, Disabilities and Handicaps[5]) manual of classification relating to the consequences of disease they focus in health and performance rather than in disabilities, since it is assumed that everyone can experience in a particular time of its life, a health deterioration. Disability is not only a feature of any social group but also is a universal experience.

Hence, according to these concepts, they base their classification in a holistic model of performance, disabilities and health. It is composed of three essential components: bodily functions and structures (referred to physiological and psychological functions), the activity (referred to individual execution of tasks and the limitations to realize them) and the participation, referred to social situations where the problems are called constraints.

Following this classification, the ICIDH is composed by 1424 categories which cover the human experience, so that, you can assort a problem into a deterioration, a limitation or a constraint.

2.1.1 Types of Gaming Disabilities

Focusing in our scope, we have addressed a brief classification of the most relevant gaming disabilities according the IGDA survey[6].

[4] http://www.aihw.gov.au/publications/dis/dda-mnc/dda-mnc-c03.pdf

[5] http://www.who.int/classifications/icf/en/

[6] http://archives.igda.org/accessibility/IGDA_Accessibility_WhitePaper.pdf

According to them, game accessibility can be defined as the ability to play a game even when functioning under limiting conditions which can be functional limitations, or disabilities. Below, we present a review of these disabilities:

Visual Disabilities:

- Blindness: Is usually defined as "the loss of vision not correctable with lenses". These people cannot play games that rely on visual cues to prompt a player, instead they must rely on sounds or especial hardware to indicate them when they need to act.
- Low Vision: Comprises any vision loss that adversely affects the performance of daily activities. A person with low blindness could detect light, perhaps see some motion, but not much else. These users could play games with both any kind of screen magnification and any sound.
- Color Blindness: Is the inability to detect some colors. It ranges from total color blindness (only can perceive different gray shades), to lighter types, where a person cannot perceive clearly the difference between red and green or yellow and blue.

Auditory Disabilities:

- Deafness: is the inability to understand speech or recognize environmental sounds. Deaf people generally use the sign language to communicate as their first learned language, even before learning their mother tongue.
- Hard of hearing: There are different kinds of hearing loss, so that a person could have mild, moderate, severe or profound hearing loss. These people may need to turn the volume of their speakers in order to achieve useful auditory cues.

Mobility Disabilities:

- Paralysis: Paralysis could occur due to an accident, birth defects or disease. In paralysis, the nerves that control the body's voluntary muscles are no longer signaling those muscles. As a result of paralysis, certain types of games requiring excellent hand/eye coordination or the ability to rapidly press a button ("twitch" games) are not really accessible. Other types of games such as turn based strategy games may be more suitable, assuming they could work with adaptive hardware.
- Neurological Disorders: Certain neurological disorders can cause mobility issues. Due to problems with transmitting impulses to muscles, people with neurological disorders also suffer many of the same issues affecting paralysis victims.
- Lack of Mobility: One of the unfortunate aspects of aging is the gradual loss of flexibility in joints and difficulties moving as fast /well as one used to. Degenerative diseases like arthritis become more common. Games requiring the player to participate with their whole body, such as Dance, Dance Revolution, may not be appealing or physically possible. Even controllers can cause problems for an increasingly older population. With slower reflexes, "twitch" games become more

difficult. A game that required a lot of mouse motion and rapid, accurate clicking on the screen will probably not appeal to older gamers.

Cognitive Disabilities:

- Memory Loss: The inability to retain information can make certain types of games more difficult. An action game with a complex storyline or an adventure game with a complex map can be difficult for people who have memory problems. Providing built in maps or journaling can make the gaming experience more enjoyable.
- Attention Deficit Disorder: There is currently a lot of discussion and research going on regarding Attention Deficit Disorder (ADD) and Attention Deficit Hyperactive Disorder (ADHD). A game that requires prolonged concentration for long periods, may not interest a person with ADD or ADHD. However, games such as first person shooters have the appeal of constant change, immediate feedback, and the ability start and stop whenever the user desires.
- Dyslexia: Dyslexia is considered a learning disability. His definition is having difficulty with words in reading, spelling and writing in spite of having normal intelligence and ability. It is considered a problem with processing information. People with dyslexia will often confuse similar letters (for example “b” and “d”), reverse words (“now” seen as “won”), or transpose words in sentence. Because reading is difficult for them, those with dyslexia may not be interested in a game that provided most of its feedback and instructional information in text.

According to our scope and according to the IGDA survey, we have addressed the mainly gaming disabilities in order to clarify which are the needs. Nonetheless, is this work, we focus especially in visual disabilities since they are the major problem to solve in this kind of tools.

In order to provide a reason why we are tackling this topic, next, we deal with the statistic approach, by making a review of two surveys, one based on the EEUU and the other based on Spain.

2.1.2 Disabilities Survey

As we have said above, a disability is any restriction or lack (resulting from an impairment) of ability to perform an activity in the manner or within the range considered normal for a human being.

Only in USA in the year 1994 there was 3,85 millions of people with any kind of disability.

In Table 1, it is shown the percentages by age and gender[7], where we can see there is a total of 15% of the United States population with any kind of disability.

[7] http://unstats.un.org/unsd/demographic/sconcerns/disability/disform.asp?studyid=175

On the other hand, it is also represented the percentage of people between 15 and 59 years, over all the population, that presents any disability, as well as the percentage of people who has more than 59 years.

United States 1994 Survey			
All areas	**Total**	**15-59**	**60 +**
Total	15.0	11.9	38.2
Male	14.4	11.8	36.9
Female	15.7	12.0	39.1

Source: United States Department of Health and Human Services, National Center for Health Statistics, Vital and Health Statistics, Current Estimates from the National Health Interview Survey, 1994, Series 10, No. 193 (1995).

Table 1. Percentage of persons with disability by age and sex in United States.

Next, focusing in Spain, we give an overview of the presence of disabilities in the Spanish population[8].

According to the Survey on Disabilities, Impairments and Health Status (EDDES) Act 1999, the official percentage of people with disabilities in Spain is 9%, representing approximately three and half million people. The survey uses a concept of disability based on the International Classification of Disabilities and Impairments[9].

This disability is primarily female. The absolute number of women with disabilities (2.05 million) is much higher than that of men (1.47 million) so that 58.3% of people with disabilities are women. This high rate is explained, first, because there are more women than men in the total count of population, but especially because from the maturity and especially at older ages there are more women than men due to the male mortality.

Before age 50, women are less likely than men to have disabilities, but in the older age female rates far outweigh the male, so that in the 80 and over 65.7% of women have disability, compared to 59.2% of men.

In the next graphic is shown the total percentages of men and women with or without disabilities.

[8] http://www.ine.es/prodyser/pubweb/disc_inf05/discapa_inf.htm

[9] http://www.ine.es/prodyser/pubweb/disc_inf05/disca_ig_cap2.pdf

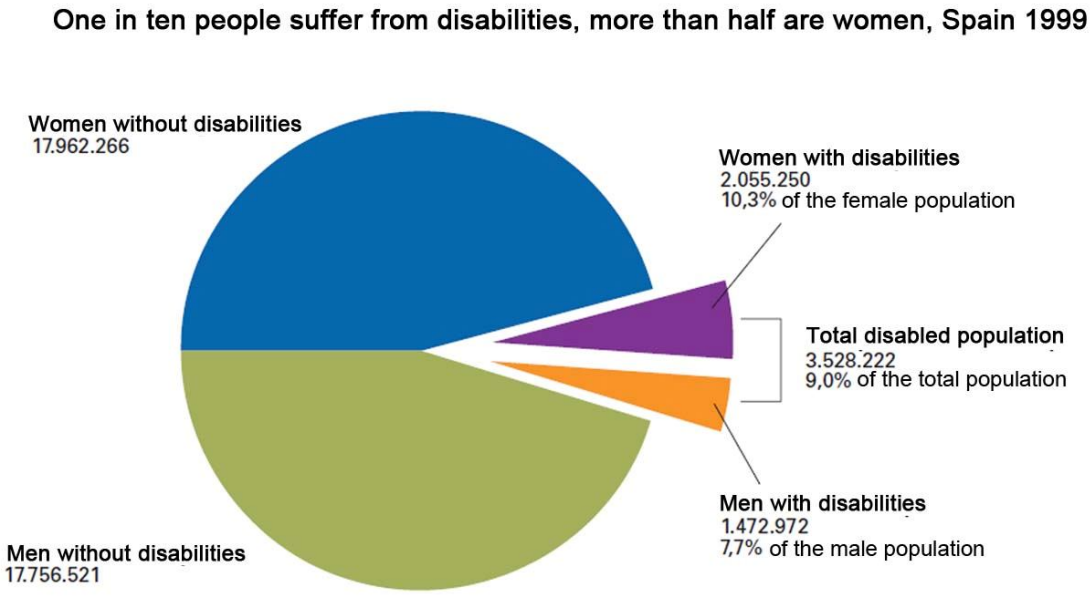
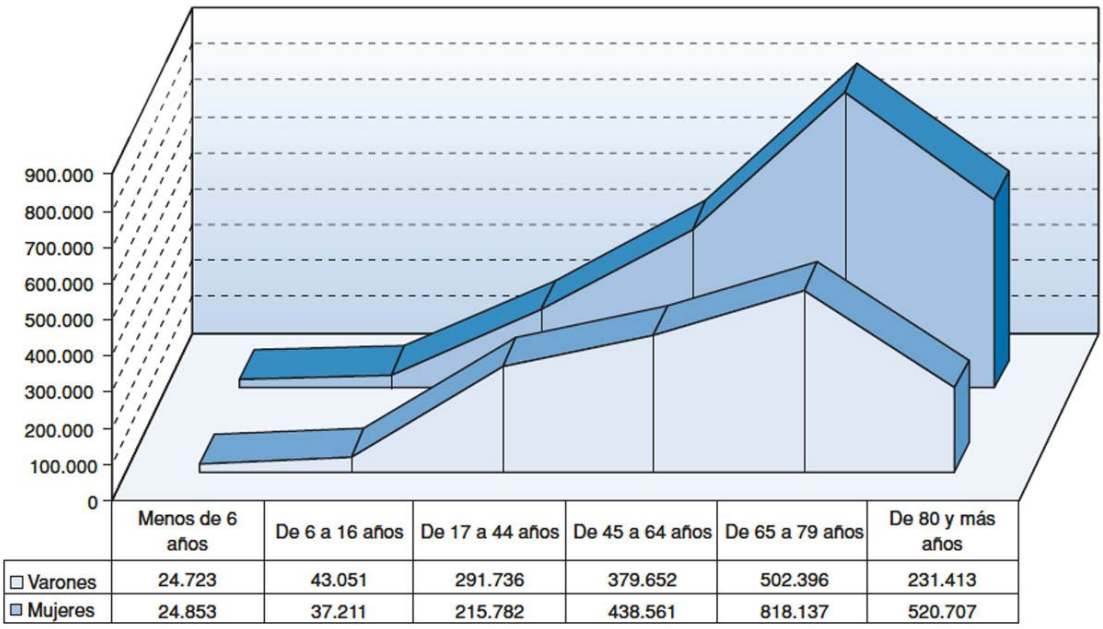


Figure 1. Disability in Spain

In this other graphic, we can see de disability rates by age and gender, making sets of 100.000 people. Thus, we can see in an easier way how the progression becomes bigger between women and men according as the age is greater. The higher point it is reached for women between 65 years and 80 years, where the female population is superior to male one.



	Menos de 6 años	De 6 a 16 años	De 17 a 44 años	De 45 a 64 años	De 65 a 79 años	De 80 y más años
Varones	24.723	43.051	291.736	379.652	502.396	231.413
Mujeres	24.853	37.211	215.782	438.561	818.137	520.707

Fuente: INE. Encuesta sobre Discapacidades, Deficiencias y Estado de la Salud, 1999

Figure 2. Evolution of disability by age group and sex. Rate per thousand inhabitants in each age group and sex, 1999

Thanks to this graphic, it is easier to check that disability is not a term referred to a close social group, but is rather a social question which affects everyone when becoming elderly.

And to finish, we show in Table 2 the percentage of people who have any kind of disability grouped by age, making three groups, and by disability group.

	6-64 years	65-79 years	80 years and over
Mental disabilities	16,68	6,33	8,59
Visual disabilities	15,39	18,24	18,03
Hearing impairment	16,25	18,14	16,44
Language, speech and voice shortcomings	1,17	1,05	0,9
Motor disabilities	32,77	32,37	22,43
Nervous system deficiencies	8,04	5,9	5,31
Visceral deficiencies	6,96	8,2	4,8
Other deficiencies	2,75	9,78	23,5
TOTAL	100	100	100

Table 2. Distribution of deficiencies by group of deficiency and age, 1999.

With this Table we emphasize in the fact that the higher group with a kind of deficiency is the motor one in every age group, followed by hearing and visual impairments. That is one of the reasons why our study is mainly addressed for these kinds of deficiencies.

In next section we study existing regulations and existing guidelines to make a review of the current state in these important fields.

2.2 Regulations and Guidelines

In this section we address the main regulations issues and the various methodologies we have studied. We highlight the Regulation section because the more the state focuses its efforts on introducing new laws, or just approaches, for build up a society more accessible in any way of our life, the more steps forward in this field will be done, increasing the awareness in this scope. We think that one of the firsts steps to achieve that a society allows the integration of everyone is firstly, make aware everyone who compose that society, of the problems and needs that disabled people may find out to realize common tasks that society in general has not thought they could mean a barrier for anyone.

Focusing on the computer field, tasks as easy as checking the mail, filling in a form, access into software and so on, could not be realized by disabled people if they were not adapted in some way. Thus, we see in these next two section current regulations and guidelines.

2.2.1 Regulations

Several countries have regulations regarding the accessibility of products for the disabled community. The United States, created the Standards of Quality on the Electronics and Computer Technology, which were published in the Federal Register on December 21, 2000 in the USA. The quality standards of the Committee on Access are the only standards related to accessibility in technology that have a legal basis under Section 508 of the Rehabilitation Act. Every software prototype (any equipment used to transmit, receive, use or store the information) must meet the requirements of Section 508 or are excluded from the process of selecting contracts for all government agencies. This means that every software tool used by the administration must be accessible.

In addition to the US regulations, The United Nations has proposed "Standard Rules on the Equalization of Opportunities for Persons with Disabilities". Rule 10 relates to accessibility[10]:

- States will ensure that persons with disabilities are integrated into and can participate in cultural activities on an equal basis.
- States should ensure that persons with disabilities have the opportunity to utilize their creative, artistic and intellectual potential, not only for their own benefit, but also for the enrichment of their community, be they in urban or rural areas. Examples of such activities are dance, music, literature, theatre, plastic arts, painting and sculpture. Particularly in developing countries, emphasis should be placed on traditional and contemporary art forms, such as puppetry, recitation and story-telling.
- States should promote the accessibility to and availability of places for cultural performances and services, such as theatres, museums, cinemas and libraries, to persons with disabilities.
- States should initiate the development and use of special technical arrangements to make literature, films and theatre accessible to persons with disabilities.

Therefore, focusing on computer scope, it would not be surprising to see more stringent regulations becoming law in the future. By planning for accessibility in new games, we can be prepared for these regulations.

The long-term implications of applying these laws at the federal level are the large organizations and private corporations start to institute similar internal regulations for protection against lawsuits and to comply with the standards of selection of contracts.

[10] http://unstats.un.org/unsd/disability/rules.txt

In this manner, these organizations can reach to a larger public and also they are prepared for a possible co-operation with the public administration, otherwise it would not be possible.

2.2.2 Guidelines

Whereas the field of Web accessibility is fairly covered with leading initiatives such as the WCAG or the WAI[11] initiative, which develops a series of accessibility standards and guidelines that are introduced in Essential Components of Web Accessibility, for other types of technologies have not been proposed widely accepted guidelines on accessibility. We have highlighted the IBM checklist as the most complete for our scope, because it focuses on how to achieve the accessibility through the Java language. That is why we address this guideline as the most useful for our goal because the authoring tool <e-Adventure> in which we develop the necessary features in section 5, uses the Java language.

Hence, in this section we analyze the major accessibility initiatives for authoring tools, even when any of them, such as ATAG initiative, is focus on Web initiatives.

2.2.2.1 Web Content Accessibility Guidelines (WCAG 2.0) [12]

Web Content Accessibility Guidelines 2.0 covers a wide range of recommendations for making Web content more accessible. Following these guidelines will make content accessible to a wider range of people with disabilities, including blindness and low vision, deafness and hearing loss, learning disabilities, cognitive limitations, limited movement, speech disabilities, photosensitivity and combinations of these. Following these guidelines will also often make your Web content more usable to users in general.

It addresses how to develop the information in an accessible way, in a Web site, including text, images, forms, sounds, and such.

WCAG is primarily intended for:

- Web content developers (page authors, site designers, etc.).
- Web authoring tool developers.
- Web accessibility evaluation tool developers.
- Others who want or need a technical standard for Web accessibility.

Thus, as we have said, we highlight this methodology because is made to help Web developers to create accessible contents. It is composed by 14 guidelines which describe how make a good design of these Web contents to make them as accessible as possible.

[11] http://www.w3.org/WAI/

[12] http://www.w3.org/TR/WCAG10/

2.2.2.2 Accessible Rich Internet Applications (WAI-ARIA)[13]

WAI-ARIA, the Accessible Rich Internet Applications Suite, defines a way to make Web content and Web applications more accessible to people with disabilities. It especially helps with dynamic content and advanced user interface controls developed with Ajax, HTML, JavaScript, and related technologies.

Currently certain functionality used in Web sites is not available to some users with disabilities, especially people who rely on screen readers and people who cannot use a mouse.

WAI-ARIA addresses these accessibility challenges, for example, by defining new ways for functionality to be provided to assistive technology. With WAI-ARIA, developers can make advanced Web applications accessible and usable to people with disabilities.

From this methodology we can extract information related with assistive tools and how to establish a god interaction with them.

2.2.2.3 User Agent Accessibility Guidelines (UAAG)[14]

The User Agent Accessibility Guidelines (UAAG) documents explain how to make user agents accessible to people with disabilities, particularly to increase accessibility to Web content. It is primarily for developers of Web browsers, media players, assistive technologies, and other user agents.

For instance:

- People who want to choose user agents that are more accessible can use UAAG to evaluate user agents.
- People who want to encourage their existing user agent developer to improve accessibility in future versions can refer the user agent vendor to UAAG.

This methodology provides guidelines for designing user agents that lower barriers to Web accessibility for people with disabilities. User agents include browsers and other types of software that retrieve and render Web content. A user agent that conforms to these guidelines will promote accessibility through its own user interface and through other internal facilities, including its ability to communicate with other technologies (especially assistive technologies). Furthermore, all users, not just users with disabilities, should find conforming user agents to be more usable.

[13] http://www.w3.org/WAI/intro/aria

[14] http://www.w3.org/WAI/intro/uaag.html

2.2.2.4 Authoring Tool Accessibility Guidelines (ATAG 1.0)[15]

It addresses how to develop software that creates Web sites. Its purpose is to assist developers in designing authoring tools that produce accessible Web content and to assist developers in creating an accessible authoring interface.

They show how to make authoring tools accessible to persons with disabilities. These software tools are used to create Web pages and content. One of the main purposes of ATAG is to define the way in which the tools help Web developers to produce Web content.

We focus in these guidelines because they give us the requirements to make a software application accessible, even though it's focused on Web content, we highlight the guidelines 3, 5 and 7 because they are more general and we can apply them in our scope.

- **Guideline 3: Support the creation of accessible content**

  Well-structured information and equivalent alternative information are the main points of accessible design, allowing information to be presented in a way most appropriate for the needs of the user without constraining the creativity of the author.

  Producing equivalent information, such as text alternatives for images and auditory descriptions of video, can be one of the most challenging aspects of Web design and design in general, and authoring tool developers should attempt to facilitate and automate the mechanics of this process. For example, prompting authors to include equivalent alternative information such as text equivalents, captions, and auditory descriptions at appropriate times can greatly ease the burden for authors.

- **Guideline 5: Integrate accessibility solutions into the overall "look and feel"**

  When a new feature is added to an existing software tool without proper integration, the result is often an obvious discontinuity. Differing color schemes, fonts, interaction styles, and even software stability can be factors affecting author acceptance of the new feature. In addition, the relative prominence of different ways to accomplish the same task can influence which one the author chooses. Therefore, it is important that creating accessible content be a natural process when using an authoring tool.

- **Guideline 7: Ensure that the authoring tool is accessible to authors with disabilities**

  The authoring tool is a software program with standard user interface elements and as such must be designed according to relevant user interface accessibility guidelines. When custom interface components are created, it is essential that they be accessible through the standard access mechanisms for the relevant platform so that assistive technologies can be used with them.

[15] http://www.w3.org/TR/WAI-AUTOOLS/

Some additional user interface design considerations apply specifically to authoring tools. For instance, authoring tools must ensure that the author can edit (in an editing view) using one set of stylistic preferences and publish using different styles. Authors with low vision may need large text when editing but want to publish with a smaller default text size. The style preferences of the editing view must not affect the markup of the published document.

Authoring tools must also ensure that the author can navigate a document efficiently while editing, regardless of disability. Authors who use screen readers, refreshable braille displays, or screen magnifiers can make limited use of graphical artifacts that communicate the structure of the document and act as signposts when traversing it. Authors who cannot use a mouse (e.g., people with physical disabilities or who are blind) must use the slow and tiring process of moving one step at a time through the document to access the desired content, unless more efficient navigation methods are available. Authoring tools should therefore provide an editing view that conveys a sense of the overall structure and allows structured navigation.

2.2.2.5 IBM Software checklist[16]

We focus on this checklist, because unlike the ones we have seen above, this one addresses accessibility issues in software desktop tools, and also shows this topic by using the Java language, which is even more interesting in our scope because the <e-Adventure> authoring tool, in which we will exemplify our approaches, is developed in this language.

When writing a Java application, it is advisable think about the fact that users with varying abilities want to use that product. There will be users who cannot use their hands, and users who are blind. That means they may not be able to use a mouse, type on a keyboard, see a screen, or read a printed user's guide.

Over the years, disabled users and Assistive Technology Vendors (ATVs) have invented incredibly clever work-around for these problems. Voice input replaced the mouse, mouth sticks and switches replaced the keyboard, text-to-speech synthesis replaced the screen, and online books replaced printed books.

Some products include unintentional roadblocks to people who have disabilities. Some examples:

- Programs that provide only mouse input (impossible for people who are blind or who use voice control. This also applies to hardware that does not have a mouse).
- Programs that use audio-only messages (individuals who are deaf cannot hear the messages. This also applies to people with systems that do not have sound capability).
- Operating systems (OS) that define Ctrl + Alt + Del to reboot the system, since it is impossible to do this with only one hand.

[16] http://www-03.ibm.com/able/guidelines/java/snsjavag.html#2.0

These IBM guidelines are interesting to us due to our software tool is developed in Java and they show not only a general way of developing software tools but also they focus on how to face them by using Java utilities.

Hence, we highlight several checkpoints which should be enabled in any software application, like making the application keyboard accessible, providing a logical component layout, give the needed object information to achieve a good interaction with assistive tools and so on.

2.2.2.6 IGDA Accessibility in Games[17]

We have also focused on this guideline because is especially addressed to meet the accessibility needs in video games, which is the scope that concerns us. It starts by giving a complete state of the art to focus in several video games examples that include various accessibility features. They also make a survey targeted to game developers industries, to test the extent of accessibility that videogames present. Finally they also analyze the various approaches that can be used in order to incorporate accessibility in games.

2.3 Assistive technologies

There are a number of assistive technology solutions available today being used for general computer access that could also being used with games. Some are designed for disabilities but not for games while others are designed for games but not disabilities. The following list highlights some types of assistive technologies that are currently used, being the first of them, a hardware solution and the rest software:

- Alternative Pointing Devices: These devices allow individuals to control their computers through other devices than the standard pointing ones.

  As example, we include head and eye-tracking systems, specialized joysticks and even systems that use a combination of body, eye and brain signals for control.
- On-Screen Keyboards: This tool is used by people who are unable to use a standard keyboard. An on-screen keyboard using a pointing method such as pointing devices, switches or Morse-code input systems. It is usually difficult to perform complicated key sequences quickly using an on-screen keyboard, so keyboard intensive games can be very hard if not impossible to play using this type of technology.
- Speech Recognition: These programs are primarily used by people with mobility impairments. These utilities enable people to control computers with their voice instead of a mouse or keyboard. Speech recognition can be used for text entry, mapping commands to keyboard macros with a system, and verbally to position a mouse cursor.
- Screen Magnifiers: This type of software helps people with low vision by allowing them to zoom in or enlarge portions of the computer screen so they are easier to

[17] http://archives.igda.org/accessibility/IGDA_Accessibility_WhitePaper.pdf

see. This works well with applications that tend to display static data such as text or diagrams. Screen magnifiers may have problems working with games that are animation intensive or use full-screen modes.

- Screen Readers: These utilities are primarily useful for people who are blind. These aids make on-screen information available as synthesized speech or a refreshable Braille display. They can only translate text-based information and are often used with text intensive programs such as web browsers, email, and document viewers.

  The screen reader software most recognized and used by people with some degree of visual impairment is Jaws[18], a screen reader speech synthesis, which allows users with limited visual, scroll through all the areas listed on the screen.

  The arriving of graphical user interface (GUI) has complicated the situation of these tools because a GUI is composed by plenty of characters and graphics on the screen at particular positions, and therefore there is no purely textual representation of graphic content of the screen. Screen readers are therefore forced to use new low-level techniques to retrieve messages from the operating system and use them to build the off-screen model, a representation of the screen in the which stores the relevant text.

  Screen readers can also communicate information on menus, controls, and other visual constructs to permit blind users to interact with these interfaces. However, creating the off-screen model is a major technical challenge.

Operating Systems and applications designers have attempted to tackle these problems by providing access roads to these readers to access content without having to maintain any off-screen model. This involves the provision of alternative access by using Applications Programming Interfaces (API).

Thus, focusing on our scope and due to the software authoring tool <e-Adventure>, on which we are going to test our methodology is developed in the Java language we highlight the next ones among the existing API's.

2.3.1 Java Access Bridge

The use of Java for development of multiplatform applications is booming today, because it allows that applications with a minimum of care can be implemented in different environments, regardless of operating system that the host machine is running.

However, this multiplatform feature that Java provides is an accessibility problem for many users who use screen readers.

Java provides the multiplatform feature for their applications, because the execution of the application is made inside the Java Virtual Machine (Java Runtime) that interprets the Java application code.

The problem is that this virtual machine, for security reasons, is isolated from the host operating system, so that certain technical aids, such as screen readers, cannot access to the tree of objects in the Java application interface.

[18] http://www.freedomscientific.com/products/fs/jaws-product-page.asp

Therefore it is essential to install a bridge of accessibility, which allows screen readers to access certain properties of objects in Java.

The Java Access Bridge is compatible with the visual objects of the Java Foundation Classes, so that the installation of this layer of accessibility does not guarantee access to all Java applications. The use of visual components that do not belong to the class SWING increases, notably, the possibility of the application interface is not accessible to Java Access Bridge. In addition, use the properties of visual objects SWING relating to accessibility in order to ensure that the interface is accessible is advisable.

Accessibility on the Java platform under the Windows OS, is reached by using the Java Accessibility API (JAAPI) which is one of the core foundation services in the Java™ Foundation Classes[19].

The Java Accessibility Utilities help assistive technologies take advantage of applications developed using the Java Accessibility API. An assistive technology requires locating the components that implement the API, support for being loaded into the Java Virtual Machine, tracking events, and so on.

These utilities are not needed by application developers, they only need to be concerned with developing accessible applications using the Java Accessibility API. Instead, the Java Accessibility Utilities are meant to be used primarily by assistive technologies.

Summarizing, accessibility on the Java platform consists of four basic elements:

- JAAPI: Provides some kind of a contract between a Java application and the assistive technology (such as a screen reader or Braille display device).
- Java Accessibility Utilities: Provides the ability to get the information from the application and process it for further displaying with special devices.
- Java Access Bridge (JAB): This is the most important element in providing accessibility to the Java platform under the Windows operating system. It was introduced in J2SE 1.3.
- Java Foundation Classes (JFC): This is a library of GUI components, which fully implement the JAAPI.

Using the JAB together with a screen reader, like Jaws, allows us to have voiced feedback from a Java application.

2.3.2 Speech API

Speech Application Programming Interface or SAPI[20] is an API developed by Microsoft to allow the use of speech recognition and speech synthesis within Windows applications:

- DirectSpeechRecognition API: It is the low-level interface to incorporate speech recognition into Windows applications.

[19] http://java.sun.com/products/jfc/tsc/articles/accessibility/

[20] http://msdn.microsoft.com/en-us/library/ms720151%28v=VS.85%29.aspx

- Direct TextToSpeech API: It is the low-level interface of the SAPI to incorporate spoken text-Windows applications.

In general the Speech API is a freely-redistributable component which can be shipped with any Windows application that wishes to use speech technology. Applications that use SAPI include Microsoft Office, Microsoft Agent and Microsoft Speech Server.

SAPI also have high-level interface, it is called Control Object Model Interfaces (COM). COM Interface contains Voice Text API, Voice Command API, Voice Dictation API and Voice Telephony API. The next schema shows the structure:

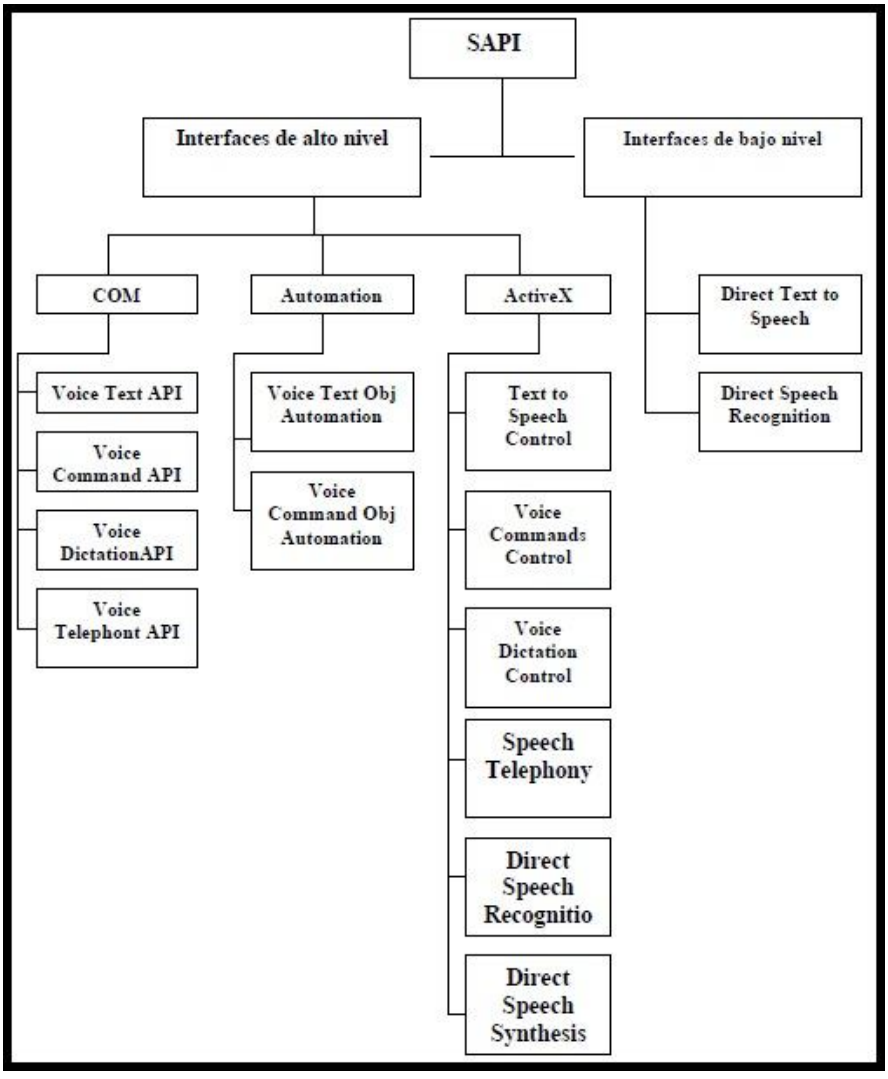


Figure 3. Schema interfaces SAPI by Long Bryan
in Speech Synthesis & Speech Recognition Using SAPI 5.1

The problem that appears is we have no way to link the Java project to the COM interface.

That is the reason why it is necessary to use the library com4j.jar[21]. This library aims to develop a Java library that allows Java applications to interoperate with Microsoft Component Object Model.

It is a Java tool that imports a COM type library and generates the Java definitions of that library. It is also possible to add a voice package called Espeak[22] which includes Spanish, French and so on voices.

2.3.3 Java Libraries

Another way to develop speech recognition and text-to-speech regardless of the platform, is based on including the Java libraries:

- FreTTS[23]: It is a speech synthesis system written entirely in the Java™ programming language. It is based upon Flite[24] which is a small run-time speech synthesis engine developed at Carnegie Mellon University. Flite is derived from the Festival Speech Synthesis System from the University of Edinburgh and the FestVox project from Carnegie Mellon University.

- Sphinx[25]: It is a state-of-the-art speech recognition system written entirely in the Java™ programming language. It was created via a joint collaboration between the Sphinx group at Carnegie Mellon University, Sun Microsystems Laboratories, Mitsubishi Electric Research Labs (MERL), and Hewlett Packard (HP), with contributions from the University of California at Santa Cruz (UCSC) and the Massachusetts Institute of Technology (MIT).

The drawbacks of using these libraries are a low voice quality and the voices being only in English, but by contrast, the use of them allows us use our Java application with speech recognition and speech synthesis on any platform, we are not limited to Windows.

Other advantage is that they are under the LGPL License (Lesser Gnu Public License) which is less restrictive than GPL. Using LGPL you can use parts of LGPL software libraries on proprietary software distributed under another license, which, in contrast, could not be done if the libraries were GPL.

2.4 Accessibility in Video Games

Having seen the potential audience for accessible games, the reasons for providing accessibility and the technologies developed now we must look at how this can best be accomplished.

---

[21] https://com4j.dev.java.net/

[22] http://espeak.sourceforge.net/

[23] http://freetts.sourceforge.net/docs/index.php#what_is_freetts

[24] http://www.speech.cs.cmu.edu/flite/

[25] http://cmusphinx.sourceforge.net/sphinx4/#what_is_sphinx4

Accessibility is probably one of the most important open issues on the video game industry agenda, since most video games are not accessible for people with disabilities. This issue may be related to a lack of awareness of game developers and the risky nature of the video game industry which may perceive that the return of the investment is not clear enough to support the inclusion of accessibility features in the games as it poses significant challenges and an extra development effort (Magerko, 2008).

Another reason is the lack of a widely accepted standard or specification to describe how to design interfaces or methodologies for accessible video game development (Grammenos, 2007) (Savidis, 2004).

Providing accessibility to video games or creating accessible games does not mean making them easier or even boring, simply need to provide alternative ways of transmitting information so that it may be received by all users, although this may lead to generate repetitive or too easy games that do not to capture user attention.

It is necessary to create accessible games keeping the intrinsic characteristics of a game, such as providing challenges, generating motivation and creativity, developing new capabilities in users and, in the case of educational video games, achieve learning by the student. Especially, it should be advisable to incorporate this adaptation as fine-grained as possible for each player, so that the game will be more engaging for each specific user(Houlette, 2004).

To realize these requirements and to go further in the study of new initiatives, there are some groups, such as The International Game Developers Association (IGDA) that propose, in their white paper that we have already included in section 2.2, various approaches which attempt to establish indications to develop accessible games or more especially to turn a non-accessible game into an accessible one, through the inclusion of subtitles and so on. But, in contrast, they not propose any methodology that covers all the possible requirements.

The development of many accessibility features require technical solutions that increase the total development cost (e.g. speech synthesis, voice recognition), especially if these features, are not considered from the beginning and it is necessary to make some transformations once the tool has been developed. This issue is even more relevant in the field of educational games, where budgets are more limited and the importance of accessibility is even greater. Therefore it is necessary to find new ways of reducing the cost of improving games accessibility.

One of the techniques adopted to incorporate accessibility in any software and also in videogames is to allow keyboard navigation of all controls with visual and spoken feedback, which impacts directly on the mobility, vision and auditory groups.

Nevertheless there are very different initiatives to improve the accessibility of video games (K. Bierre et al., 2005).

Focusing on hardware, one of the most common approaches is to widen the range of input devices the game supports, including compatibility with game pads adapted for people with different disabilities (Kearney, 2005), or new interaction devices such as systems that track the movement of the eyes or the head (Argue et al., 2004) and not only the standard mouse or joystick. There are also special keyboards that provide an alternative way to type the keyboard commands in.

Nonetheless, if none of these features are allowed, another idea, would be to enable an optional simplified interface mode with just the basic controls or the most used ones, hiding the less used ones until needed. This point could also be useful for cognitive problems, since it shows a simplified interface and therefore an easier scene of the game.

As example of game that addresses these features, we highlight Troopanum 2[26], an Action Game that focuses on blindness, low vision and mobility impairments, by allowing a variety of control devices, using audio cues for gamers with visual disabilities and representing the documentation in plain text suitable for a screen reader. Another example would be UA-Chess[27], a universally accessible multi-modal chess game, which can be played between two players, including people with disabilities (low-vision, blind and hand-motor impaired), either locally on the same computer, or remotely over the Internet.

In order to include also cognitive problems, which are the most difficult to handle, it is advisable to include in-game tutorials, which is not only helpful for this kind of disability but also for everyone who starts playing a game and ignore the mechanism to interact with it. This option is very useful when talking about attention disorders, since they may not be able to read a huge manual, but they can follow little hints when needed or just providing extra feedback which will help them to overcome the challenge in case of not finding the solution.

Another technique usually realized is to enable various degrees of difficulty, and also allow the user to modify certain features to make a further adaptation. Enabling several levels of difficulty is available in grand variety of games, where you have the choice at the start of the game, and where users will choose according to their skills and background.

However, in most cases, the adjustments are focused on the game-player interaction channels, i.e. the inputs and outputs of the game system. Since a game is primarily an interactive experience, these adaptations may pose a significant challenge from the technical point of view, since they can be composed by very different contents, requiring the use of assistive tools. These technologies already explained in previous sections, such as screen readers, screen magnifiers, speech synthesizers, voice recognizers or virtual keyboards, must be compatible with the games.

Another approach is to provide a multimodal design of the game interaction. This involves combining images with captions, subtitles, sounds and voices (Röber, 2005), allowing the interaction with the game using voice commands (Targett, 2003), etc. However, such approaches are often developed ad-hoc, providing particular solutions for each game, which makes difficult to reuse efforts. More general solutions, including frameworks and other development tools could help in this concern, especially for small projects.

Focusing on vision disabilities, it is advisable to incorporate the option to customize the fonts, in order to make them bigger in case of low vision, as well as the use of a standard text to achieve an easy compatibility with the OS in term and with the assistive tools. This feature will help not only low vision problems but also cognitive

[26] http://www.bscgames.com/bsc_troop2.asp

[27] http://www.ics.forth.gr/hci/ua-games/games.html

ones, since the users may need the aid of an assistive tool to help them in understanding problems.

Certain visual, hearing and motor problems can be dealt with by making minor adjustments in the game such as the time thresholds for replying the attack of an enemy, allowing the modification of degrees of difficulty to a further extent than usual, adding an optional simplified interface mode with just the basic controls, allowing an alternative to sound files by using bass vibration from the subwoofer or using better in-game tutorials as automatic help. etc. (K Bierre et al., 2004).

For instance, for blind gamers an option to turn off the 3D rendering in the start menu is important since they may not have good enough 3D graphic hardware to play your game, and they should not need it. Also, an option for the 3D engine to use no hardware acceleration is good, to avoid problems with erratic graphic drivers that cause crashes. This will enable blind gamers to not have to worry about updating graphic drivers.

These users also need obtain the majority of visual information by using another channel of transmision, to achieve visual disabled with a high percentage of blindness, could understand and handle the video game without feeling frustrated.

Some ways to achieve this challenge could be the use of sonar, which conveys to the users the distance to objects or enemies, or even, by pressing a key, it could convey useful information of what type of object they are facing in the scene.

Several examples of games that cover any of these initiatives are:

- *Terraformers*[28], 3D Adventure Game, which uses several methods to give feedback to users and to orient them, such as the sonar that indicates the position of the closest object in the direction the player is facing, voice feedback or a GPS system which provides a description of the area the player is in. Also users can enable a high contrast 3D mode.
- *Access Invaders*[29], which can be played in Windows and GNU/Linux. It has been done by Universally Accessible Games (UA–Games[30]) constitute a research activity of the Human–Computer Interaction Laboratory of ICS–FORTH[31]. In this context, we research, design and develop Universally Accessible Games and we create and test new related concepts, interaction techniques, methods and software tools. It can be concurrently played by individuals with different abilities and preferences, including people with disabilities, e.g., low-vision, blind and hand-motor impaired.
- *Honeycombs*[32], Puzzle game, focuses on color problems, especially in the choice of colors that are distinctive when using the grayscale.
- *Game Over!*[33], it is a game that can be played by no one. It has been developed by the research group UA-Games. The goal of Game Over!, is to be used as an educational tool for disseminating, understanding and consolidating game

[28] http://www.terraformers.nu/

[29] http://www.ics.forth.gr/hci/ua-games/access-invaders/

[30] http://www.ics.forth.gr/hci/ua-games/index.html

[31] http://www.ics.forth.gr/index.html

[32] http://uk.pc.ign.com/objects/546/546765.html

[33] http://www.ics.forth.gr/hci/ua-games/game-over/

accessibility guidelines. It has in each level, the guideline that it is violated[34] what make the developers check how easy it would be make it accessible since the beginning and how hard is play in these inaccessible conditions.

- *Terrestrial Invaders*[35], is a descendent of the Access Invaders game that also follows the principles of UA-Games. It is a "byproduct" of the development process of *Game Over!*, that through its various accessibility features, is able to address most of the accessibility guidelines that *Game Over!* violates.

Focusing now on auditory disabilities, they need the presence of subtitles that represent the dialogs and also the presence of bass vibration when a sound is happening to give them the needed feedback, either deafness or hard of hearing. As example in this section, we can present *Avandoria*[36], which is a Role-Playing game that faces both, deafness and hard of hearing problems by using subtitles.

As we have seen, the largest group is the mobility one. Thus, the features that may be enabled in this scope are the possibility of customizing the controls, which is useful for people with reduced mobility who cannot press at the same time for instance three keys.

An important aspect to consider when dealing with educational games that are accessible is the game genre, since not all genres are appropriate for learning. These games should be intuitive and easy to handle at once to be able to attract the attention of students and motivate them, even when accessibility features are enabled. An interesting trend to drive the student learning experience is the use of an attractive narrative story that involves the students in order to keep them motivated, focused, and concentrated on the tasks that are proposed. Therefore, the genre best suited to these requirements is the adventure game genre, also called point-and-click adventure games, because in these games, there is a story that challenges the user and this one must resolve it, such as the *Monkey Island* saga (Dickey, 2006). These features capture the players attention since this kind of games are based in fact reflect and think to solve the successive challenges, rather than the use of action.

2.5 Accessibility in Desktop Authoring Tools

Most desktop applications that include accessibility features allow the adaptation of the Graphical User Interface (GUI) by configuring some parameters, such as the size and font color, which can solve some of the most common problems. In this context, the IBM checklist initiative focuses on providing guidelines for developing software accessible tools. Also, in the ATAG guidelines, the accessibility of the authoring tool is also considered as a strong point as we have studied in-depth in section 2.2.

Another common strategy is to seek compatibility with external assistive tools like screen readers such as Jaws (Hitchcock & Stahl, 2003). This is the approach that has

[34] http://www.ics.forth.gr/hci/ua-games/game-over/game_levels.html

[35] http://www.ics.forth.gr/hci/ua-games/ti/

[36] http://www.topmudsites.com/forums/mudinfo-avandori.html

been applied, for example, to increase the accessibility of well-known tools like Microsoft Word™ (Hetzroni & Shrieber, 2004) or Microsoft Excel™ (Doush et al., 2009).

However, this approach is insufficient to make accessible other complex tools such as Paint™, Adobe PhotoShop™ or Adobe Flash™. First, these tools are usually focused on the creation of highly visual content (e.g. images, photographs or maps).

Second, the amount of editable information that the tools need to provide to the user is very high, which makes navigation through the elements more difficult for anyone, but especially for visual disabled people.

Third, these tools usually implement interaction modes that are not accessible because they require high eye-hand coordination, such as drag and drop (Wang et al., 2009). In this type of tools, the development of accessibility to target visual disabilities can become an arduous and expensive task. In this sense the tools that have achieved greater success in accessibility terms are the ones which combine support from external assistive technologies with the introduction of facilities to simplify the edition of content. For example, the templates that Microsoft Power Point™ includes facilitate the creation of presentations by visually disabled authors as they do not have to deal with the spatial organization of the contents of each slide, which is specified by the template.

The accessibility of desktop authoring tools is a complex issue that has not received enough attention yet. Several examples of this lack of accessibility inclusion are game development environments which allow people with no technical background develop their own games, such as:

- Game Maker[37]: Game Maker authoring tool allows to make exciting computer games, without the need to write a single line of code.
- Unity3D[38]: Game authoring tool with one of the largest user communities.
- Adventure Maker[39]: Graphical adventure authoring tool.
- Alice[40]: Interactive content for education authoring tool.
- The FPS Creator[41]: Tool that allows to create an own person shooter game.
- Adventure Game Studio[42]: Tool that allows to create an own point-and-click adventure games, similar to the early 90's “Sierra” and “Lucasarts Adventures”. It consists of an easy-to-use editor to create your games, and a run-time engine to play them.

In these cases, it is necessary to include accessibility for every individual game and not doing it once for every game through the editor.

Game authoring tools are a good example of these highly visual applications. Games are a rich and complex medium, and therefore game authoring tools need to present a

[37] http://www.yoyogames.com/gamemaker

[38] http://unity3d.com/unity/features/

[39] http://www.adventuremaker.com/

[40] http://www.alice.org/

[41] http://www.thegamecreators.com/?m=view_product&id=2001

[42] http://www.adventuregamestudio.co.uk/

lot of information and features that the user has to configure to compose the final game, such as characters, objects, conversations, cut-scenes or scenes. The simplification of the tools, removing or hiding configuration options that are not indispensable could address this issue, at least partially. Nevertheless, this approach is not always feasible as an oversimplification of the tool could limit the expressive resources available to create the video games, leading to the production of poor video games in terms of engagement and educational value (Foss & Eikaas, 2006; Squire, 2003). Therefore one of the main challenges is how to structure the information that the tool presents so all users can navigate through the elements of the GUI using different input devices (e.g. mouse, keyboard or a microphone).

2.6 Conclusions

The need to adapt contents and software as much as possible to each user is an open research field that it has been addressed since the AHS (Adaptive Hypermedia Systems) were developed and introduced in the Artificial Intelligence.

As we have seen, the target public to which accessibility features (as a special subset of adaptability) is quite various, therefore the adaptive experience is more valuable if it is adapted to each user or group of users.

In this regard, we have seen how this issue has been addressed in computers in general, and in the Web in particular. The multiple approaches presented, have been motivated by the need of making people with special needs part of the Information Society.

As a consequence, the common pattern within the initiatives that promote accessibility in the Web is their aim to guarantee that the user can get to the information. After that is done, it does not really matter what happens with the content that the user got.

Following this emerging trend, if video games start being used as an immersive learning experience, they must be adaptable too.

But, although the approach may be valid for the Web, just making games “accessible” for users with special needs, this may not be enough. The reason is that the motivation that users have to utilize the Web and video games are different.

While users mainly tend to use the Web for gathering information, they use video games for getting outstanding engaging, immersive and finally, satisfactory experiences.

Not only video games are considered a learning experience, but also authoring tools might be used as an engaging experience to develop any kind of software or content that it could be used as a learning unit.

At this point, it is fairly hard the incorporation of specific accessibility features in the authoring tools, since it is not enough guarantee the access to the information.

This kind of tools present a challenge when incorporating more specific techniques to provide access for everyone since they are highly visual, what demand high interaction

visual-hand, and they present a lot of settings to configure which means that they have plenty of information to present to the user who must be able to receive it.

Thus we advocate for a more specific perspective to cater for people with special needs reach to manage themselves as a non-disabled person when talking about the creation of high visual contents.

Therefore, next chapter raises the requirements and guidelines to meet by developers who attempt to carry out accessible authoring tools.

3. GENERAL MODEL PROPOSED

As it has been discussed so far, the main goal of this work is to specify which accessibility features are necessary to take into account, to include and to develop in an authoring tool in order to allow that everyone can interact with the software, even people who have some kind of disability.

Our aim it is to develop a reliable general evaluation model which can be applied to any software tool and also facilitate the understanding of the application in question, including also for non-disabled people.

Several requirements have been identified to provide a software tool with accessibility following the ideas of the various guidelines introduced above. The question is how we can provide accessibility in game authoring tools and less specific, in highly visual desktop software tools.

In this way, we have focused especially in visual disabilities since these kinds of tools mainly report visual feedback and they require high interaction. Nevertheless, we have also included auditory and motor disabilities.

We have developed a methodology for the evaluation of accessibility, based on certain points that are necessary to enable accessibility. We have studied the possible needs according to disability which possess (Jiménez Peñuela, 2005; Valle Pérez), we have taken the guidelines explained by IBM and WAI-ATAG and we have adapted them to fulfill our requirements more specific.

However, an application developer does not have to provide all of the solutions. You need to enable your application so that programmers writing and creating Assistive Technologies (AT) have the ability to obtain from the application program what it needs to present to its users.

This chapter includes three sections which discuss first of all the requirements, then the methodology with each of the identified guidelines, why it is necessary to develop it and what kind of disability is mainly addressed for, and finally a section called Preliminary Evaluation where we show the assessment of different authoring tools and a section called Conclusion to summarize.

3.1 Requirements

In this section we want to clarify which features are necessary to tackle in terms of authoring tools.

Thus, focusing in our potential scope, we address, according to the various groups of disabilities depicted in section 2.1, which feature is addressed for each type of disability in high visual tools and within this group into authoring tools.

In section 3.2, it has been addressed one methodology studying the major requirements to turn a software tool into another one which is accessible or just the guidelines to making it, since the beginning, in an accessible way. So, taking into account these guidelines we have made a table in which we emphasize in what type of disability is addressed for each of them.

Therefore, we present a table, where we highlight as visual disabilities blindness, low vision and color blindness, as auditory disabilities deafness and hard of hearing and the

last one, motor disabilities. And on the other side, we introduce the categories where we place the guidelines.

The Table 3 we obtain is the next one:

Category	Visual Disabilities			Auditory Disabilities		Motor Disabilities
	Blindness	Low Vision	Color Blindness	Deafness	Hard of Hearing	
Object information	**x**	**x**				**x**
Provide a visual focus indicator that moves among interactive objects as the input focus changes.	x	x				x
Provide semantic information about user interface objects.	x	x				
Associate labels with controls, objects, icons and images.	x	x				x
When we use electronic forms, the form should allow people using assistive technology to access the information.	x					x
Place elements in a logical order so that the assistive technologies could interact with them in a logical way.	x	x				x
Images and Color	**x**	**x**	**x**			
Provide an alternative text description for images.	x	x				
Use color as an enhancement.	x	x	x			

Category	Visual Disabilities			Auditory Disabilities		Motor Disabilities
	Blindness	Low Vision	Color Blindness	Deafness	Hard of Hearing	
When color customization is supported, provide a variety of color selections capable of producing a range of contrast levels.	x	x	x			
Support system settings for high contrast for all user interface controls and client area content.	x	x	x			
Multimedia Content Accessible	**x**	**x**		**x**	**x**	
Provide an option to display a visual cue for all audio alerts.				x	x	
Provide accessible alternatives to significant audio and video.	x			x	x	
Provide a textual summary which explain what the multimedia content is conveying.	x			x	x	
Provide an option to adjust the volume.	x				x	
Provide an option to display animation in a non-animated presentation mode.	x	x		x	x	
Configurable Fonts	**x**					
Provide text through standard system function calls or through an API (application programming interface) which supports interaction with assistive technologies.	x					

Category	Visual Disabilities			Auditory Disabilities		Motor Disabilities
	Blindness	Low Vision	Color Blindness	Deafness	Hard of Hearing	
Inherit system settings for font, size, and font color for all user interface controls.	x					
Keyboard Access	**x**	**x**				**x**
Provide keyboard equivalents and shortcuts for all actions	x	x				x
Do not interfere with keyboard accessibility features built into the operating system.	x	x				x
Simple and Concise Language				**x**		
Presence of icons to support the words in the menus.				x		
Use a direct language, without excess words.				x		
Use a familiar vocabulary.				x		
Highlight the keywords.				x		
Setting the Scenes in Alternative Way	**x**	**x**				**x**
Text Speech when composing scenes	x	x				x
Voice Recognition when composing scenes	x					x

Category	Visual Disabilities			Auditory Disabilities		Motor Disabilities
	Blindness	Low Vision	Color Blindness	Deafness	Hard of Hearing	
Structuring the Information	**x**	**x**	**x**	**x**	**x**	**x**
Documentation	**x**	**x**	**x**	**x**	**x**	**x**
Documentation, online information and help	x	x	x	x	x	x
Documentation in an accessible format	x	x	x	x	x	x

Table 3. Table where it is shown which disability affects each guideline of the methodology

Thus, in the next sections we tackle with each guideline studying its highlighted points, its target group and its description.

3.2 Evaluation Model

In the previous section we have depicted each guideline, explaining why we have included it, dividing it in several highlighted points that are necessary to check them over the tool in question, including also the other guidelines that we have started from.

The methodology that we propose is especially addressed to develop video game authoring tools that incorporate adequate alternative access mechanisms that are accessible for users with motor disabilities, hearing and partially sighted. For users with blindness would be more complex and it would be necessary substantial improvements to support alternative interfaces.

Within this methodology, we have followed the aim of including these approaches:

- **Support for speech synthesis**
  People who are not able to perceive the visual interface of an application, need an equivalent alternative to provide them with this information through channels they can use. The Text To Speech (TTS or text to speech) is a system whereby a program is able to verbalize, through a synthetic voice (computer generated), all the information we need to convey. Thus, implementing a layer of accessibility

based on TTS where it is provided spoken texts equivalent to visual scenes that appear in a program would allow the software's feedback was completely accessible to a visually impaired person.

- **Support for voice recognizing**

  To make the application accessible to people with a physical disability that prevents them from using pointers and systems standards such as keyboard input systems, would be valuable to implement a voice recognition layer that would allow these people to interact with the application.

  When designing the layer of grammar for speech recognition would be necessary to take into account the following aspects:

  - There should be common words in any game to get a list of words to use for that particular game. The words could be: "help", "What I can say?", etc.
  - It is necessary to provide the layer a grammatical system flexible enough to permit using different tenses and removal of articles or prepositions.

- **Navigation through keyboard.**

  In case of visually impaired users, who cannot use a pointer to move around the screen or click on the objects that interest them, it is advisable implement another layer of accessibility, in order to give an input method in an accessible way.

  For a game creation based on interactive virtual scenes, with characters, objects and movement through different scenarios, there are two possible approaches, first is the creation of nested menus and second, is using a grammar to give textual commands to interact. The grammar may be the same we have defined for the speaking recognition layer.

  Between these two approaches, we explain why is easier the use of nested menus:

  - If we use a grammar to give textual commands, the disabled user will have to learn it while non-disabled users do not.

    It should also take into account the possible use of the platform by a person with cognitive disabilities. These people may have problems to memorize and use these commands, so they need to use the keyboard.
  - Quick access to information: With textual commands, the user should write a few words to access each active zone in the game, or to perform an action.

    But this action may delay a user who does not have a good write speed.

    However, with menus the user must push enough buttons repeatedly to access the information more conveniently and easily.
  - The use of textual commands can adversely affect the actions that the user can perform with the objects around him.

    A game may have certain options that can be easily guessed but others ones there are not so easy to guess them.

    Thus, the speech synthesizer should verbalize every action and the user should select the desired option once heard all of them, leading to a more slowly system when interacting with objects in the environment.

Moreover, in most cases, the use of textual commands, unknown to the user, may require to implement a semantic distance measurement to detect words or phrases in relation of synonymy.

We highlight that no tool should be fully accessible to all profiles of disability. Therefore, once we have presented the possible layer we should incorporate, next we present the categories we have addressed to develop a good code and a good design.

Each category has been already presented in previous section, where we have clarify what disability group is addressed for.

Thus, in the next section, we follow the same order of categories.

3.2.1 Object information

This section is gathered from the checklist provided by IBM. It is focused on providing the needed information about the elements which compose the total interface, like the focus or semantic meaning, to communicate with assistive tools. It is important in a high visual tool, the way in which it conveys the information, so that an assistive tool is able to process it.

This section addresses the problem of how to pass on the needed information in order to achieve people with visual disabilities especially, can perceive the context of the application by using screen readers and so on.

Provide a visual focus indicator that moves among interactive objects as the input focus changes.

We need that a visually disabled person knows where she is any time as well as a non-disabled person knows it, by seeing the visual focus. This focus indicator must be programmatically exposed to assistive technology in order to transmit the speech of where we are.

Provide semantic information about user interface objects.

When an image represents a program element, the information conveyed by the image must also be available in text. This guideline focuses on visual disabilities, since it is necessary to convey a description of these elements in order to transmit it to a screen reader.

Associate labels with controls, objects, icons and images.

If an image is used to identify programmatic elements, the meaning of the image must be consistent throughout the application. But not only the images, any object that we use in the GUI should include a label that depicts in a couple of words for what it is going to be used. This subsection focuses also in visual disabilities and in motor disabilities as well, since it is easier to wander the objects by typing than by using the mouse motion.

When we use electronic forms, the form should allow people using assistive technology to access the information, field elements and functionality required for completion and submission of the form, including all directions and cues.

This section focus in good programming techniques in order to make easier the extraction of information in forms, where there is an interactive interaction with users. This process is quite important since it provides the collection of information by the assistive tool.

Place elements in a logical order so that the assistive technologies could interact with them in a logical way.

You need to add components to your application in a logical order so that the user who is navigating by the application through keyboard or through speech, is getting the information in the order the developer added it.

This logical layout affects the keyboard tabbing sequence and the order an assistive technology uses to read your application. Assistive technologies, today, use the order with which components are added to determine the logical order to use to speak application components.

3.2.2 Images and Color

We have included this section because it is an important part of the accessibility issues in which colors are an essential part to build a GUI, but they are not so important if we are facing with visual disabilities. Thus, it is advisable to make a good use of colors and especially try to build a GUI useful for non-disabled as well as for visual disabled.

For this purpose, we have studied the Guideline 1.4 of the WCAG 2.0: "*Distinguishable: Make it easier for users to see and hear content including separating foreground from background*", focusing especially in "*1.4.1 Use of Color", "1.4.5 Images of Text"* and **"***1.4.8 Visual Presentation*".

This section addresses visual techniques to make good programming in visual contents issues.

Provide an alternative text description for images.

It is the first requirement that is addressed when making accessible either a Web site or a software tool. This refers to visuals disabilities, so that the screen reader can describe what the image is showing.

Use color as an enhancement, not as the only way to convey information or indicate an action.

It is not advisable the use of colors to convey information, such as the use of green when something has been done properly and the red in the opposite case. This use of color must always be enclosed by text. In this way, screen readers will be able to transmit the information and color blind users could access the information the color is conveying.

When color customization is supported, provide a variety of color selections capable of producing a range of contrast levels.

Here, we highlight the need of providing the user a wide spectrum of colors. Thus, it will be easier for the users make a customizable selection according to their needs, taste and so on.

Support system settings for high contrast for all user interface controls and client area content.

This technique is rather used in Web sites, where this requirement is almost indispensable.

In our scope, it would be advisable that the tool in question, could be able to extract the OS settings related to colors and contrasts.

3.2.3 Accessible Multimedia Content

This section tackles the issue of making accessible any kind of multimedia content. This challenges developers because it is necessary to include more resources to turn it into an accessible content, and this, usually raises the costs of developing.

In our scope, we do not deal with multimedia content, because we are facing with adding accessibility to a software tool and not in video games. In spite of this issue, we include this section since we deal with images, videos, conversations and so on, to compose a game scene, so that we consider that it is important specify how this resources must be done to the developer can extract the required information from them.

We have inspired ourselves in the IBM checklist to include this section which focuses especially in visual and auditory disabilities.

Provide an option to display a visual cue for all audio alerts.

This technique allows the user to choose visual cues for audio alerts. The application can provide visual cues for alerts by displaying a dialog window or providing a status indicator on the taskbar that flashes to attract the user's attention. It is focused on auditory disabilities.

Provide accessible alternatives to significant audio and video.

Here we focus in the transmission of the information in a multimodal way, for instance, text captioning for audio by using subtitles will help to auditory disabled persons or auditory description for video that will help to visual disabled persons.

Provide a textual summary which explain what the multimedia content is conveying.

Here, we focus on the need to show in a visual way the summary of the audio contents and videos. Both of them are necessary to provide visual and auditory alternatives.

Provide an option to adjust the volume.

Users may not be able to hear or distinguish sounds at certain volumes, so they need the ability to adjust the volume. A user that is hard of hearing can have difficulty hearing auditory output and needs to adjust the volume so they can hear the warnings and messages. A user with a vision disability may depend on speech output and will need to adjust the volume to accommodate different environmental conditions.

Provide an option to display animation in a non-animated presentation mode.

This section focuses on the transmision of information in a multimodal way. For instance, if we have animations in applications, it would be advisable to provide a checkbox to enable or disable the animation.

Or provide an alternative or redundant way of presenting the information in a non-animated mode: If there is an animated icon, in addition to an animated icon provide an equivalent text message.

For example, if the loading of an application is represented with an animated icon then, in addition to the animated icon, provide the user the option of reading a text equivalent message, like “loading" on the status line. Or, as in the dialog box below, provide the download percentage in text along with the animated status bar.

3.2.4 Configurable Fonts

Fonts are one of the main points that are arisen when adding accessibility. Developers should make available the fonts settings in order to the user can customize them. Because of some users may need larger text and others may not, it is necessary at least incorporate to levels of letters. This feature is especially addressed for visual disabilities, and based on a combination of the WCAG and the IBM checklist.

Provide text through standard system function calls or through an API (application programming interface) which supports interaction with assistive technologies.

Screen readers determine what software is doing by watching calls to drawing functions and remembering what text and graphics have been drawn and where. Screen readers also save attributes of the text such as font, size, color, and style, to be reported to the user. They also watch information being copied from one location to another and being erased or overwritten by other text or graphics.

The component in a screen reader that does all this work is called an off-screen model (OSM). OSM's rely on being able to monitor normal system drawing operations. Screen readers use the OSM to get information such as text content, text input caret location, and text attributes. If text is displayed in a non-standard way, the screen reader may not be able to detect it and read it to a blind user.

Inherit system settings for font, size, and font color for all user interface controls.

Someone with a vision disability who has difficulty reading small text or text that does not have sufficient background contrast can use display settings available through the operating system to make information on the screen more accessible. These settings allow users to modify display settings to make software accessible without purchasing additional hardware and software.

3.2.5 Keyboard Access

This is one of the features mainly addressed regarding accessibility issues. It is quite important to enable the accessibility via keyboard by using the tab key, arrows keys and enter, to allow visual disabled to interact with the interface since they cannot do it via mouse motion.

This guideline is indexed in every checklist. Thus, in the IBM checklist it is referenced as "*2.1 Make your application keyboard accessible*", as well in the WCAG 2.0 as "Guideline *2.1 Keyboard Accessible: Make all functionality available from a keyboard*", and in WCAG 1.0 as "*Guideline 9. Design for device-independence*".

This subsection addresses visual impairments issues and also motors ones, since it is necessary to enable shortcuts which allow an easier manipulation by persons who have motor disabilities.

Provide keyboard equivalents and shortcuts for all actions.

It is advisable to develop these features, first referencing the possible sections your program may have in the menu bar and second, setting up shortcuts for the most used actions. In this manner, we will get to the required action in one step and without looking for it.

Usually, it is set to use ALT+letter to provide keyboard equivalents to access to sections in the menu bar, and CONTROL+letter to set up a shortcut for any action that is going to be used often.

Do not interfere with keyboard accessibility features built into the operating system.

The operating systems have a set of accessibility options which enable users with disabilities to customize system wide settings to improve accessibility. For example, a Windows user with a physical disability may not be able to press multiple key stroke sequences, such as Ctrl+Alt+Delete, simultaneously.

Keyboard accessibility options make it possible for people with a variety of disabilities to use their computer. If the application software interferes with these options, some users may find their system unusable.

3.2.6 Use Simple and Concise Language

This subsection is especially addressed to provide facilities to persons who have understanding problems, such us cognitive disorders or auditory disabilities.

Also we take into account the fact that the natural language may not be the mother tongue of people with auditory disabilities since they may use the sign language. That is why it could be difficult for them the understanding of long sentences.

This point is indexed in WCAG 1.0 in "*2.2 Making Content Understandable and Navigable*".

Presence of icons to support the words in the menus.

As well as we have said that icons must be go with a brief text or a label which describe them, it is the same in the opposite way, it is advisable the presence of icons going with texts.

Using a direct language, without excess words.

This feature is easy to satisfy since we are dealing with software tool which almost always work with short sentences next to the elements to depict them. It is rarely used the sentence subordination which it is advisable to avoid it.

Using a familiar vocabulary.

The use of commonly used words will simplify the understanding process for everyone, not only for people who presents any kind of disability.

Highlight the keywords.

Set various priority levels to highlight the most important sections, with a larger size or bold type, the subsections and the sections deeper in the tree of priority.

3.2.7 Setting the Scenes in Alternative Way

This point is especially addressed focusing on dealing with images, so this point it is the most restrictive of this methodology, since it focus on how to place other images, such as a characters or objects, or how to define interactive areas over a background image.

Studying the way where the scenes are composed, usually they are made by using panels drag and drop, by dragging the component into the exact position where we want to place it, and dropping it. This process is highly visual and to incorporate accessibility are necessary techniques to recognize where the objects are placed in the background image, description of items and so on.

Also this point, as we have said in the motivations, is also especially focused in visual disabilities, since the depicted process needs a high interaction eyesight-mouse.

Text Speech when composing scenes

Since these tools are highly visual, it is also necessary to check if it provides sound feedback to the user when creating a scene which implies placing objects, characters and so on.

In case of a game scene with a virtual scene, where there are objects, characters, exits and so on, it is important to provide to visually impaired users as much information about the environment in which they are interacting. But this need does not mean an additional benefit giving them more information than usual.

First of all, they need a general description of the image scene, and secondly they need more detailed descriptions according to their choices.

This information can be given through the presence of tags or dividing the scene in several smaller parts to localize easier the selected area where the game creator wants to place the object.

For instance, focusing on the active areas of a scenario for a non-disabled user, they could see them passing over the screen and watching the cursor change, but on the contrary, for the visually disabled user, it is needed the presentation of a spoken list of active objects, so that both profiles have the same information either by text to speech or through images highlighted.

Voice Recognition when composing scenes

Using the voice recognition allows the interaction with motor disabled people. As we have seen, it should be advisable the definition of a flexible grammar where we contemplate the synonyms and the absence or not the articles and prepositions.

For instance, by saying “Place <object1> on/in/over/under <object2>”

3.2.8 Structuring the Information

This point is mainly addressed to the design stage, when developers must think about how they place the components of the GUI.

As any other stage before starting to develop code, it is important to devote a period of time to make a good design of the tool, to study and identify the requirements, to establish use cases and so on.

Thus, this stage of design the structure is quite important in terms of accessibility, due to the need of enable the possibility to access everywhere in the tool. This means, that it is advisable to think about a good distribution of the components, especially when there is a lot of information.

When there is too much information to distribute, usually, it is normal the structuring of the information in nested menus that are hard to access if you are not using the usual input device, the mouse.

Whatever the device used to access, it is absolutely necessary the access to every component, so in case of not using the mouse, it is necessary control the focus and handle it in a good way in order to reach all the components by using the keyboard.

In addition, to reach with the focus all the elements by using keyboard means that we have done a good design and a good distribution of the components so they can be easily accessed by an assistive tool or by any other device.

This point is especially addressed to provide facilities to everyone, since developing a good programming facilitates the use of everyone by making the software clearer and easier to use it, but it is especially focused on the use of assistive tools which must access to the software in question. If the code is not developed in a good way, they will not be able to extract the needed information and the same if they cannot reach every component, so this point is valuable for visual disabilities when using a screen reader, screen magnifier, Braille devices and so on.

3.2.9 Documentation

This feature cannot be forgotten because it is valuable for everyone who interacts with some new software. It is also depicted in the IBM checklist.

Documentation, online information and help

A serious problem with many software applications is that documentation for keyboard access features is non-existent. Documenting the features in print format does not help blind users. To correct this problem, documentation about accessibility features should be available electronically in the application's accessible help facility, and a user should be able to activate the help using the keyboard, typically the F1 key. In the case of blind users, they can use the same screen reader program to access the help facility that they use to access the application. It also improves usability to define a "Keys Help" section in your help facility.

Documentation in an Accessible Format

Some users may not be able to access documentation if it is not in an accessible format.

Thus, to make the documentation accessible we must be aware of providing the next formats, ASCII text, Javadoc in JDK 1.4 is accessible, Accessible HTML or Accessible PDFs.

Once, we have depicted the methodology, in the next section we introduce how we test it and over what tools, in this manner we enable the set of tools for what this methodology is addressed for.

3.3 Preliminary Evaluation

In order to establish if the guidelines selected are reliable enough, we establish a set of tools in which it would be advisable check the guidelines proposed.

Usually, assistive technology (AT) facilitates the use to major authoring tools for educational content such as Microsoft Word™ or PowerPoint™, etc., that are not so visual as the ones which address the creation and edition of video games or even design and photo editing tools such as Adobe Photoshop™, Paint™, Gimp™ and so on.

Although the PowerPoint tool is also quite visual, it is easier to communicate with an assistive tool, since in this tool the contents are fairly structured through templates which place the text in prefixed areas. In this way, it is easier for assistive tools access to contents that are placed always under various templates.

Focusing on more visual tools, we highlight tools that present an image in order to touch up various features or to design a new image with all the features that may be necessary to make a professional design.

Furthermore, we go a step over, and we study tools that have a high visual interaction and we focus especially on video games authoring tools, since this is a field less addressed and they need special features, more than common already developed.

Some of these features are the incorporation of accessibility when the tool presents panels drag and drop that are totally inaccessible for a disabled person.

A drag and drop panel, allows the placement of objects by dragging the cursor until the accurate point where we want to place the object. This task requires a huge visual interaction and that is the reason why it is one of the challenges to make it, in an accessible way.

Another feature to take into account, is how to place all the needed configuration settings in an accessible format, so that they could be reached by using the keyboard, this means that assistive tools also will reach them.

For all these reasons, and once, we have argued our motivations and the selected guidelines, it is advisable to check them in some tool, and to propose solutions.

To realize these proceedings we take into account accessibility criteria focused on access barriers experienced by people with visual, motor and auditory impairments.

In each section, we show the problems encountered in the user interface development tool and we focus on the presence of mechanisms or elements that address accessibility. It has also added a paragraph indicating the possibility of using certain technologies or features that the application uses to create games that allow the design of accessible interfaces for some specific disability profiles.

3.3.1 Applicability over the Unity 3D tool

This platform of development of electronic games allows the development of multimedia application on multiple platforms, creating 3D virtual environments.

Regarding the environment, it presents a GUI fairly complicated with a lot of settings to configure, plenty of elements to include and various panels with subpanels as we can see in Figure 4.

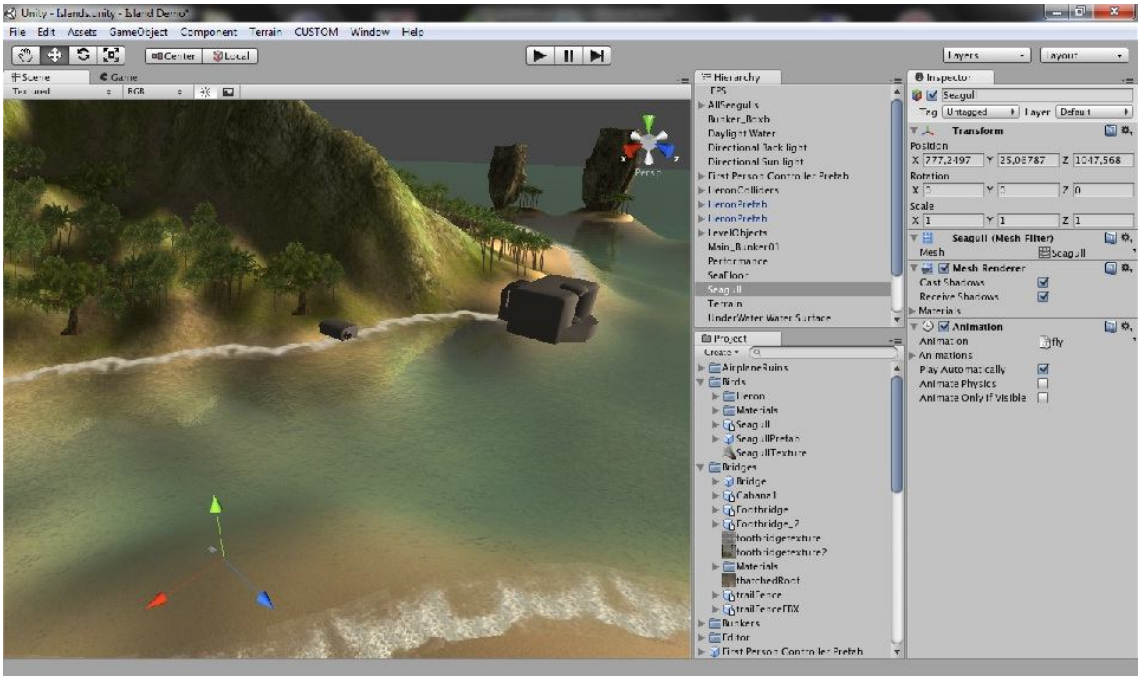


Figure 4. Screenshot of the Unity 3D authoring tool

Thus, focusing on the features that present the development tool under the Windows OS, we highlight the next problems:

- The division into sections of the interface can be confusing for new users, although the documentation is a tutorial for beginners.
- The GUI is completely inaccessible due the big quantity of settings to configure.
- All the contents lack of alternatives for product support.
- Also in the contents, it is not respected customization contrasting color, font sizes and other customizations allowed by the operating system to increase the level of accessibility.
- Documentation that accompanies the development tool is in HTML and PDF format. Although some errors were found accessibility, it is not entirely inaccessible, so a user can browse support product and view the various sections of the documentation.
- Regarding the accessibility support, the product does not contain the use of software features for creating entertainment content accessible to a specific disability profile.

There are some possible ways to create software products accessible to some disabled profiles by incorporating several features to this development tool:

- This tool allows the management of key events, joystick and mouse, so it might be possible to create a customizable interface with all the events activated, for users who can use only one of these input methods.
- It is possible to create sound events guide for users with visual disabilities.
- It is possible to create on-screen controls and visual elements belonging to the game interface and displaying visual warnings for users with hearing disabilities.

3.3.2 Applicability over the AdventureMaker tool.

This program allows the creation of interactive games such as "point & click" of flat and three-dimensional interfaces, based on the management of visual and interactive surfaces.

We can see the GUI of this tool in Figure 5:

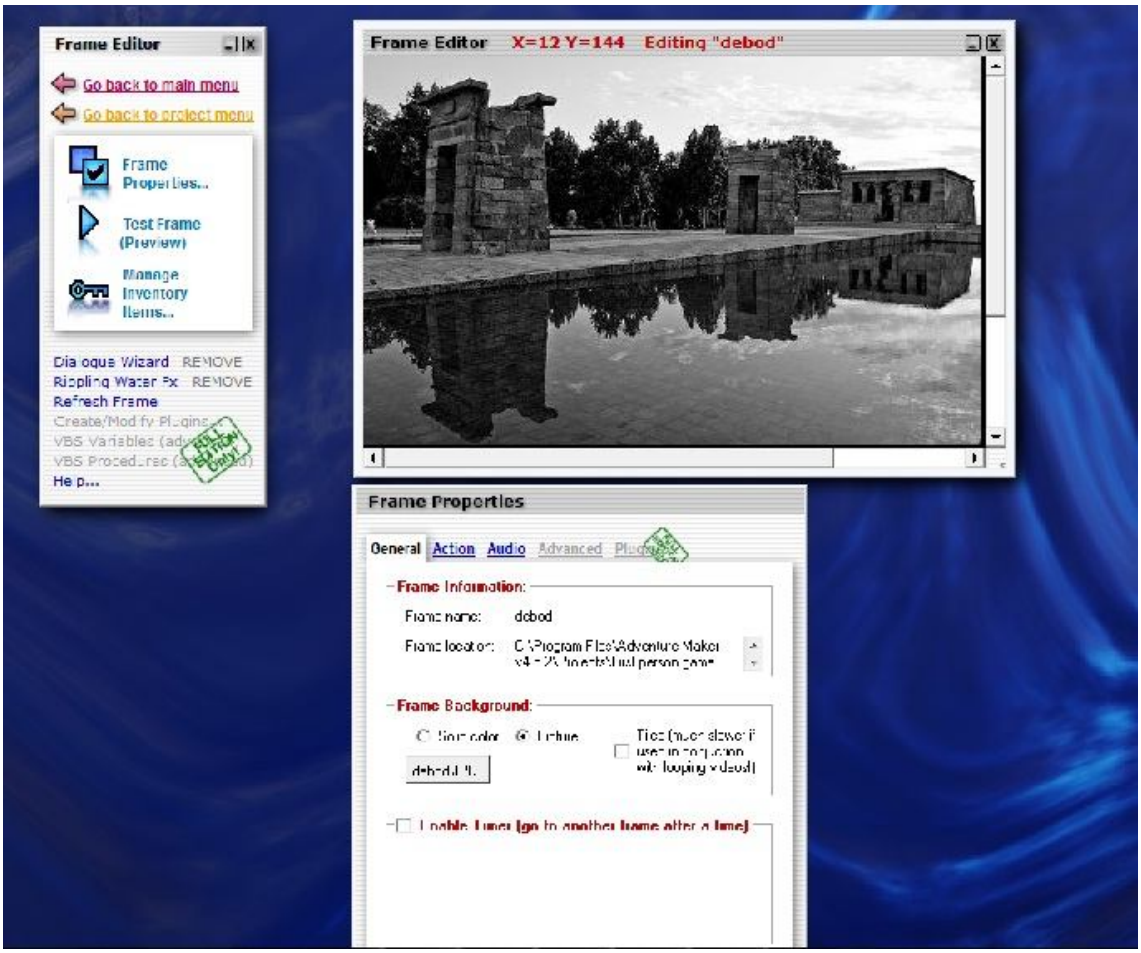


Figure 5. Screenshot of the AdventureMaker authoring tool

After evaluating the product's interface for Windows gives the following results:

- The interface has been designed with controls that allow you to "point & click" but they are non-standard Windows libraries, so that errors were found when using the screen reader to interpret the type of control and its status.
- Control of the application using the keyboard is partial, because there is not the presence of a coherent and complete navigation through the tab key.
- Screen magnifiers products may encounter problems when magnifying the playing area since it is not used the technologies of multimedia resources management for the specific OS.
- Also it is not respected the customization of color, font, sizes and other customizations allowed by the operating system to increase the level of accessibility.

- The documentation that accompanies the development tool can be viewed from both, the interface development tool and web browser.

We present possible ways to introduce accessible features to some disabled profiles support on this development tool:

- The platform does not allow proper management of keyboard or joystick events, so the only well-managed input method is the mouse. This severely limits the possible ways to create accessible interfaces for different profiles of disability.
- The platform allows a partial control system sound events but the problem is that it is not allowed precise the volume, frequency and balance to create a personalized sound environment to be helpful for people with visual disabilities.

3.3.3 Applicability over the Alice tool.

Alice is a 3D programming environment that makes it easy to create an animation for telling a story, playing an interactive game, or a video to share on the web. Alice is a freely available teaching tool designed to be a student's first exposure to object-oriented programming.

In the next Figure (6), it is shown the GUI of the authoring tool Alice

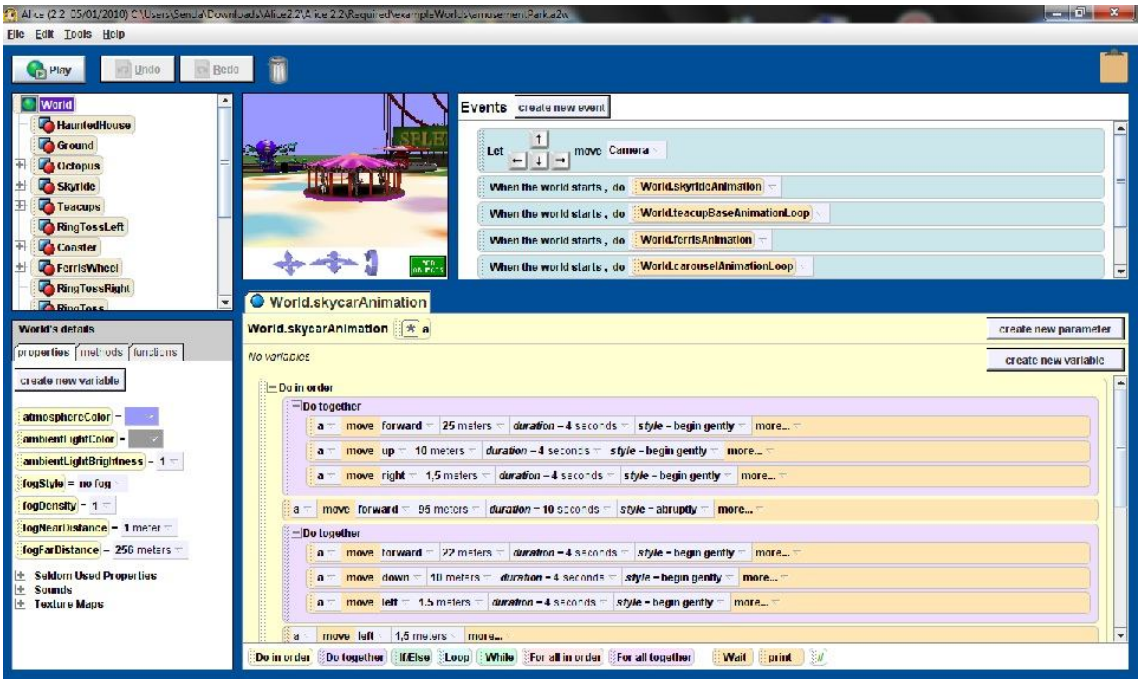


Figure 6. Screenshot of the Alice authoring tool

After evaluating the product's interface for Windows gives the following results:

- The interface has been made mainly with Java technology without any support AWT or Swing libraries, so the interface is completely inaccessible to users who use assistive technologies.

- The interface has too many elements into a single view, and while four control separations are defined, this results confusing and too complex for a novice user.
- Management of input events is varied, allowing you to create very flexible game interfaces, so there are opportunities to incorporate solutions to the problems of accessibility.
- Sound management allows the repetition and volume control, but does not allow frequency management and balance.
- The documentation is poor, being better see the one existing in the web portal.

Hence, we suggest some possible ways to create software products accessible to some disabled profiles that support this development tool.

The tool allows many features, so the flexibility to create interfaces is very good. You can use this flexibility to create accessible multimodal channels to convey the information between the user and program.

3.4 Conclusions

In this section we emphasize the conclusions we have reached by making this methodology. We highlight that there are plenty of guidelines that could be included in this methodology, but we have especially focused on the guidelines we think are especially valuable for our assessment. This methodology is not closed, and as soon as it is required the introduction of new ones, due to the fast advance of technologies, the can be added.

Thus, we have especially addressed guidelines that solve visual problems, since this methodology is applied over high visual tools, which means that is on this point where we find the most challenging feature to tackle.

Therefore, as we have said in the previous section, there are different tools on which we could apply these guidelines in order to determine if the tool in question is able or not, and we have provided a preliminary evaluation of some tools.

Between all the authoring tools, as is a wide application field, we focus on the <e-Adventure> authoring tool to explain how we address the solutions for each guideline, although it is a software also developed in Java and it would be necessary incorporate accessibility features to the various Java controls, by using the libraries Java.swing and using the JAB as we have said in section 2.3.1.

For each guideline proposed in section 3.2, we describe the current situation in this tool and we describe a solution to the problem if needed. This is completely described section 5, as in the next section we make a review of the <e-Adventure> approach and its games.

4. THE <E-ADVENTURE> PLATFORM

<e-Adventure> is an authoring tool for creating educational video games and simulations, developed by the <e-UCM> research group at Complutense University of Madrid (Javier Torrente et al., 2010). The aim of <e-Adventure> is to facilitate the development of video games for people without a technical background (i.e. mainly teachers or students). In addition <e-Adventure> is multiplatform (developed in Java), allowing the export of the games to run on Windows, Linux, Mac or even a web browser.

4.1 <e-Adventure> Games

As a strategy to reduce the development costs and the complexity of the authoring tool, <e-Adventure> focuses on the genre of point-and-click adventure games, like the Monkey Island™ or Myst™ sagas (Moreno-Ger et al., 2008; Moreno-Ger et al., 2007a).

Therefore <e-Adventure> games are composed by scenes (made up by a background image) which are interconnected (as defined by the game author) resulting in a 2D navigational environment. In this environment the player has to solve the challenges that game author defines.

Figure 7. Screenshot of a first-person <e-Adventure> game. The cursor changes when the mouse is over an interactive element.

<e-Adventure> supports two kinds of games: first-person games, where players just perceive the game as if they were the main character (Figure 7), and third-person games, where players control an avatar that takes the role of the main character in the game (Figure 8). Nevertheless, the player-game interaction is the same in both cases. In <e-Adventure> games, to discover the elements that players can interact with they need to explore the scene using the mouse. When the mouse slides over an interactive element the image of the cursor changes and the name of the element is displayed, providing in this manner visual feedback to the player. For instance, in Figure 7 the mouse cursor becomes an “eye” when it is pointing to the gloves.

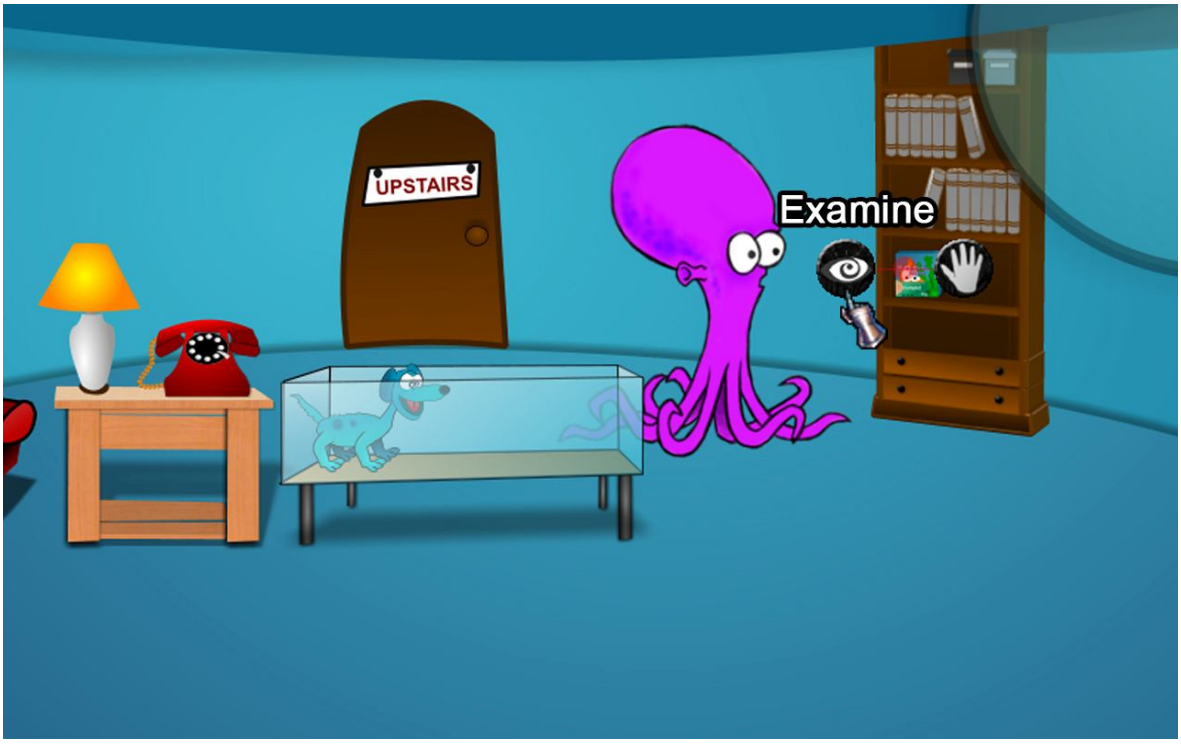


Figure 8. Screenshot of a third-person <e-Adventure> game. It shows a two-button contextual menu with the available interactions over an object.

When players click the mouse on an interactive element a contextual menu appears, including the different actions that can be performed with it. For instance, Figure 8 shows a screenshot of a third-person game where the player can examine (eye button) or grab (hand button) an object (the pet food in this case).

4.2 Introducing Accessibility in <e-Adventure> Games

The first approach to the introduction of accessibility in <e-Adventure> was focused on the games instead of the authoring tool.

The platform has several features that provide accessibility, such as the presence of instructors that enhance the possibilities to cope with the challenges, the choice of the point and click genre that is especially focused on the generation of accessible games, the adaption of the contents to provide different levels of knowledge and the adaption of the interaction channels to engage also users with special needs.

As it is described in (Javier Torrente et al., 2009), once the game is designed the author of the game can turn on/off, a set of input and output modules and other support tools (e.g. a screen magnifier) that are already built into the platform and which will be available to the player embedded in the game metaphor. Using this mechanism, the authors will be able to incorporate accessibility features when creating the game, which means that the currently game created, is going to present the same features for all users.

Also the author can enable certain features, such as a screen magnifier if she detects the users may need it, apart from the modules that are predefined.

Currently <e-Adventure> has been powered with three different input mechanisms as Figure 9 represents: a Mouse Interface Layer, a Voice Interface Layer and a Keyboard Interface Layer. The Mouse Interface provides the classic point-and-click interaction through the mouse (as described in section 4.1). Nevertheless, the Voice Interface and the Keyboard Interface provide new interaction mechanisms that are especially suitable for people with motor or visual disabilities.

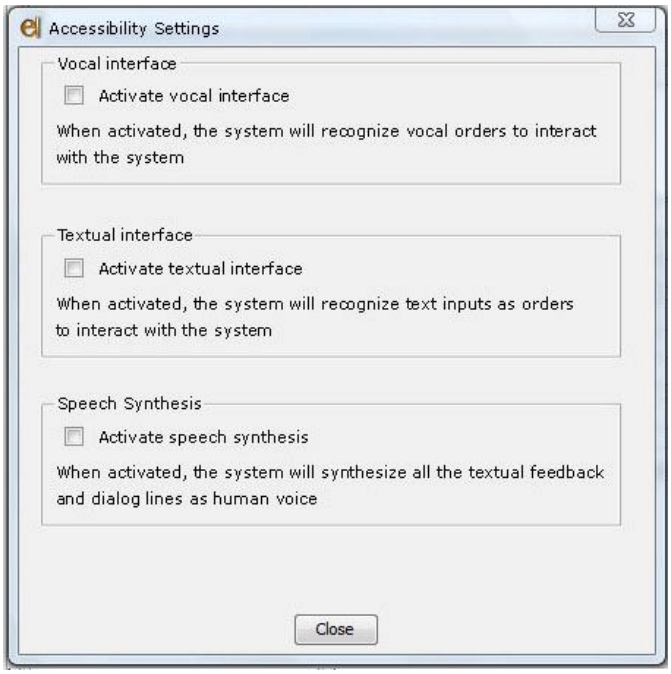


Figure 9. Screenshot of the input/output mechanisms.

On the one hand, users who cannot move a computer mouse can interact with the game through voice commands using the Voice Interface. On the other hand, users who are blind or have a severe visual disability, for instance, those that are used to interact with computers using the keyboard, profit from the Keyboard Interface which allow them to interact by introducing text commands. Both interfaces recognize the same commands in natural language but using different input devices.

A typical command matches the structure Verb + Article (optional) + Element + Conjunction (optional) + Element2 (optional). For instance, valid commands are "Describe the scene" or "Use key with locker".

Additionally <e-Adventure> games can also be enhanced with extra sonorous feedback for people with visual disabilities, including a speech synthesizer that transforms all text lines that present information in the game to voice and descriptive sounds, for example beeps, or common actions (e.g. enter/exit the game menu). Hence, the platform presents three output modules: the visual module, the sound module and the speech synthesis module.

Moreover, <e-Adventure> presents a dynamic adaptation to achieve an adaptation more accurate to each user according to his needs and skills. Thereby, to deal with cognitive disabilities deeper changes and adjustments affecting the game flow may be required. To support this kind of dynamic modifications, <e-Adventure> includes a game flow adaptation engine that can modify the game experience according to

characteristics of the player by the activating or not various flags, for example by skipping some initial levels, avoid puzzles that are too challenging, etc. (Moreno-Ger et al., 2007a; Moreno-Ger et al., 2007b). Other users with motor or cognitive disabilities may require adjusting the response times in the interaction with the game, such as for example, increase the time needed to complete a puzzle or answer a question.

All these accessibility features can be introduced in the games using the game authoring tool, which reduces the cost of introducing accessibility in educational video games. Nevertheless, the authoring tool should be also accessible following the ideas of the ATAG, IGDA or IBM checklist and adapting them to our needs.

4.3 The <e-Adventure> Authoring Tool

As **¡Error! No se encuentra el origen de la referencia.**10 depicts, the <e-Adventure> authoring tool is a GUI-based editor structured in three different areas, we could see it following the arrows in the figure: a toolbar on the upper section, a panel to access the main elements that make up a game on the left section (e.g. the scenes that make the virtual world, objects and characters, etc.) and an edition panel on the central section to configure the properties of the element selected on the left panel.

The edition panel is different for each kind of element, but is generally made up of tabs, tables and nested sub-panels in which the editable properties of each element are grouped. Therefore, navigation through this interface using the keyboard is hard as all the elements of the GUI must be wandered in a linear sequence.

The configuration of the scenes is done through several drag and drop panels like the one displayed in **¡Error! No se encuentra el origen de la referencia.**10. When a scene is selected the central panel is loaded with the background image of the scene. Upon this image, the game author can drag characters, and items onto the scene. Following a similar idea the game author can define interactive regions on the scene (i.e. portions of the background image that are defined by polygons) which must be placed using the mouse.

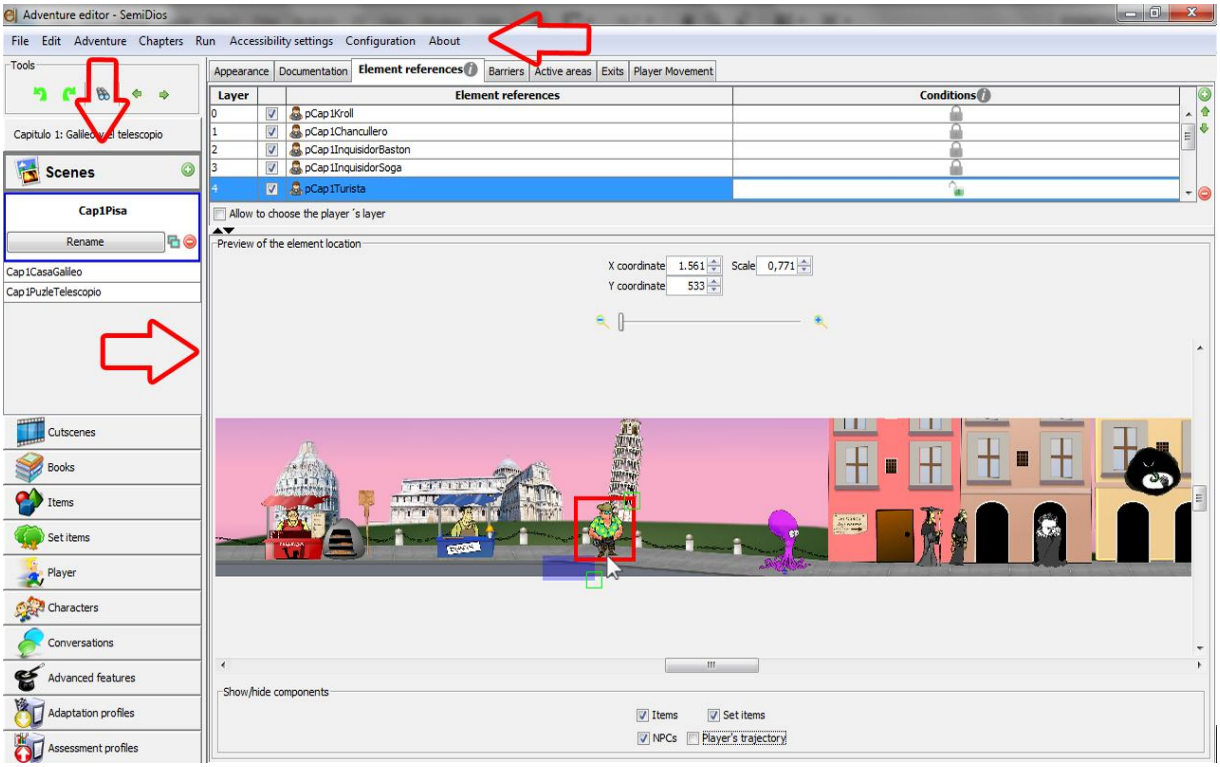


Figure 10. Screenshot of the configuration panel to place the elements on a game scene. It is shown through arrows the main sections of the editor.

5. INTRODUCING ACCESSIBILITY IN THE <E-ADVENTURE> EDITOR

The <e-Adventure> editor exemplifies many of the features that hinder the introduction of accessibility in this type of software. Among the most common problems, we highlight the configuration of visual elements that make up the virtual world of the game, and the way to structure the editable information such as the different and various setting options the tool presents.

Our motivation is to attempt to develop a holistic platform in which accessibility is addressed as a whole, considering not only accessibility in games but also in the authoring tool. To achieve this ambitious process we focus on the guidelines depicted in section 3.2 to test if our authoring tool fulfill with them.

In this section we present our approach resulting of the survey made on the <e-Adventure> tool and we provide solutions to accomplish the guidelines proposed. We subdivide this section to group the possible solutions according to topics as we have made with the guidelines.

One of the most challenging tasks is the configuration of visual elements that poses a serious accessibility problem, especially for people with visual disabilities since the game author must be able to perceive the virtual world and the position of the various elements that compose the world as a whole, (not in isolation) without the support of vision. One way to facilitate the configuration of the game universe is to use a 2D representation instead of a 3D view which is more usual in modern video games. However, it is still necessary to solve the problem of defining the features to place the elements on the scenes of the game that are accessible for people with visual disabilities. This problem occurs recurrently in the <e-Adventure> editor where the panels that allow the configuration of the scene are based on drag and drop interaction.

Another challenging issue is the way to structure the large amount of information that is needed to configure the game so it is accessible through the keyboard.

Thereby, in the next subsections we address the problems founded and the possible solutions not only focusing in these two challenging issues, but also in the rest of the guidelines to propose a holistic solution. Not all the guidelines are already developed, instead, some of them are currently under development for the <e-Adventure> 2.0 version.

We follow the guidelines proposed in chapter 3.2 to tackle with the lack of accessibility in the editor and we include also Java code to show how

5.1 Object information

This section especially addresses how developing a good programming to facilitate the task to assistive tools such as incorporate label next to components, provide a description of each object or provide the situation of the focus.

One of the major problems that pose the inclusion of accessibility for people with visual disabilities is to provide an alternative (multimodal) return of information. For this purpose, we use a speech synthesis module using the library of Microsoft speech synthesis (Speech API 5) which is invoked through Com4J library that allows calls to COM interfaces Microsoft from Java code as we have depicted in section 2.5. Through this system the user can configure the speech synthesis module using external assistive

tools, like the accessibility features included in Windows Vista (or higher) or the screen reader Jaws.

For example, a navigation structure as complex as the one presented in Figure 10, makes it hard to remember what part of the information is being edited, that is, what is the active focus loop. As a solution, we have thought to add a special command that indicates via speech synthesis a summary of the current location within the focus tree.

Similarly, the editor formerly made extensive use of visual feedback which requires further adaptation effort. In this sense, we have thought to include also special sounds associated with relevant system events, such as Image loaded successfully (or not), Image linked successfully (when you configure the navigation through scenes) or Element referenced in the image when an object or character has been successfully placed in the scene.

Usually, typical SWING controls can be accessed through a screen reader, just it is important to consider some simple recommendations:

- In those controls (labels, buttons, frames etc.) instead of textual content have an associated image or any other type of content, it is imperative to set the necessary properties of accessibility for readers to verbalize an alternative description in order to provide the user with visual impairment equivalent information.
  In this first case, we define the description through the method setToolTipText which enables the description for both disabled and non-disabled users.

```
JButton button= new JButton(“Access”);
button.setToolTipText(“Button accessible description”);
```

  In addition we have also the possibility to use the method setAccessibleDescription for the components (JFrame, JApplet) that do not inherit from `JComponent`.

```
JButton button= new JButton(“Access”);
AccessibleContext context = button.getAccessibleContext();
contex.setAccessibleDescription(“Button accessible ” +
“description, but not the tool tip”);
```

- It is important provide not only an alternative text in that elements which have an image, but also provide an accessible name for all the components that must be reached by the focus:

```
JPanel testPanel = new JPanel();
AccessibleContext context = testPanel.getAccessibleContext();
contex.setAccessibleName(“Test Evaluation”);
```

- It must always respect the tab order of controls on a form, as the visual appearance thereof. This will impact positively on the navigation of the people

who uses screen reader, because it moves with tabs, the application will run in the visual order of the same.

- Focus should be given only to the components that users interact with or to those that are being read by the assistive tool. The default focus that Java provide, already meet these needs, and it is necessary to modify it if the programmer need to add new characteristics.
- Customized components should support accessibility by implementing the `Accesible` interface and redefining the method `getAccessibleContext()`. Most Swing components meet this need, so, customized components must meet it as well.
- Use standard controls for the operations for which they were written. Often, by design, could be tempted to use, for example, a label such as a button, because for purely design issues, would not have the slightest importance. However, for a screen reader, this is really important: first, could not be accessed with the tab at the label, it does not accept the focus and therefore could not even execute the function associated with mouse click that label (the buttons, click this function is equivalent to pressing "Enter" on the keyboard once the button has the focus).

5.2 Images and Color

This section tackles with issues of enabling the customization of colors by making sure that it is provided the possibility of picking colors contrasted enough. Also be aware of not conveying information through colors and for images always be careful of including an alternative description.

Regarding colors, it is advisable to provide much more flexibility.

On the one hand, allow choose the font color, allow the choice of colors of the panels background. In this manner, users can adjust the color that they best perceive, and consequently, it would be good also, to allow the choice of colors of the panel edges to differentiate each section.

On the other hand, focusing on buttons, it would be advisable to define a color users could change, as well as for the eyelashes, distinguishing between those which are hidden and the one which is open.

In the forms, or simply in the fields where is necessary to include text, it can be defined a base color that brightens when the focus get it.

To provide textual description for images, as `ImageIcon` objects, it is necessary tho define the method `setDescription()`.

Due to the multiplatform nature of <e-Adventure> we initially have chosen the alternative of adding an “Accessibility" drop-down menu in the toolbar, where the user can select the display mode, normal or high contrast. Moreover, in this menu we also include the options that allow the configuration of other accessibility features of games, depicted later.

5.3 Multimedia Content Accessible

As it is necessary the introduction of accessibility in the editor and in games in general, we need contents to process them in an accessible format also. This means that unless we have the resources properly tagged, depicted and with the multimedia content in various format, we will not be able to develop an accessible game although we have addressed accessibility questions either in the editor or games.

To attempt this point, each resource are two files, one is the multimedia content either an image or a video, and the other is an xml file, both of them called with the same name, which has all the information about the image or video.

As we can see below we include the tag image with the attributes number of tags that the image in question has, a description of the whole image and the path where is located the image.

Then, we present the succession of tags which must agree with the number introduced in the attribute `number_tags`.

Each tag is composed by the tags name, description of the tag and points that has the tag in question which always presents the attribute number of points.

This means that a tag can be defined by one or more points, but at least it has to present one. If it presents more than one, they must be added in order according on how they are presented in the image, not randomly.

Hence, following this structure we compose the xml file, which must be validated in order to make sure the constraints are fulfilled.

Thus, the xml file “background.xml” we have designed for the multimedia content “background.jpg” is the next:

```
<image number_tags="6" description_image="Background image description"
path_image="C:/accessibility/resources/background.jpg">
      <tag>
            <name>Bed</name>
            <description>Bed Tag Description</description>
            <points number_points="1">
                  <point x="130" y="260"/>
            </points>
      </tag>
      <tag>
            <name>Bookcase</name>
            <description>Bookcase Tag Description </description>
            <points number_points="4">
                  <point x="665" y="25"/>
                  <point x="665" y="230"/>
                  <point x="755" y="235"/>
                  <point x="755" y="15"/>
            </points>
      </tag>
      <tag>
            <name>Table</name>
            <description>Table Tag Description</description>
            <points number_points="1">
                  <point x="320" y="240"/>
            </points>
      </tag>
      <tag>
            <name>Chair</name>
            <description>Chair Tag Description </description>
            <points number_points="1">
                  <point x="900" y="300"/>
            </points>
      </tag>
      <tag>
            <name>Telephone</name>
            <description>Telephone Tag Description</description>
            <points number_points="1">
                  <point x="330" y="200"/>
            </points>
      </tag>
      <tag>
            <name>Sofa</name>
            <description>Sofa Tag Description</description>
            <points number_points="1">
                  <point x="1100" y="400"/>
            </points>
      </tag>
</image>
```

5.4 Configurable Fonts

Usually is one of the first strategies that it is followed to improve the accessibility of any software tool.

In the <e-Adventure> editor, we have allowed the configuration of some parameters of the Graphic User Interface, as the text size. As discussed in previous sections this is one of the most common strategies in terms of accessibility as it can solve some the most

common problems. However, there are several approaches for providing solutions for the configuration of the user interface.

On the one hand, one option is to give the user the possibility to choose the settings manually by the use of the “Accessibility" drop-down menu in the toolbar, where we add the font size, either normal or large.

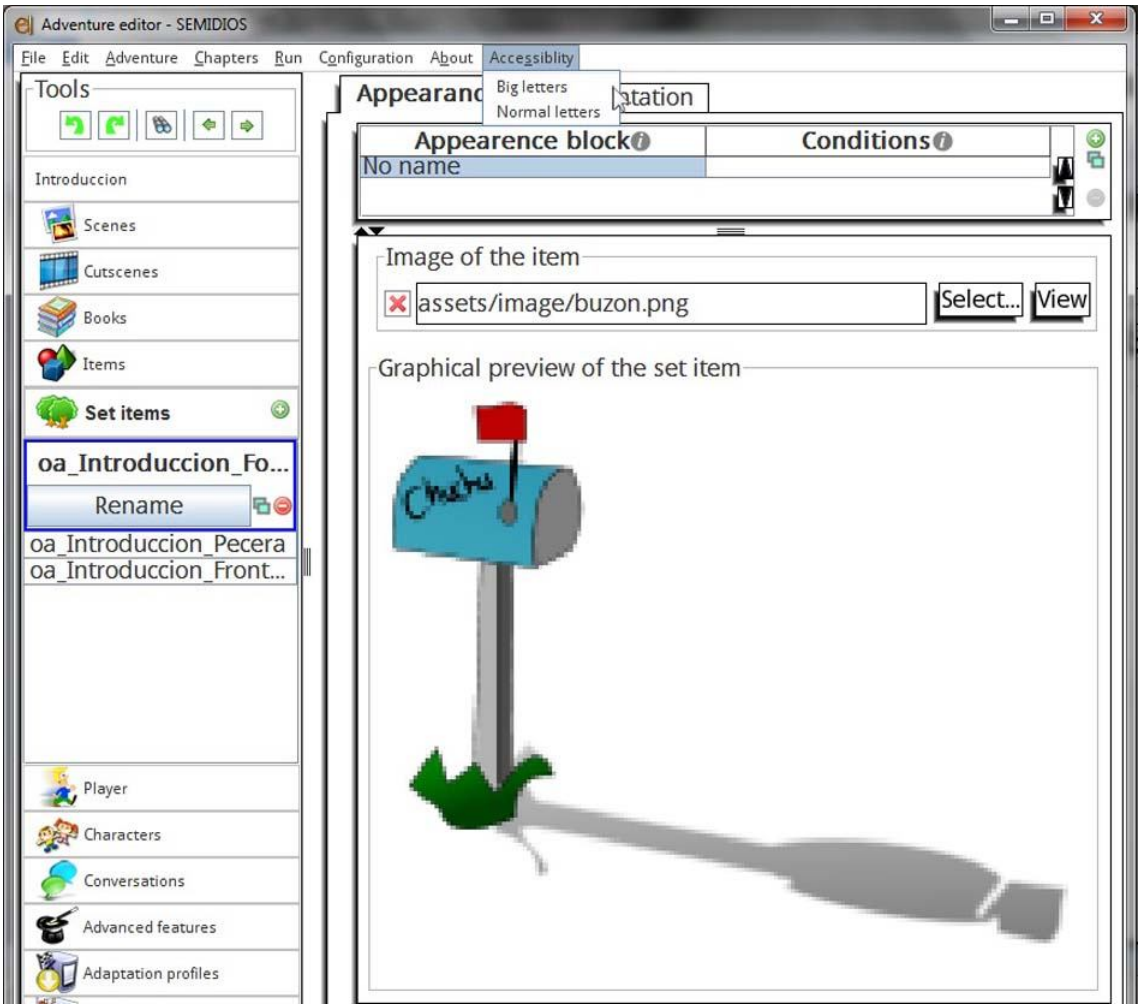


Figure 11. Screenshot of the Accessibility section to change the size of the fonts.

On the other hand there is the option to automatically load the settings according to the user profile that is stored in the configuration of the OS.

5.5 Keyboard Access

This section addresses the navigation through the application by using the keyboard. This feature allows the possibility to move over every control without using the mouse which is useful for users who are visual disabled or who have any kind of motor impairment.

Thus, to make the software as accessible as possible we enable the typical fast access to go to the menus of the toolbar by using the convention ALT + letter:

```
JMenu menuFileA = new JMenu("Accessibility settings");
menuFileG.setMnemonic(KeyEvent.VK_G);
```

If the GUI presents a `JTextField it` is not possible to assign a hotkey to them directly, we need to create a label and link it to the text field using the `setLabelFor` method as follows:

```
JLabel label = new JLabel(“First access”);
JTextField accessText= new JTextField();
label.setLabelFor(accessText);
label.setDisplayedMnemonic(KeyEvent.VK_A);
```

And we enable also the shortcuts CRTL + letter to access any action that is commonly used and we want to access it as faster as possible without making a series of steps. We must use the next commands:

```
JMenuItem menuItemL = new JMenuItem("Big Letters");
menuItemL.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_B,
ActionEvent.CTRL_MASK));
```

In our case, this option is useful when users want to edit an element that is far in the structure from where they are in that moment. This section is closely connected with section 5.8 where we talk about the structure that it is used behind the GUI. So, we explain it in-deeper in that section also.

5.6 Use Simple and Concise Language

This section is especially addressed to the way in which we convey the information. For achieving this, we incorporate the presence of icons to support the words in the menus, which gives an idea without reading the text of what we are dealing with and we highlight also the words enabling various levels of text, as we can see in Figure 12.

When explanations are needed, we use a familiar language without hard and large descriptions and without subordinate sentences.

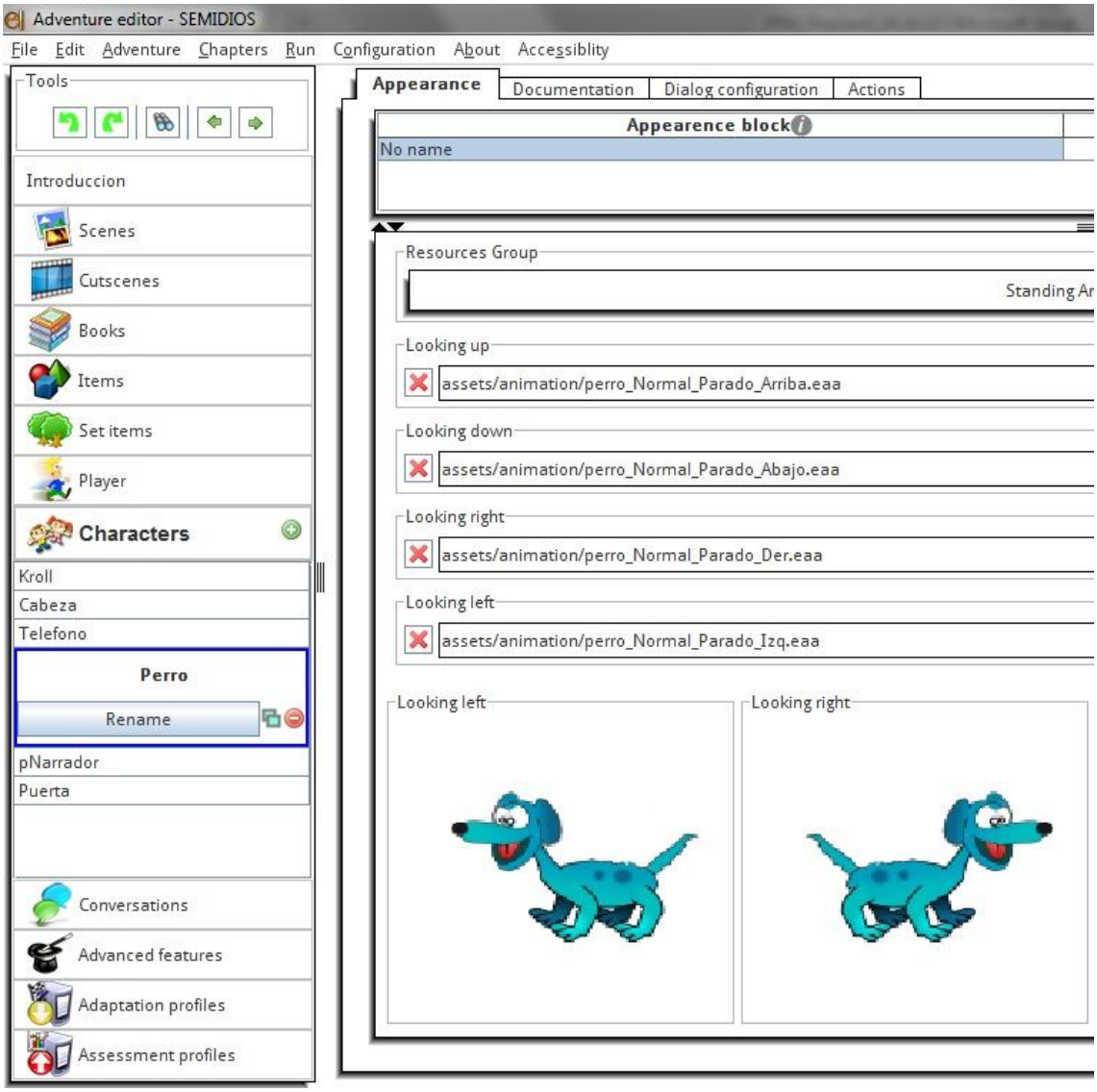


Figure 12. Screenshot showing the use of simple sentences and use of complementary icons in sections.

5.7 Setting the Scenes in Alternative Way

Another relevant problem highlighted in the introduction of this section is how configure the visual elements that make up the virtual world in an accessible way for people with visual disabilities. To achieve this we are developing an alternative interaction to typical drag and drop interaction.

The goal is to provide aids for people with visual disabilities that allow them to place elements in the scene as a whole. That is, that a visually disabled user could easily place an element in the scene taking into account both the position of other interactive elements and what the background image of the scene looks like. For instance, if there is a door drawn in the right side of the scene, the user must be able to place an interactive rectangle over this area of the scene and link it to other scene.

We have taken several actions to facilitate this issue.

First of all, elements in drag and drop panels are also accessible using the keyboard as they have been added to the focus loop tree described in 5.8. Once the drag and drop panel that contains the elements of the scene is reached, users can add and remove elements and modify the location of the elements in the scene using the arrow keys.

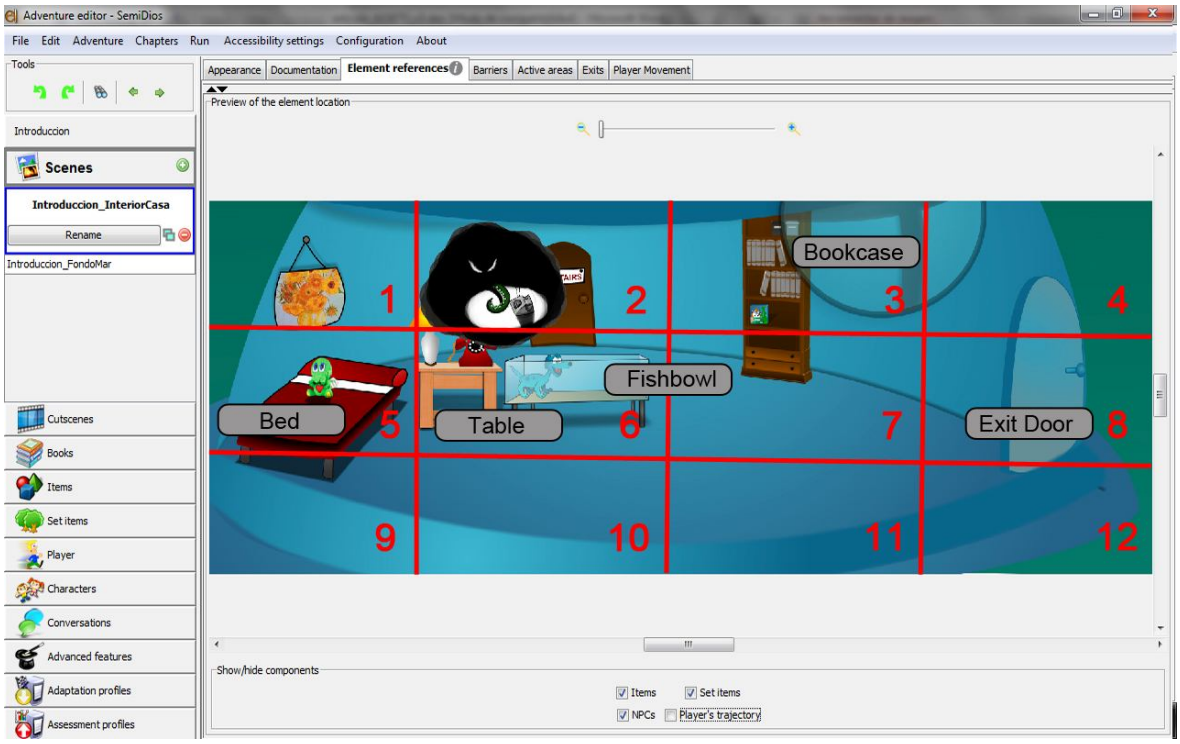


Figure 13. Screenshot of the configuration panel showing the imaginary divisions and the tags enabled in background image.

Second, we have added two special aids that facilitate the placement of elements in the context of a specific scene. On the one hand, we have developed a tags system to attach additional accessibility meta-data to the background images that will be used to configure the scenes of the game.

These tags must be described in the xml file we have shown in section 5.3.

A tag is a special mark that is attached to a background image and indicates the location of the elements that compose the image (see Figure 13). The idea is that artists could annotate the position of relevant elements in the background images, by defining at least one point, which would be the middle point or more points that define the outline of the element.

Besides a general alternative description of the scene for the speech synthesizer is also provided in the xml document to give a general idea of what image is about.

The <e-Adventure> authoring tool allows users to locate elements in the scene in relation to the position of the tags and other elements of the scene using special keyboard shortcuts. For example, in Figure 13 users could execute actions such as “place element on the left of the door” or “place element behind the bed”.

On the other hand, we are developing a system of twelve imaginary regions that divide the scene (see Figure 13). The 12 regions are fixed for all scenes and games. Therefore they can be used to place elements in the scene when no tags have been provided.

Finally, when users have selected a specific element of the scene (e.g. a character, an item, an interactive area, etc.) they can use a special command that provides the position of the element in relation to other elements through the speech synthesizer.

5.8 Structuring the Information & Keyboard Access

To cope with the need of structuring the information of the editor in an accessible way we have followed two main strategies.

First, we have reduced the number of nested elements of the editor, for instance replacing tables by more accessible components. This allows navigating throughout the interface more easily using the keyboard. See Figure 14.

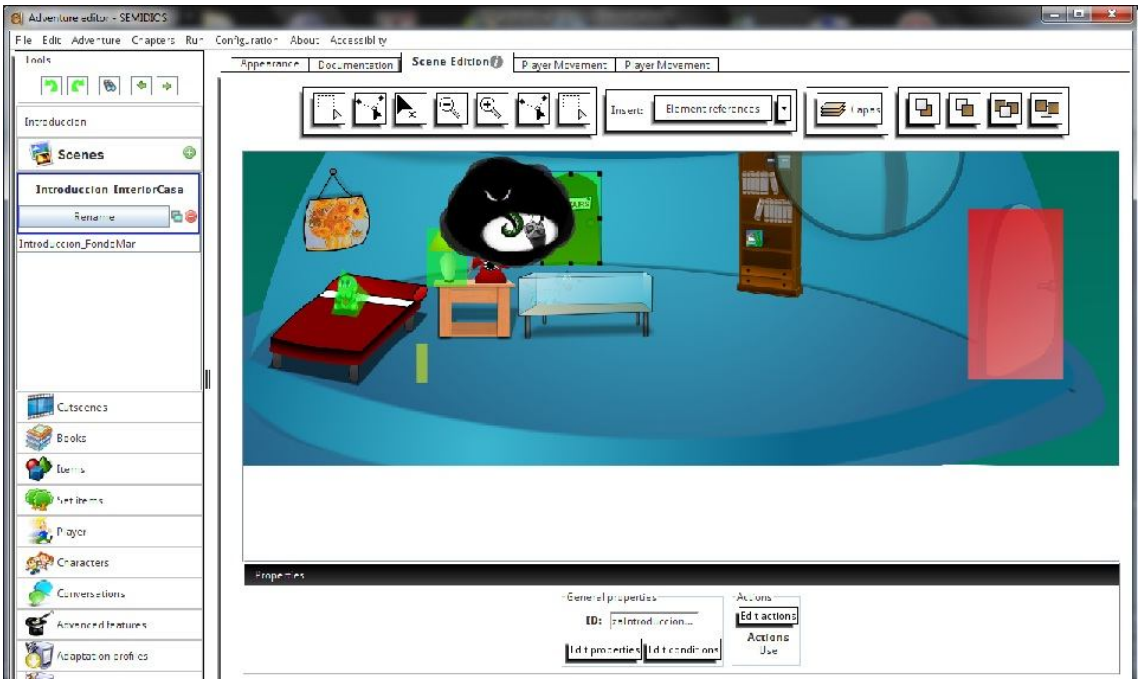


Figure 14. Screenshot of the new structure without tables.

Second, we are developing a tree structure of "focus loops" that improves further the navigation using the keyboard. A focus loop sets the order in which the elements of a specific part of the interface can be inspected with the keyboard. We have thought about what convention is the most appropriate to make the navigation as intuitive as possible.

Finally, we have decided to use the Tab key to cycle between elements within the active focus loop, which means a tree level. When one of the interface elements consists of a larger group of components, the user can go down in the tree and start exploring the components of this element using the CRTL + TAB combination, changing the active focus loop accordingly. To return the focus to the upper level the combination CRTL + SHIFT + TAB can be used.

However, since navigation through the entire focus tree is complex, we have also included keyboard shortcuts to go directly to certain points of the "focus tree". This option is useful when users want to edit an element that is far in the focus loop tree from the active focus loop.

For instance, when users are placing the elements on a specific scene they can jump directly to the left panel using the shortcut CRTL + LEFT ARROW and start editing other game elements, instead of going backwards in the tree using CRTL + SHIFT + TAB.

5.9 Documentation

In this section we aim to tackle with the need of including in the menu bar a section that includes help with a manual or any kind of documentation that explains the way in which the software tool is working. Usually for acceding to this help, it is used by convey the key F1.

To develop an accessible PDF manual it is advisable that the document must be computer-generated text and fulfill these premises:

- The PDF document must be a label ("Tagged"): A tagged PDF document is a PDF version that includes both the content of the document and information about its logical structure and about reading order so that the document can be read correctly by screen readers.

  To make a well labeled paper is to use structural elements such as headers, footers, headings, bullets, and other labels and styles, rather than just a visual work on the text (bold, font size, line blank to separate paragraphs, etc.).
- Alternative text for images: For images it is mandatory to provide descriptive text (alternative text) that can be read to provide the user with information about the image.
- Language of document: Screen readers can read documents in different ways according to the language used. Therefore it is necessary to specify the language of the document to make it accessible.
- Security available: A document with security options (for example, prohibit changes to the document as a copy or paste) may be inaccessible if security is not configured properly.
- Links: The document should be well defined and useful links to other parts of the document so that a user can go directly to the section you want to visit.
- Navigation: We should provide aid to navigation and organization as a table of contents, bookmarks or headers to facilitate the user to move around the document without having to read all the content to find what you are seeking.

5.10 Conclusions

Summarizing, in this section we have discussed the possible solutions that would be necessary to develop to incorporate accessibility in the <e-Adventure> authoring tool.

Some of them are quite accepted in the Web accessibility field and they have been re-adapted to be used in this scope while others have been developed for this specific scope.

The <e-Adventure> interface development tool has the potential to improve in an easy way accessibility features by incorporating the various accessibility statements in Java controls present in bookstores Java.swing.

In conclusion, we thought that incorporating the majority of the presented features, the <e-Adventure> authoring tool could create games that incorporate adequate alternative access mechanisms.

These mechanisms will provide accessible software tools for users with motor disabilities, hearing and partially sighted. For users with blindness, this challenge would be more complex and we should introduce also substantial improvements.

Therefore, the main enhancements to incorporate in <e-Adventure> would be:

- Support for speech synthesis.
- Support for voice recognition.
- Navigation through keyboard.
- Well structuring of the settings through nested menus.

6. DISCUSSION AND FUTURE WORK

This work started with the ambition of exploring the adaptive gaming and accessible gaming fields from a new perspective. We have discussed that the introduction of new technologies in education must take into account the accessibility aspects for both students and instructors. There are still many open accessibility issues to be addressed to introduce highly visual and interactive technologies in the educational system in an inclusive manner. We have addressed these accessibility problems in video games and video game authoring tools by introducing accessibility features in the <e-Adventure> platform.

In this chapter we will discuss the conclusions derived from this work, summarizing the contributions, as well as outline future lines of work.

6.1 Conclusions

The main challenges and conclusions we have identified are:

There is a need for intensive research in the accessible gaming field and accessible authoring tool field. Accessible gaming can become a popular trend based on the potential benefits of applying adaptive gaming to support learning. Nowadays, there is a fair knowledge about how to design adaptive game-based learning experiences that guarantee a high educational value, although there is a lot of further research required. The main challenge we face is to avoid the increase in the costs of creation of these games, especially in the education field where budgets are usually more restricted. That is the reason why it is interesting the use of authoring tools to reduce the cost of, first of all, the stage of game creation and furthermore the introduction of accessibility in an inclusive way in games.

The need of including game accessibility features from an editor, lead us to demand the presence of accessibility features in the editor itself too. The conclusion is that the introduction of accessible features in the authoring tool is still confusing since there are no standards that point out a clear and accepted methodology to develop these authoring tools in an accessible way.

- Lack of awareness in the gaming community about the need of tending to the requirements of people with special needs. The gaming industry has been completely unaware about the need of making games that are playable by people with special needs. Thanks to initiatives of different condition (e.g. legal initiatives, communities of disabled gamers such as AbleGamers.com, associations of developers such as the IGDA, etc.) the awareness of this problem is increasing. Nonetheless, the current approaches are especially focused on developing accessible games, and there are no comprehensive approaches that attempt to develop highly visual tools, such as game authoring tools, in an accessible way.
  In our opinion the approaches proposed for developing “accessible” authoring tools are missing important details due to the excessive influence that web based accessibility methodologies have on them. When accessibility is related to software, the goals vary depending on what the software is intended for.

In this regard there are clear differences between web-oriented authoring tools, text-oriented authoring tools or high visual-oriented authoring tools. Thus when discussing accessibility in highly visual-oriented authoring tools the goal should be to guarantee that the creation process of any kind of content can be done by everyone who attempts it without feeling frustration or being completely lost.

Nowadays, to achieve for accessibility to be addressed from the point of view of adaptation in video games from the authoring tool that creates them is a challenge. Taking in consideration the additional functional requirements accessibility involves, there is a need to provide a flexible design of the authoring tool and a good interaction with assistive tools that give support to multiple software.

The introduction of accessibility in an authoring tool as <e-Adventure> raises two different challenges. First, to make accessible the <e-Adventure> editor, which is a highly visual, drag and drop desktop authoring tool. Second, to make the content the tool produces (the video games) accessible by including easily configurable features in the platform that facilitate the introduction of accessibility in the games with a marginal extra cost.

- Need of regulation issues. Several countries have regulations tackling the accessibility of software for disabled people in public administrations. As we have seen in section 2.1, the presence of this kind of laws in every country encourage large organizations and private corporations to follow similar internal regulations for protection against lawsuits and to comply with the standards of selection of contracts.
  The challenge in this scope is to achieve that not only the United States and a few others countries pass these laws, but also every country which is concerned about the complete integration of every citizen. It would not be surprising to see more stringent regulations becoming law in the future; therefore, planning for accessibility in any kind of software private industry will be prepared to fulfill this kind of regulations.

Considering the targeted challenges of 1) the need of intensive research in the field, 2) the special consideration that accessibility deserves in games authoring tools and 3) the need of regulation issues, we have proposed a methodology that attempts to meet the challenges that are posed nowadays.

6.2 Contributions

Following the above-mentioned premises, we have carried out this work producing some interesting outcomes that can be applied in real contexts thanks to the application over the <e-Adventure> authoring tool, which will facilitate future research.
Summarizing, the main contributions of this work would be the following:

- To have reviewed and analyzed three active research topics: accessibility in video games authoring tools, accessibility in games, the need of regulation issues and identifying challenges and motivations to achieve these three topics to interact in a proper way, where the fields involved could profit from mutual cooperation.
- To have proposed how that cooperation could be established linking the three areas to achieve a holistic accessible platform.
- To have developed an approach that integrates concepts from composing scenes in an accessible way as we have incorporated in the guideline.
- To have particularized the methodology for a specific scenario, <e-Adventure>, considering adaptation to deal with special requirements.
- To have implemented various approaches of the methodology in the <e-Adventure> educational game authoring tool, as part of the objectives of the GAMETEL research project where the <e-UCM> research group is involved, allowing the development of accessible educational video games authoring tools.
- To have embedded explicit support for accessible design in <e-Adventure>. Dealing with special games is costly as it requires investing in expensive technologies such as text-to-speech or voice recognition. Thus to have an authoring tool that has these features embedded out-of-the-box may help in reducing such high development costs.
- The last remarkable contribution are the results of this investigation: a research paper that will be published in an national workshop, which describes how the <e-Adventure> editor supports the development of educational games that cater for the special needs of students with visual, hearing, mobility and cognitive impairments, without adding an extra development cost.

6.3 Future Work

Despite the work done, this is just the beginning of an ambitious line of research that we expect will generate positive results within the next few years. This work has been useful to understand the complexity of the fields involved and to develop strategies for approaching the next steps of this research. Therefore these are the next issues we are planning to address in the near future:

- The introduction of accessibility in <e-Adventure> has focused on a small subset of primarily visual disabilities. Next versions of the platform will pay special attention to hearing, motor and cognitive skills impairment.
- Although we have proposed a general methodology, we have specifically applied it to the <e-Adventure> editor tool. In next iterations, the generalization to a broader sector would be advisable, including a comparative of several tools
- Revise and refine the concepts of our general methodology by developing new guidelines and testing them, when the evolution of technologies requires it.
- To develop a general accessible model for educational authoring tools. This model will include how to design and implement adaptive educational game authoring

tools for improving the learning outcomes of diverse students according to their specific needs.

As a last remark, we would like to emphasize that this is just the beginning of a promising research line that we expect will deliver adaptable, accessible and enjoyable game-based learning experiences from the point of view of creating games through authoring tools.

REFERENCES

Argue, R., Boardman, M., Doyle, J., & Hickey, G. (2004). Building a Low-Cost Device to Track Eye Movement.

Bierre, K., J. Chetwynd, B. Ellis, D. M. Hinn, S. Ludi, T. Westin. (2005). *Game Not Over: Accessibility Issues in Video Games.* Paper presented at the HCII.

Bierre, M. H., T Martin, M McIntosh, T Snider. (2004). *Accessibility in Games: Motivations and Approaches* International Game Developers Association (IGDA).

Csikszentmihalyi, M. (1991). *Flow: The Psychology of Optimal Experience*: {Harper Perennial}.

Dickey, M. D. (2006). Game Design Narrative for Learning: Appropriating Adventure Game Design Narrative Devices and Techniques for the Design of Interactive Learning Environments. *Educational Technology Research and Development, 54*(3), 245-263.

Doush, I. A., Pontelli, E., Simon, D., Son, T. C., & Ma, O. (2009). Making Microsoft Excel™ Accessible: Multimodal Presentation of Charts. *ACM SIGACCESS Conference on Computers and Accessibility, Assets*.

Fernández-Manjón, B., Sierra, J. L., Martínez-Ortiz, I., & Moreno-Ger, P. (2009). Estandarización y Modelado Educativo. Informe nº 20. *Instituto Superior de Formación y Recursos en Red para el Profesorado (ISFTIC) (In Press)*.

Foss, B. A., & Eikaas, T. I. (2006). Game Play in Engineering Education— Concept and Experimental Results. *International Journal of Engineering Education, 22*(5), 1043-1052(1010).

Grammenos, D., A. Savidis, & C. Stephanidis. (2007). Unified Design of Universally Accessible Games. In S. B. Heidelberg (Ed.), *Universal Access in Human-Computer Interaction. Applications and Services* (Vol. 4556/2007, pp. 607-616).

Hetzroni, O. E., & Shrieber, B. (2004). Word Processing as an Assistive Technology Tool for Enhancing Academic Outcomes of Students with Writing Disabilities in the General Classroom. *Journal of Learning Disabilities, 37*(2), 143-154.

Hitchcock, C., & Stahl, S. (2003). Assistive technology, universal design, universal design for learning: Improved learning opportunities. *Journal of Special Edcuaiton Technology, 18*(4), 45-52.

Houlette, R. (2004). Player modeling for adaptive games. In *AI Game Programming Wisdom* (Vol. II, pp. 557–566): Charles River Media.

Jiménez Peñuela, J. R. Desarrollo de Sitios Web accesibles a la Población Sorda.

Kearney, P. R. (2005). *Playing in the Sandbox: Developing games for children with disabilities*. Paper presented at the DiGRA 2005 Conference: Changing Views - Worlds in Play. Retrieved 30th May, from http://www.digra.org/dl/display_html?chid=06278.12473.pdf.

Magerko, B. (2008). Adaptation in Digital Games. *Computer, 41*(6), 87-89.

Moreno-Ger, P., Burgos, D., Sierra, J. L., & Fernández-Manjón, B. (2008). Educational Game Design for Online Education. *Computers in Human Behavior, 24*(6), 2530-2540.

Moreno-Ger, P., Burgos, D., & Torrente, J. (2009). Digital games in e-Learning environments: current uses and emerging trends. *Simulation & Gaming, 40*(5), 669-687.

Moreno-Ger, P., Sancho Thomas, P., Martínez-Ortiz, I., Sierra, J. L., & Fernández-Manjón, B. (2007). Adaptive Units of Learning and Educational Videogames. *Journal of Interactive Media in Education, 2007*(05).

Moreno-Ger, P., Sierra, J. L., Martínez-Ortiz, I., & Fernández-Manjón, B. (2007). A Documental Approach to Adventure Game Development. *Science of Computer Programming, 67*(1), 3–31.

Overmars, M. (2004). Teaching Computer Science through Game Design. *IEEE Computer, 37*(4), 81-83.

Röber, N., M. Masuch. (2005). *Playing Audio-Only Games: A Compendium of Interacting with Virtual, Auditory Worlds*. Paper presented at the Digital Games Research Association Conference (DIGRA). Changing Views: Worlds in Play.

Sánchez, J., Sáenz, M., & Ripoll, M. (2009). Usability of a Multimodal Videogame to Improve Navigation Skills for Blind Children. *ACM SIGACCESS Conference on Computers and Accessibility, Assets*.

Savidis, A., C. Stephanidis. (2004). Unified User Interface Design: Designing Universally Accessible Interactions. *International Journal of Interacting with Computers, 16*(2), 243-270.

Schrier, K. L. (2005). *Revolutionizing history education: Using augmented reality games to teach histories.* Massachusetts Institute of Technology, Cambridge, MA.

Squire, K. (2003). Video games in education. *International Journal of Intelligent Simulations and Gaming, 2*(1), 49-62.

Targett, S., M. Fernström. (2003). *Audio Games: Fun for All? All for Fun?* Paper presented at the International Conference on Auditory Display, Boston, MA, USA.

Torrente, J., del Blanco, Á., Marchiori, E. J., Moreno-Ger, P., & Fernández-Manjón, B. (2010). *<e-Adventure>: Introducing Educational Games in the Learning Process*. Paper presented at the IEEE EDUCON 2010.

Torrente, J., del Blanco, Á., Moreno-Ger, P., Martínez-Ortiz, I., & Fernández-Manjón, B. (2009). *Implementing Accessibility in Educational Videogames with <e-Adventure>*. Paper presented at the ACM International Workshop on Multimedia Technologies for Distance Learning (MTDL 2009), In Conjunction with the ACM International Conference on Multimedia (In press).

Torrente, J., Moreno-Ger, P., Fernández-Manjón, B., & Sierra, J. L. (2008). *Instructor-oriented Authoring Tools for Educational Videogames.* Paper presented at the 8th International Conference on Advanced Learning Technologies (ICALT 2008), Santander, Spain.

Valle Pérez, E. d. La web accesible para personas con discapacidad del Servicio de Biblioteca y Documentación. *Jornadas Andaluzas de Documentación*.

Van Eck, R. (2006). Digital game-based learning: It's not just the digital natives who are restless. EDUCAUSE Review, 41(2), 16-30.

Wang, Z., Li, B., Hedgpeth, T., & Haven, T. (2009). Instant Tactile-Audio Map: Enabling Access to Digital Maps for People with Visual Impairment. *ACM SIGACCESS Conference on Computers and Accessibility, Assets*.

APPENDICES

Appendix A: "Introduciendo Criterios de Accesibilidad en una Herramienta de Juegos Educativos: <e-Adventure>"

Complete Citation:

Elisa Sanz-Troyano, Javier Torrente, Pablo Moreno-Ger y Baltasar Fernández-Manjón (2010). Introduciendo Criterios de Accesibilidad en una Herramienta de Juegos Educativos: <e-Adventure>. CEDI 2010, XI Simposio Nacional de Tecnologías de la Información y las Comunicaciones en la Educación (SINTICE). Valencia, Spain.

Abstract:

In response to the needs of a changing society, in recent years has increased the interest in the application of new technologies in education. However, the introduction of new technology raises questions of accessibility, which can jeopardize the universal right to education for students with disabilities. Within this context, video games and game authoring tools have been proposed as a means to promote the development of new skills. This article describes the main challenges of accessibility and is described as being addressed within the authoring tool <e-Adventure> educational games. The ultimate goal is to convert <e-Adventure> a comprehensive solution for the development of accessible educational game considering the accessibility of both developed and developing games authoring tool itself.

About the CEDI 2010[43], Simposio Nacional de Tecnologías de la Información y las Comunicaciones en la Educación (SINTICE 2010):

This paper has been accepted for publication in the proceedings of the XI Simposio Nacional de Tecnologías de la Información y las Comunicaciones en la Educación (SINTICE 2010[44]). SINTICE 2010 aims to encourage the Computer Education, promote training of people in the new educational technologies, promote the exchange of research, ideas and experiences, and evaluate the educational quality of existing products in the scope of the CEDI 2010 in Valencia, Spain on the next September 7th 2010.

SINTICE 2010 will take place in conjunction with the annual Congreso Español De Informática (CEDI), which is a top national multi-conference which is intended as a forum for the professionals dedicated mainly to research, development, innovation and university education within the field of computer engineering.

[43] http://cedi2005.ugr.es/2010/contenido.php?apartado=organizacion&sub=presentacion

[44] http://cedi2005.ugr.es/2010/descripcion_sintice.html

Introduciendo Criterios de Accesibilidad en una Herramienta de Juegos Educativos: <e-Adventure>

Elisa Sanz-Troyano, Javier Torrente, Pablo Moreno-Ger, Baltasar Fernández-Manjón
Dpto. de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Univ. Complutense de Madrid
28040 - Madrid
estroyano@estumail.ucm.es, {jtorrente, pablom, balta}@fdi.ucm.es

Resumen

Como respuesta a las necesidades de una sociedad en constante evolución, durante los últimos años ha crecido el interés en la aplicación de las nuevas tecnologías en el ámbito educativo. No obstante, la introducción de nuevas tecnologías plantea problemas de accesibilidad, lo que puede poner en riesgo el derecho universal a la educación para alumnos con discapacidad. Dentro de este contexto, los videojuegos y las herramientas de autoría de videojuegos han sido propuestos como medio para fomentar el desarrollo de nuevas habilidades. En este artículo se plantean los principales retos de accesibilidad y se describe como se están abordando dentro de la herramienta de autoría de videojuegos educativos <e-Adventure>. El objetivo final es convertir <e-Adventure> en una solución integral para el desarrollo de videojuegos educativos accesibles considerando la accesibilidad tanto de los juegos desarrollados como de la propia herramienta de autoría.

1. Introducción

En los últimos años el *boom* de las tecnologías en todos los ámbitos de la vida diaria (e.g. relación social, ocio, trabajo) plantea como requisito imprescindible la accesibilidad de estas nuevas aplicaciones de modo que se evite la discriminación y la existencia de nuevas "brechas digitales" para los usuarios con algún tipo de discapacidad.

Hay campos especialmente relevantes como la educación donde se ha producido también esta incorporación de tecnologías integradas con nuevas estrategias pedagógicas y contenidos que preparen a los alumnos para los desafíos que una sociedad en constante evolución presenta (Schrier, 2005). Se están tratando de proporcionar contenidos más dinámicos e interactivos para los estudiantes y, entre ellos, se está abordando la integración de videojuegos en el flujo educativo dado su potencial como herramientas de apoyo del aprendizaje (P. Moreno-Ger et al., 2009; Squire, 2003). Los videojuegos se han usado por ejemplo, para mejorar el aprendizaje mediante la interacción del alumno con los videojuegos o como elemento de aprendizaje mediante la propia creación de videojuegos (Overmars, 2004).

A medida que los videojuegos han empezado a despertar interés en el ámbito académico como material educativo, han ido surgiendo propuestas para facilitar la autoría de los videojuegos y simplificar su aplicabilidad en el contexto educativo. Una de las principales propuestas son las herramientas de autoría ya que permiten reducir los elevados costes de producción de los videojuegos y facilitan su desarrollo por personas que no dispongan de amplios conocimientos técnicos (J. Torrente et al., 2008). De esta manera se simplifica que tanto profesores como alumnos puedan desarrollar sus propios videojuegos.

El problema de la accesibilidad, tanto de las herramientas de autoría para videojuegos como de los propios videojuegos no está resuelto. Y este problema es mayor cuando se plantea el uso de los juegos en el campo educativo en general y en el del e-Learning en particular. En e-learning la accesibilidad de los contenidos está más o menos cubierta gracias a las iniciativas que facilitan el acceso a las tecnologías Web (Magerko, 2008) y a las diversas tecnologías que facilitan el acceso a las principales herramientas de autoría de contenido educativo (por ejemplo, Microsft Word™, Power Point™, entornos de e-Learning como Moodle™, etc.). Pero es necesario proponer soluciones de accesibilidad integrales que consideren tanto el uso de videojuegos educativos como su creación mediante herramientas de autoría accesibles.

El propósito de este artículo es analizar los retos que plantea la introducción de accesibilidad en entornos de autoría y ejecución de videojuegos educativos (los expuestos anteriormente son algunos de los principales). A partir de este análisis presentamos cómo dichos problemas se están abordando en la herramienta <e-Adventure>, una plataforma para la creación de videojuegos educativos (Javier Torrente et al., 2009). El objetivo es proporcionar una herramienta de autoría para videojuegos que por un lado, sea accesible para personas con discapacidad, y por otro lado permita introducir características de accesibilidad en los juegos producidos mediante la configuración de módulos específicos sin que esto suponga un aumento significativo del coste de desarrollo.

Siguiendo esta línea, este artículo se estructura de la siguiente manera: En la sección 2 se estudia el trabajo relacionado centrándose en estudios de accesibilidad tanto en videojuegos como en herramientas de autoría. A continuación, en la sección 3, se discuten los requisitos generales de accesibilidad que se deberían tener en cuenta en cualquier herramienta de autoría de juegos. En la sección 4 se explica de una forma más detallada cómo se está adaptando <e-Adventure> para cumplir con dichos requisitos. Para finalizar, en la última sección (5) se discuten las ventajas e inconvenientes de nuestro enfoque así como las líneas de trabajo futuro.

2. Accesibilidad en videojuegos y herramientas software de escritorio

Mientras que el campo de la accesibilidad Web está bastante cubierto con iniciativas punteras como son WAI[45] o WCAG[46], para otro tipo de tecnologías, no basadas en la Web, no existen pautas ampliamente aceptadas sobre accesibilidad. Otra iniciativa también del W3C a destacar en nuestro ámbito, más reciente y por lo tanto no tan conocida como las anteriores, es la ATAG o Authoring Tool Accessibility Guidelines[47], que se centra en estudiar cómo hacer accesibles herramientas de autoría para diseñar contenido Web accesible. Aunque es una iniciativa enfocada también al campo Web, está enfocada a ser utilizada por los desarrolladores de herramientas de autoría para obtener interfaces de autoría accesibles. Una vez estudiadas las iniciativas generales, vamos a analizar las principales iniciativas de accesibilidad para dos tipos de sistemas que son de especial relevancia para este trabajo: las herramientas software de escritorio y los videojuegos.

2.1. Accesibilidad en aplicaciones software de escritorio

La accesibilidad en aplicaciones software de escritorio es un campo difícil de cubrir y que hasta ahora ha recibido atención dispar. La mayor parte de aplicaciones de escritorio que incluyen características de accesibilidad permiten la adaptación de la interfaz gráfica mediante la configuración de ciertos parámetros, como son el tamaño y color de la fuente, lo que permite resolver algunos de los problemas más frecuentes. Otra de las estrategias más comunes consiste en buscar la compatibilidad con herramientas externas de apoyo que suelen utilizar las personas con discapacidad (Hitchcock & Stahl, 2003). Esto incluye integración con lectores de pantalla como *Jaws*[48], o herramientas de reconocimiento de voz. Éste es el enfoque que se ha aplicado, por ejemplo, para aumentar la accesibilidad de herramientas software tales como el procesador de texto *Microsoft Word* (Hetzroni & Shrieber, 2004), o la hoja de cálculo *Microsoft Excel* (Doush et al., 2009).

No obstante, este enfoque resulta insuficiente para otro tipo de herramientas complejas como *Paint*, *Adobe PhotoShop* o *Adobe Flash*, en las que se presenta al usuario mucha información editable, y además se trabaja con un contenido altamente visual (tratamiento de imágenes, fotografías o mapas) y con modos de interacción poco accesibles como son *drag&drop* (arrastrar y soltar), (Wang et al., 2009). En este tipo de herramientas, incluir accesibilidad para que, por ejemplo, una persona con discapacidad visual pueda llegar a desenvolverse medianamente en el entorno, puede convertirse en una tarea ardua y costosa. En este sentido las herramientas que tienen un mayor éxito en materia de accesibilidad combinan el soporte para tecnologías de asistencia externas con la introducción de simplificaciones en la edición del contenido. Éste es el caso, por ejemplo, de *Microsoft Power Point*, que aunque es una herramienta de carácter muy visual, incluye una serie de plantillas que permiten una estructuración lineal y sencilla de los contenidos.

Dentro de esta familia de aplicaciones con mucha parte gráfica y visual, se pueden encuadrar también las herramientas de autoría de juegos. Estas herramientas trabajan con mucha información para configurar características tan variadas como personajes, objetos, conversaciones, escenas, etc. que conformarán el juego final. Estas herramientas presentan, por tanto, un reto significativo desde el punto de vista de la accesibilidad ya que la información de que se dispone es mucha y es difícil estructurarla correctamente, de tal manera que se cumpla con los requisitos de accesibilidad. Es necesario tratar de incluir nuevas técnicas que reflejen lo más posible el estado del entorno para tener una idea de los contenidos que se están tratando y cómo interactuar con ellos. No obstante, debido a la cantidad de información, su naturaleza y a su estructuración, es difícil, por ejemplo, navegar por la aplicación mediante teclado y orientar al usuario sobre el contexto para evitar que se pierda. Algunos de estos problemas se pueden abordar con técnicas complementarias como, por ejemplo, incluir un sintetizador de voz que vaya informando al usuario del su contexto actual.

2.2. Accesibilidad en videojuegos

Por lo general, la accesibilidad es uno de los asuntos no suficientemente abordados en la industria de los videojuegos. Ya sea por una falta de concienciación, o porque el retorno de la inversión en materia de accesibilidad no está claro para los estudios de desarrollo, el resultado es que los videojuegos son poco accesibles para personas con discapacidad. Además los videojuegos son un contenido multimedia interactivo, por lo que las adaptaciones necesarias para convertirlos en contenidos accesibles plantean un desafío (Magerko, 2008). No obstante, existen iniciativas puntuales de distinta índole tanto hardware como software que tratan de mejorar la accesibilidad de los videojuegos (K. Bierre et al., 2005).

Centrándonos en el hardware, uno de los enfoques más comunes es incorporar accesibilidad mediante la compatibilidad con dispositivos de soporte como son mandos de juego adaptados para personas con distintos tipos de discapacidad (Kearney, 2005), o incluso sistemas que reconozcan el movimiento de los ojos o de la cabeza[49]. También existen teclados especiales que proporcionan una manera alternativa de generar comandos de teclado mediante un sistema hardware e incluso software. Pueden ser teclados con una separación de teclas muy grande, teclados que se visualizan en la pantalla o incluso teclados activados por el movimiento del ojo.

Actualmente, la accesibilidad mediante dispositivos externos es el enfoque más común, dado que la inclusión de nuevas funcionalidades software requiere un coste añadido que la industria no suele estar dispuesta a asumir ya que este sobrecoste es recurrente en cada nuevo juego. Este problema se agrava en el

[45] http://www.w3.org/WAI/

[46] http://www.w3.org/TR/WCAG10/

[47] http://www.w3.org/TR/WAI-AUTOOLS/

48http://www.funcaragol.org/html/mainjaws.htm

[49] http://www.naturalpoint.com/#

campo de los juegos educativos, dónde los presupuestos son aún más limitados y la importancia de la accesibilidad es aún mayor. Una alternativa muy utilizada para solucionar este sobrecoste en los juegos, es utilizar herramientas (también software) de asistencia, como lectores de pantalla, magnificadores, sintetizadores / reconocedores de voz (en el caso de presentar discapacidad motora) o teclados virtuales.

Otra alternativa estudiada, desde el punto de vista de inclusión de recursos, se basa en diseñar la interacción de forma multimodal (por ejemplo, combinando subtítulos e imágenes con sonidos y voces (Röber, 2005), interactuando con el juego mediante comandos de voz (Targett, 2003), etc.). No obstante, este tipo de soluciones suelen desarrollarse *ad-hoc*, y por ello conllevan asimismo un aumento considerable del coste (por ejemplo, sonorización de los diálogos entre personajes), lo que vuelve a chocar con los problemas de limitaciones de presupuesto.

3. La plataforma <e-Adventure>

<e-Adventure> es una herramienta para la creación de videojuegos y simulaciones educativas desarrollada por el grupo <e-UCM> de la Universidad Complutense de Madrid[50]. El objetivo de <e-Adventure> es facilitar el desarrollo de este tipo de contenidos a personas sin un perfil técnico (esto es, fundamentalmente docentes o alumnos).

Como estrategia para reducir los costes de desarrollo así como la complejidad de la herramienta de autoría, <e-Adventure> se centra en el género de las aventuras gráficas y juegos *point-and-click* (apuntar y señalar) en dos dimensiones, estilo Monkey Island™ o Myst™ (Moreno-Ger et al., 2008; Moreno-Ger et al., 2007a).

Además <e-Adventure> es multiplataforma (desarrollado en Java), permitiendo la exportación de los juegos para ser ejecutados en Windows, Linux, Mac o incluso un navegador de Internet.

3.1. Análisis de accesibilidad del editor

<e-Adventure> ejemplifica muchas de las características que suelen dificultar especialmente la introducción de accesibilidad en este tipo de software. Entre los problemas más frecuentes podemos destacar la configuración de los elementos visuales que componen el mundo virtual del juego, así como la estructuración de la información editable; esto es, las numerosas opciones de configuración de la herramienta.

En primer lugar, por lo general los videojuegos suelen constar de complejos mundos virtuales que se componen de escenarios poblados de personajes, objetos, animaciones y efectos visuales.

Esto plantea un serio problema de accesibilidad, sobre todo para personas con discapacidad visual ya que el autor del juego debe ser capaz de recrear el mundo virtual y posicionar los distintos elementos que lo componen en conjunto y no de forma aislada, sin el apoyo del sentido de la vista.

Una de las formas de facilitar el acceso a la configuración del universo del juego consiste en simplificarlo a un mundo en 2 dimensiones, en lugar de las 3 dimensiones en las que se suelen producir los videojuegos actualmente. No obstante, sigue siendo necesario en estos casos solucionar el problema de colocar los elementos en los escenarios del juego. Este problema se produce de forma recurrente en el editor de <e-Adventure>. Además, este tipo de herramientas suele basarse en mecanismos de interacción que son poco accesibles, como por ejemplo *drag&drop* (arrastrar y soltar).

En segundo lugar, este tipo de herramientas proporcionan un gran número de opciones de configuración que son difícilmente simplificables sin que esto suponga reducir excesivamente su capacidad expresiva. Esto obliga a una estructuración de la información mucho más compleja que en otro tipo de aplicaciones software, recurriendo a estrategias de diseño que dificultan la accesibilidad a la hora de navegar mediante teclado e impiden la comprensión rápida de la información. Esta estructuración de la información se suele llevar a cabo mediante menús anidados, tablas o diálogos emergentes.

<e-Adventure> ejemplifica ambos problemas. El editor de <e-Adventure> (ver Figura 1) dispone de un panel lateral para acceder a los principales elementos que configuran un juego. Esto es, las escenas que componen el mundo virtual, los objetos y personajes que pueden encontrarse en ellos, las escenas de corte que suelen utilizarse para proporcionar fuentes extra de información o los diálogos entre personajes. En la parte superior de la ventana se encuentra la barra de herramientas con las diferentes opciones y en la parte central el panel de edición de las propiedades de cada elemento. Este panel a su vez suele componerse de pestañas, tablas y otros sub-paneles anidados en los que se agrupan las características editables de cada elemento. Esta forma de estructurar la información dificulta la navegación por teclado entre los distintos paneles y opciones de configuración, que debe realizarse secuencialmente recorriendo los distintos elementos de la interfaz de usuario mediante teclas como el tabulador o las flechas de dirección.

3.2. Análisis de accesibilidad de los juegos

Otro problema especialmente relevante concierne el nivel de accesibilidad de los juegos producidos con la herramienta de autoría. Este es un problema que ha sido abordado en el editor <e-Adventure> (Javier Torrente et al., 2009) mediante la activación de ciertas características como, por ejemplo, síntesis de voz que proporcionará feedback auditivo, el cual supondrá que se activen los recursos necesarios para ese juego en particular.

[50] http://e-adventure.e-ucm.es/

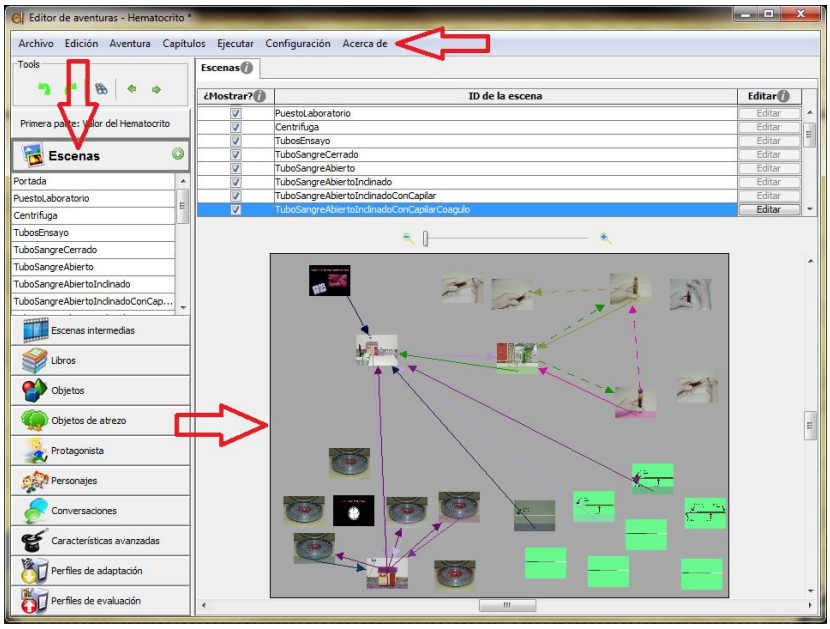


Figura 1. Captura de la ventana principal del editor de <e-Adventure>

Para mejorar la accesibilidad de los juegos es necesario incorporar ciertas modificaciones que harán que éste sea diferente para cada usuario. Sin embargo, en la mayoría de los casos, las adaptaciones se centran en los canales de interacción de usuario del juego, es decir, la entrada y salida de los sistemas del juego. Puesto que un juego es sobre todo una experiencia interactiva, estas adaptaciones pueden suponer un reto significativo desde el punto de vista técnico, ya que requieren el uso de tecnologías sofisticadas como las descritas en la sección 2.2.

Otros problemas de accesibilidad frecuentes pueden resolverse mediante la flexibilización de algunos parámetros de configuración del juego. Ciertas características visuales, auditivas y de movilidad pueden ser adaptadas realizando cambios mínimos. Por ejemplo, las personas con visibilidad reducida, necesitarán un texto con un tamaño mayor que una persona con visibilidad total. Otros usuarios con movilidad reducida o algunas discapacidades cognitivas pueden requerir la relajación de los tiempos de respuesta en la interacción con el juego (por ejemplo, aumentar el tiempo necesario para completar un puzzle o para responder a una pregunta. En cambio, puede ser necesario realizar cambios y adaptaciones más profundos si el usuario presenta una discapacidad cognitiva, en este caso será necesario incluso adaptar la traza del juego (Javier Torrente et al., 2009).

4. Introducción de Accesibilidad en <e-Adventure>

La accesibilidad es un campo complejo en el que las necesidades de los usuarios requieren adaptaciones de la interfaz muy variadas según su discapacidad. Por ello, en una primera iteración nos hemos centrado en las discapacidades visuales como objetivo, aunque muchas de las soluciones propuestas benefician también a otros usuarios. En esta sección describimos cuáles han sido las principales líneas de actuación.

4.1. Configuración de la interfaz gráfica de la herramienta de autoría

En una primera fase de desarrollo hemos flexibilizado la configuración de la interfaz de usuario. Según lo discutido en la sección 2.1 esta es una de las estrategias más comunes en materia de accesibilidad ya que permite resolver algunos de los problemas más comunes. No obstante, existen varios enfoques a la hora de proporcionar soluciones para la configuración de la interfaz de usuario.

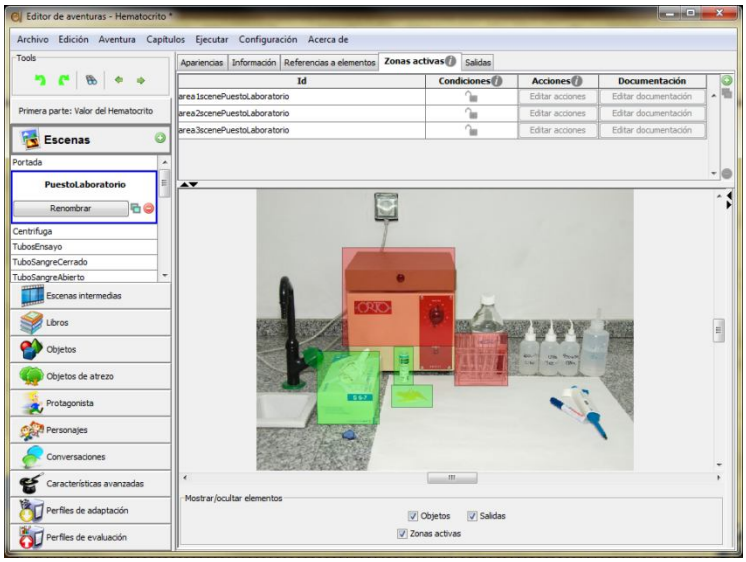


Figura 2. Selección de una Escena, que tiene pestañas y que a su vez la pestaña contiene una tabla y un sub-panel.

Una de las opciones consiste en dar al usuario la posibilidad de elegir los parámetros de configuración manualmente; por otro lado existe la alternativa de cargar automáticamente las opciones de configuración según el perfil del usuario, que puede cargarse en algunos casos desde el Sistema Operativo. Debido a la naturaleza multiplataforma de <e-Adventure>, inicialmente hemos optado por la primera alternativa, añadiendo un menú desplegable bajo el nombre *"Accesibilidad"* en el menú principal, en el que el usuario puede seleccionar el tamaño de fuente (normal o grande) Y el modo de visualización (normal o alto contraste).

Para incorporar accesibilidad en los paneles donde se configura el mundo del juego, en los que la interacción se realiza mediante interacción *drag&drop* fundamentalmente, hemos incorporado un sistema que permite editar la posición de los elementos en las escenas con las flechas de dirección.

4.2. Estructuración y navegación por la información

Tal y como se ha descrito en el apartado 3.1, uno de los retos más complejos a la hora de introducir accesibilidad en <e-Adventure> es la estructuración de la gran cantidad de información que presenta y el mecanismo de navegación que permita acceder a todos los componentes mediante el teclado. Esto lo hemos resuelto mediante la definición de una estructura arbórea de "ciclos de foco".

Un ciclo de foco establece el orden en el que se pueden inspeccionar mediante teclado los elementos de una parte concreta de la interfaz. Por defecto hemos adoptado la convención de utilizar el tabulador para circular entre los elementos dentro del ciclo de foco (nivel del árbol) activo. Cuando uno de los elementos presentes en la interfaz se compone a su vez de una agrupación mayor de componentes, el usuario puede "saltar" en el árbol y comenzar a explorar los componentes de dicho elemento utilizando la combinación ctrl.+tabulador, cambiando el ciclo de foco activo de esta manera. Para volver al nivel de foco superior se utiliza la combinación ctrl.+shift+tabulador.

No obstante, dado que la navegación a través del árbol completo de foco es compleja también hemos incluido atajos de teclado para acceder directamente a ciertos puntos del "árbol de foco", así como para acceder a opciones frecuentes de la barra de menús. Por convención se ha utilizado ALT+shift+código o CRTL+código (A, E, F). Al igual que se ha habilitado acceso a las diferentes pestañas de la barra de menús, mediante el uso de ALT+letra.

Por otro lado, al estar presentes varios niveles de ciclo de foco, algunas acciones pueden cambiar automáticamente el ciclo de foco. Por ejemplo, al estar recorriendo el esquema lateral de elementos, si utilizamos la tecla de retorno o la barra espaciadora para seleccionar el elemento, el foco cambia automáticamente al ciclo de foco que corresponde al panel de edición del elemento.

4.3. Retorno multimodal de la información

Uno de los principales problemas que plantea la accesibilidad para personas con discapacidad visual es como proveer un retorno alternativo (multimodal) de la información. Por ejemplo, una estructura de navegación tan compleja como la presentada en 4.2 dificulta recordar qué parte de la información se está editando en ese momento (esto es, cuál es el ciclo de foco activo). Como solución se ha añadido un comando especial que indica mediante síntesis de voz al usuario el camino que se ha seguido sobre el árbol.

De esta manera el retorno de la información al usuario se realiza tanto de forma visual como auditiva. En este sentido, además de la capacidad de síntesis de voz se han incluido sonidos especiales asociados a eventos concretos, como por ejemplo:

- Imagen bien/mal cargada
- Imagen bien enlazada: Cuando se configura la sucesión de escenas y es necesario enlazar la actual con la que le sucederá.
- Elemento referenciado en la imagen: En el caso de un personaje o un objeto con el que se va a interactuar.

Centrándonos más en aspectos técnicos, la síntesis de voz se ha implementado mediante el uso de la librería de síntesis de voz de Microsoft (Speech API 5.0) a la que se invoca a través de la librería Com4J, que permite realizar llamadas a interfaces COM de Microsoft desde código Java. A través de este sistema el usuario puede configurar los aspectos de síntesis de voz mediante herramientas de asistencia externas, como por ejemplo las funcionalidades que incluye Windows Vista (o superior).

4.4. Configuración de la accesibilidad en el juego

Uno de los aspectos más relevantes en el desarrollo de la accesibilidad en <e-Adventure> es la configuración de los propios videojuegos a través de la herramienta de autoría (Javier Torrente et al., 2009). Una vez que el juego está diseñado, el autor del juego puede activar o desactivar una serie de módulos de entrada / salida y otras herramientas de apoyo (por ejemplo una lupa integrada en la metáfora del juego) que vienen ya implementados por defecto.

En general, para hacer accesibles estos videojuegos el autor dispone de las siguientes herramientas (ver Figura 3): *Capa de Interfaz del Ratón* que proporciona la interacción clásica *point-and-click* mediante el uso del ratón, la *Capa de Interfaz de la Voz* que permite la interacción mediante reconocimiento de voz y la Capa de Interfaz por teclado.

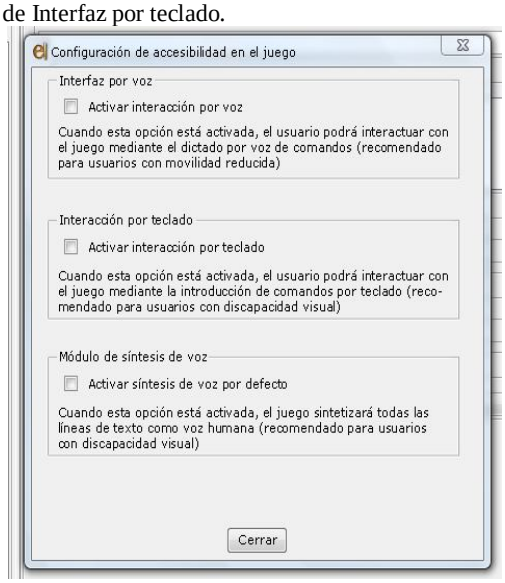


Figura 3. Configuración de las características de accesibilidad de un juego desde el Editor

5. Discusión y Trabajo Futuro

En este artículo hemos analizado la necesidad de realizar la introducción de nuevas tecnologías en el campo educativo garantizando la inclusión de todos, tanto alumnos como profesores. En este sentido, muchos son los retos que deben resolverse todavía para introducir tecnologías altamente visuales e interactivas como son los videojuegos y herramientas de autoría de juegos en el sistema educativo de una forma incluyente. Es por ello que estamos desarrollando la accesibilidad en la herramienta <e-Adventure>, tal y como hemos presentado en este artículo.

La introducción de accesibilidad en una herramienta de autoría como <e-Adventure> plantea dos retos muy diferenciados: por un lado, lograr que una herramienta *drag&drop* y de contenido altamente visual sea accesible; por el otro, incluir funcionalidades fácilmente configurables en la plataforma que permitan introducir accesibilidad en los juegos producidos con un coste extra marginal.

Pese al trabajo realizado, las metodologías o soluciones propuestas presentan una serie de limitaciones. Por un lado, la introducción de accesibilidad en <e-Adventure> se ha centrado en un subconjunto reducido de discapacidades principalmente visuales. En próximas iteraciones el trabajo deberá prestar especial atención a discapacidades auditivas, motoras y cognitivas.

Por otro lado, muchos de los problemas y soluciones propuestos para la herramienta <e-Adventure> son extrapolables a otro tipo de aplicaciones de escritorio con características similares (interacción *drag&drop*, alto contenido visual, etc.). De esta manera, una de nuestras líneas de trabajo futuro consiste en la generalización de las funcionalidades que se están desarrollando para <e-Adventure> a otras herramientas y paradigmas, favoreciendo de esta manera el desarrollo de la accesibilidad en un sector más amplio.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Comité Español de Ciencia y Tecnología (TIN2007-68125-C02-01) y el Ministerio de Industria, Turismo y Comercio (TSI-020301-2009-9, TSI-020110-2009-170, TSI-020312-2009-27), así como por la Universidad Complutense de Madrid y el Gobierno de la Comunidad de Madrid (grupo de investigación consolidado 921340 y red de excelencia *e-Madrid* S2009/TIC-1650), y por la Unión Europea (505469-2009-LLP-ES-KA3-KA3MP).

Referencias

[1] Argue, R., Boardman, M., Doyle, J., & Hickey, G. (2004). Building a Low-Cost Device to Track Eye Movement.

[2] Bierre, K., Chetwynd, J., Ellis, B., Hinn, D. M., Ludi, S., & Westin, T. (2005). *Game Not Over: Accessibility Issues in Video Games.* Paper presented at the HCII.

[3] Bierre, K., Hinn, M., Martin, T., McIntosh, M., & Snider, T. (2004). *Accessibility in Games: Motivations and Approaches* International Game Developers Association (IGDA).

[4] Csikszentmihalyi, M. (1991). *Flow: The Psychology of Optimal Experience*: {Harper Perennial}.

[5] Dickey, M. D. (2006). Game Design Narrative for Learning: Appropriating Adventure Game Design Narrative Devices and Techniques for the Design of Interactive Learning Environments. *Educational Technology Research and Development, 54*(3), 245-263.

[6] Doush, I. A., Pontelli, E., Simon, D., Son, T. C., & Ma, O. (2009). Making Microsoft Excel™ Accessible: Multimodal Presentation of

Charts. *ACM SIGACCESS Conference on Computers and Accessibility, Assets*.

[7] Fernández-Manjón, B., Sierra, J. L., Martínez-Ortiz, I., & Moreno-Ger, P. (2009). Estandarización y Modelado Educativo. Informe nº 20. *Instituto Superior de Formación y Recursos en Red para el Profesorado (ISFTIC) (In Press)*.

[8] Foss, B. A., & Eikaas, T. I. (2006). Game Play in Engineering Education— Concept and Experimental Results. *International Journal of Engineering Education, 22*(5), 1043-1052(1010).

[9] Grammenos, D., A. Savidis, & C. Stephanidis. (2007). Unified Design of Universally Accessible Games. In S. B. Heidelberg (Ed.), *Universal Access in Human-Computer Interaction. Applications and Services* (Vol. 4556/2007, pp. 607-616).

[10] Hetzroni, O. E., & Shrieber, B. (2004). Word Processing as an Assistive Technology Tool for Enhancing Academic Outcomes of Students with Writing Disabilities in the General Classroom. *Journal of Learning Disabilities, 37*(2), 143-154.

[11] Hitchcock, C., & Stahl, S. (2003). Assistive technology, universal design, universal design for learning: Improved learning opportunities. *Journal of Special Edcuaiton Technology, 18*(4), 45-52.

[12] Houlette, R. (2004). Player modeling for adaptive games. In *AI Game Programming Wisdom* (Vol. II, pp. 557–566): Charles River Media.

[13] Jiménez Peñuela, J. R. (2005). Desarrollo de Sitios Web accesibles a la Población Sorda.

[14] Kearney, P. R. (2005). *Playing in the Sandbox: Developing games for children with disabilities*. Paper presented at the DiGRA 2005 Conference: Changing Views - Worlds in Play. Retrieved 30th May, from http://www.digra.org/dl/display_html?chid=06278.12473.pdf.

[15] Magerko, B. (2008). Adaptation in Digital Games. *Computer, 41*(6), 87-89.

[16] Moreno-Ger, Burgos, Sierra, & Fernández-Manjón. (2008). Educational Game Design for Online Education. *Computers in Human Behavior, 24*(6), 2530-2540.

[17] Moreno-Ger, Sierra, Martínez-Ortiz, & Fernández-Manjón. (2007a). A Documental Approach to Adventure Game Development. *Science of Computer Programming, 67*(1), 3–31.

[18] Moreno-Ger, Thomas, S., Martínez-Ortiz, Sierra, & Fernández-Manjón. (2007b). Adaptive Units of Learning and Educational Videogames. *Journal of Interactive Media in Education, 2007*(05).

[19] Moreno-Ger, P., Burgos, D., & Torrente, J. (2009). Digital games in e-Learning environments: current uses and emerging trends. *Simulation & Gaming, 40*(5), 669-687.

[20] Overmars, M. (2004). Teaching Computer Science through Game Design. *IEEE Computer, 37*(4), 81-83.

[21] Röber, N., M. Masuch. (2005). *Playing Audio-Only Games: A Compendium of Interacting with Virtual, Auditory Worlds*. Paper presented at the Digital Games Research Association Conference (DIGRA). Changing Views: Worlds in Play.

[22] Sánchez, J., Sáenz, M., & Ripoll, M. (2009). Usability of a Multimodal Videogame to Improve Navigation Skills for Blind Children. *ACM SIGACCESS Conference on Computers and Accessibility, Assets*.

[23] Savidis, A., C. Stephanidis. (2004). Unified User Interface Design: Designing Universally Accessible Interactions. *International Journal of Interacting with Computers, 16*(2), 243-270.

[24] Schrier, K. L. (2005). *Revolutionizing history education: Using augmented reality games to teach histories.* Massachusetts Institute of Technology, Cambridge, MA.

[25] Squire, K. (2003). Video games in education. *International Journal of Intelligent Simulations and Gaming, 2*(1), 49-62.

[26] Targett, S., M. Fernström. (2003). *Audio Games: Fun for All? All for Fun?* Paper presented at the International Conference on Auditory Display, Boston, MA, USA.

[27] Torrente, J., del Blanco, Á., Marchiori, E. J., Moreno-Ger, P., & Fernández-Manjón, B. (2010). *<e-Adventure>: Introducing Educational Games in the Learning Process*. Paper presented at the IEEE EDUCON 2010.

[28] Torrente, J., del Blanco, Á., Moreno-Ger, P., Martínez-Ortiz, I., & Fernández-Manjón, B. (2009). *Implementing Accessibility in Educational Videogames with <e-Adventure>*. Paper presented at the ACM International Workshop on Multimedia Technologies for Distance Learning (MTDL 2009), In Conjunction with the ACM International Conference on Multimedia (In press).

[29] Torrente, J., Moreno-Ger, P., Fernández-Manjón, B., & Sierra, J. L. (2008). *Instructor-oriented Authoring Tools for Educational Videogames.* Paper presented at the 8th International Conference on Advanced Learning Technologies (ICALT 2008), Santander, Spain.

[30] Valle Pérez, E. d. La web accesible para personas con discapacidad del Servicio de Biblioteca y Documentación. *Jornadas Andaluzas de Documentación*.

[31] Van Eck, R. (2006). Digital game-based learning: It's not just the digital natives who are restless. *EDUCAUSE Review, 41*(2), 16-30.

[32] Wang, Z., Li, B., Hedgpeth, T., & Haven, T. (2009). Instant Tactile-Audio Map: Enabling Access to Digital Maps for People with Visual Impairment. *ACM SIGACCESS Conference on Computers and Accessibility, Assets.*

[33]

[34]