CENTRO DE CIENCIAS BÁSICAS

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

TESIS

Design of a Process for the Development of Serious Games for Auditory Rehabilitation

PRESENTA

David Augusto Céspedes Hernández

PARA OBTENER EL GRADO DE MAESTRO EN CIENCIAS CON OPCIÓN A LA COMPUTACIÓN, MATEMÁTICAS APLICADAS

TUTORES

Dr. Francisco Javier Álvarez Rodríguez

Dr. Juan Manuel González Calleros

COMITÉ TUTORAL

Dr. Jaime Muñoz Arteaga

Dr. Jean Vanderdonckt

Dr. Jorge Octavio Ocharán Hernández

Aguascalientes, Ags, 10 de Diciembre del 2015

Autorizaciones

UNIVERSIDAD AUTONOMA
DE AGUASCALIENTES

DAVID AUGUSTO CESPEDES HERNANDEZ
MAESTRIA EN CIENCIAS CON OPCION
A LA COMPUTACIÓN Y MATEMATICAS APLICADAS
P R E S E N T E.

Estimado alumno:

Por medio de este conducto me permito comunicar a Usted que habiendo recibido los votos aprobatorios de los revisores de su trabajo de tesis y/o caso práctico titulado: **"Design of a process for the development of serious games to support auditory rehabilitation",** hago de su conocimiento que puede imprimir dicho documento y continuar con los trámites para la presentación de su examen de grado.

Sin otro particular me permito saludarle muy afectuosamente.

A T E N T A M E N T E
Aguascalientes, Ags., a 27 de enero de 2016
"Se lumen proferre"
EL DECANO

M. en C. JOSE DE JESUS RUIZ GALLEGOS

c.c.p.- Archivo.
JJRG,yscd

UNIVERSIDAD AUTONOMA
DE AGUASCALIENTES

FORMATO DE CARTA DE VOTO APROBATORIO

M. en C. JOSÉ DE JESÚS RUIZ GALLEGOS
DECANO (A) DEL CENTRO DE CIENCIAS BÁSICAS
P R E S E N T E

Por medio del presente como Asesor designado del estudiante **DAVID AUGUSTO CÉSPEDES HERNÁNDEZ** con ID **184886** quien realizó la tesis titulada: **DESIGN OF A PROCESS FOR THE DEVELOPMENT OF SERIOUS GAMES TO SUPPORT AUDITORY REHABILITATION**, y con fundamento en el Artículo 175, Apartado II del Reglamento General de Docencia, me permito emitir el **VOTO APROBATORIO**, para que el pueda proceder a imprimirla, y así como continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

A T E N T A M E N T E
"Se Lumen Proferre"
Aguascalientes, Ags., a 10 de Diciembre de 2015.

Dr. Francisco J. Álvarez Rodríguez
Director de tesis

c.c.p.- Interesado
c.c.p.- Secretaría de Investigación y Posgrado
c.c.p.- Jefatura del Depto. de Ciencias de la Computación
c.c.p.- Consejero Académico
c.c.p.- Minuta Secretario Técnico

UNIVERSIDAD AUTONOMA
DE AGUASCALIENTES

FORMATO DE CARTA DE VOTO APROBATORIO

M. en C. JOSÉ DE JESÚS RUIZ GALLEGOS
DECANO (A) DEL CENTRO DE CIENCIAS BÁSICAS
P R E S E N T E

Por medio del presente como Co-Director designado del estudiante **DAVID AUGUSTO CÉSPEDES HERNÁNDEZ** con ID **184886** quien realizó la tesis titulada: **DESIGN OF A PROCESS FOR THE DEVELOPMENT OF SERIOUS GAMES TO SUPPORT AUDITORY REHABILITATION**, y con fundamento en el Artículo 175, Apartado II del Reglamento General de Docencia, me permito emitir el **VOTO APROBATORIO**, para que el pueda proceder a imprimirla, y así como continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

A T E N T A M E N T E
"Se Lumen Proferre"
Aguascalientes, Ags., a 19 de Enero de 2016.

Dr. Juan Manuel González Calleros
Co-Director de tesis

c.c.p.- Interesado
c.c.p.- Secretaría de Investigación y Posgrado
c.c.p.- Jefatura del Depto. de Ciencias de la Computación
c.c.p.- Consejero Académico
c.c.p.- Minuta Secretario Técnico

UNIVERSIDAD AUTONOMA
DE AGUASCALIENTES

FORMATO DE CARTA DE VOTO APROBATORIO

M. en C. JOSÉ DE JESÚS RUIZ GALLEGOS
DECANO (A) DEL CENTRO DE CIENCIAS BÁSICAS
P R E S E N T E

Por medio del presente como Asesor designado del estudiante **DAVID AUGUSTO CÉSPEDES HERNÁNDEZ** con ID **184886** quien realizó la tesis titulada: **DESIGN OF A PROCESS FOR THE DEVELOPMENT OF SERIOUS GAMES TO SUPPORT AUDITORY REHABILITATION**, y con fundamento en el Artículo 175, Apartado II del Reglamento General de Docencia, me permito emitir el **VOTO APROBATORIO**, para que el pueda proceder a imprimirla, y así como continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

A T E N T A M E N T E
"Se Lumen Proferre"
Aguascalientes, Ags., a 18 de Enero de 2016.

Dr. Jaime Muñoz Arteaga
Asesor de tesis

c.c.p.- Interesado
c.c.p.- Secretaría de Investigación y Posgrado
c.c.p.- Jefatura del Depto. de Ciencias de la Computación
c.c.p.- Consejero Académico
c.c.p.- Minuta Secretario Técnico

UNIVERSIDAD AUTONOMA
DE AGUASCALIENTES

FORMATO DE CARTA DE VOTO APROBATORIO

M. en C. JOSÉ DE JESÚS RUIZ GALLEGOS
DECANO (A) DEL CENTRO DE CIENCIAS BÁSICAS
P R E S E N T E

Por medio del presente como Asesor designado del estudiante **DAVID AUGUSTO CÉSPEDES HERNÁNDEZ** con ID **184886** quien realizó la tesis titulada: **DESIGN OF A PROCESS FOR THE DEVELOPMENT OF SERIOUS GAMES TO SUPPORT AUDITORY REHABILITATION**, y con fundamento en el Artículo 175, Apartado II del Reglamento General de Docencia, me permito emitir el **VOTO APROBATORIO**, para que el pueda proceder a imprimirla, y así como continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

A T E N T A M E N T E
"Se Lumen Proferre"
Aguascalientes, Ags., a 19 de Enero de 2016.

Prof. Dr. Jean Vanderdonckt (UCL)
Asesor de tesis

c.c.p.- Interesado
c.c.p.- Secretaría de Investigación y Posgrado
c.c.p.- Jefatura del Depto. de Ciencias de la Computación
c.c.p.- Consejero Académico
c.c.p.- Minuta Secretario Técnico

UNIVERSIDAD AUTONOMA
DE AGUASCALIENTES
FORMATO DE CARTA DE VOTO APROBATORIO

M. en C. JOSÉ DE JESÚS RUIZ GALLEGOS
DECANO (A) DEL CENTRO DE CIENCIAS BÁSICAS
P R E S E N T E

Por medio del presente como Asesor designado del estudiante **DAVID AUGUSTO CÉSPEDES HERNÁNDEZ** con ID **184886** quien realizó la tesis titulada: **DESIGN OF A PROCESS FOR THE DEVELOPMENT OF SERIOUS GAMES TO SUPPORT AUDITORY REHABILITATION**, y con fundamento en el Artículo 175, Apartado II del Reglamento General de Docencia, me permito emitir el **VOTO APROBATORIO**, para que el pueda proceder a imprimirla, y así como continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

A T E N T A M E N T E
"Se Lumen Proferre"
Aguascalientes, Ags., a 19 de Enero de 2016.

MIS. Jorge Octavio Ocharán Hernández
Asesor de tesis

c.c.p.- Interesado
c.c.p.- Secretaría de Investigación y Posgrado
c.c.p.- Jefatura del Depto. de Ciencias de la Computación
c.c.p.- Consejero Académico
c.c.p.- Minuta Secretario Técnico

Agradecimientos

Agradezco el apoyo del Consejo Nacional de Ciencia y Tecnología (CONACYT) a través del programa de becas nacionales con número 376408, para la realización de los estudios de posgrado de los que la presente tesis resultó. Así mismo, al programa de beca mixta del mismo organismo por la aportación de recursos para la realización de una estancia de investigación en el extranjero.

Agradezco igualmente la oportunidad brindada por la Universidad Autónoma de Aguascalientes, para la realización de los estudios de posgrado, así como por los recursos económicos brindados para la participación en congresos de impacto nacional e internacional y para la realización de una estancia de investigación en el extranjero yal cuerpo de profesores del posgrado por sus valiosas aportaciones a mi iniciación en la investigación y por los conocimientos que me aportaron a través de las diferentes materias que forman parte del plan de estudios de la maestría.. En específico, reconozco las facilidades otorgadas por el Centro de Ciencias Básicas de esta institución y por la Unidad de Atención Integral a la Salud.

En este apartado, agradezco también las facilidades otorgadas por la Université catholique de Louvain por recibirme en su escuela de administración, para realizar en ella una estancia de investigación.

Gracias al Dr. Francisco Javier Álvarez Rodríguez, quien fungió como director de este trabajo, por sus asesorías y gestiones sin las cuales esta tesis no hubiera podido ser concluida de manera exitosa.

Al Dr. Juan Manuel González Calleros, en su postura de co-director de este trabajo, agradezco sus constantes asesorías presenciales y remotas, así como sus aportaciones y observaciones que permitieron formar este trabajo desde sus bases.

De la misma manera quiero agradecer la dedicación y el tiempo invertido para la revisión de este trabajo a mis asesores Dr. Jaime Muñoz Arteaga, Prof. Jean Vanderdonckt y MIS. Jorge Octavio Ocharán Hernández. Sus comentarios y observaciones enriquecieron este trabajo y brindaron aspectos relevantes que resultaron de vital importancia para lograr los objetivos planteados.

Dedicatorias

A mis padres, Patricia Hernández y David Céspedes por su gran apoyo, porque a pesar de estar lejos, se mantuvieron siempre al pendiente de mí. Por esas largas llamadas en las que escuchaban cómo había sido mi semana, los enteraba de mis planes y ellos me compartían también sus vidas. Gracias por los dulces y la comida enviados, las fotos de Camilo, las felicitaciones en la madrugada y los abrazos de despedida en la central de camiones. Agradezco también a mi hermana Stephanny, por los chistes, fotos, gifs, y memes compartidos, porque daban pie a empezar pláticas y a compartir momentos de la misma manera o aún más que cuando estábamos juntos.

Gracias a mis profesores de la maestría por compartir conmigo sus conocimientos y por ayudarme a crecer como investigador, estudiante y persona, y en especial a aquellos que formaron parte de mi comité de tesis. Gracias también al personal administrativo de la universidad por orientarme y facilitarme múltiples procesos.

A los compañeros y amigos que estuvieron a mi lado durante estos dos años, y que me apoyaron en distintas situaciones, entre otras, al ingresar a la maestría, al mudarnos a un estado que no conocíamos, al visitar países que no conocíamos o al encaminar mi trabajo y compartir conmigo sus experiencias.

Quiero dedicar este trabajo a dos personas, la primera es alguien que ha estado junto a mí durante ya casi 6 años, compartiendo tantas cosas materiales e inmateriales, alegrías, tristezas, viajes, casas, fotografías, colecciones, enojos, cuidados, miedos, sorpresas, y compromisos y que me hace pensar que aún quedan demasiadas cosas más por venir. Gracias Liliana Rodríguez Vizzuett por ser mi compañera en todo esto y por soportar mis locuras. Como lo prometí en la tesis anterior, aquí está tu nombre en otra dedicatoria, y espero poder ponerlo en otra más. Ojalá podamos seguir viviendo aventuras juntos y creciendo uno al lado del otro. T.A.C.N.Y.P.S.

La segunda persona a quien quiero dedicar este trabajo es a mi Abue Anita Rivera. Gracias por todas tus enseñanzas, por darme tu vida durante tantos años y por compartir conmigo tus historias y canciones. Gracias también, porque aunque no sabíamos que tu partida estaba por llegar, me permitiste despedirme de ti. Hoy sé que seguirás cuidándome desde donde estés y quiero que sepas que siempre te recordaré con mucho cariño.

Table of Content

Table of Tables

Table of Figures

Resumen

Las discapacidades auditivas son mundialmente reconocidas como una de las más comunes y que afecta a una gran cantidad de personas. En la literatura se reportan múltiples trabajos enfocados en el diseño y desarrollo de sistemas para la rehabilitación de usuarios con discapacidades, así como para la definición de modelos que permitan la expresión de sus capacidades y discapacidades. Desafortunadamente, cuando se trata de desarrollar sistemas interactivos para asistir en programas de terapia auditiva, el conocimiento de análisis y diseño, que es propio de la aplicación de metodologías está ausente. En este trabajo de tesis, se efectúa el enriquecimiento de un modelo de proceso de desarrollo, con elementos de rehabilitación y diseño de juegos, así como de herramientas relacionadas a la Interacción Humano-Computadora, con el fin de facilitar la implementación de juegos serios para soporte a la rehabilitación auditiva en tiempos cortos y con recursos humanos limitados. Para llevar a cabo este enriquecimiento, se describen actividades, roles y herramientas para cada una de las etapas involucradas en el proceso propuesto. Finalmente, para realizar la verificación del proceso y la validación de juegos serios desarrollados al seguirlo, se condujo un experimento en el que se contrastaron proyectos en los que se seguía con otros en los que no.

Abstract

Hearing impairments are widely recognized as one of the most common disability that affects many people. There are several works addressing the design and development of systems for the rehabilitation of users with disabilities, as well as for the definition of models to express their capabilities and disabilities. Unfortunately, when we want to develop interactive systems to assist hearing impairment therapy, knowledge of analysis and design, typical of the methodologies is absent. In this thesis work, the enrichment of a development process model, with rehabilitation, and game design elements along with Human-Computer Interaction related tools, for promoting the implementation of serious games to support auditory rehabilitation in short time and with limited human resources is performed. To accomplish this enrichment, activities, roles, and tools are described for each of the stages that are involved in the process definition. Finally, in order to carry out verification of the process and validation of serious games produced under its guidelines, an experiment was performed in which the development is contrasted while using them or not.

1. Introduction

In several contexts such as healthcare, military and educational, games have been applied as a way to improve skills and for training professionals [Arna13] [McDo06]. This type of games is called serious games [Mich05] [Breu10]. Despite the fact that the concept of serious games itself, might be considered as an oxymoron since “Games are inherently fun and not serious” [Kuip13], it is important to clarify that many works had been conducted in the field addressing the development of these games, always considering fun as a main component of them [Ritt10].

Several works addressing the design and development of systems for the rehabilitation of users with disabilities [Kiku07] [Burd00], as well as for the definition of models to express people’s capabilities and disabilities [Casa08] [Kakl13] had been carried out. However, hearing impairment is widely recognized as one of the most common disabilities [Vent82], there is a lack of work focusing on providing tools to ease the design and development of applications for the domain of auditory rehabilitation.

In this sense, the aim of this thesis work, is not only on providing an application that may be used to help on the performance of assisted therapy sessions, but to propose the enrichment of a software development process model to promote the implementation of serious games to support auditory rehabilitation. This leads to the definition of the following research question:

> *What elements, disciplines, or techniques may enrich a software development process for promoting the development of serious games to support auditory rehabilitation, in short time and with limited human resources involved?*

To the effects of this research question, it is important to consider as “short time”, a period of 4 to 8 weeks due to the pertinence usage of the software application towards the therapy program objectives, while as “limited human resources”, it is considered to involve maximum 5 people in the development team playing as much roles as it is required.

1.1. Hypothesis

The literature review that is summarized in the second chapter, made it possible to establish a Hypothesis aligned to the previously introduced research question:

H1.- The stages of a software development process model may be enriched with concepts on serious games design, rehabilitation, and Human-Computer Interaction techniques or tools, in order to promote the production of serious games to support auditory rehabilitation in short time and with limited human resources involved.

In order to prove the established hypothesis, a general objective is settled. This goal along with the specific objectives is presented in the next section.

1.2. General and Particular Objectives

The general objective of this work is to enrich a software process model to promote the development of serious games to support auditory rehabilitation, by adding to its stages game design elements, concepts on rehabilitation, and Human-Computer Interaction (HCI) techniques or tools.

For accomplishing this general objective, it is decomposed in shorter specific goals. These specific objectives, may be seen also as a list of steps to follow in order to accomplish the general one, and therefore to prove the established hypothesis. Thus, in Table 1.1, the specific objectives are listed in the format of a traceability matrix which must be updated in the Conclusions and Future Work chapter.

1.3. Scope and Limitations

From the definition of the objectives, it is possible to establish limits for this thesis work. The description of these guidelines allows focusing on what is important for proving the hypothesis, and what is out of the aim of the project.

Scope:

- Propose features to be added for enriching a generic software development process model. Those features must be related to the domain of rehabilitation, concepts on game design, and tools derived from a HCI approach.
- Participate on the design and development of applications and games according to requirements gathered from the attendance to speech therapy sessions, for acquiring experience in the development of such software tools.

- Propose a metamodel for expressing characteristics of a serious game to support auditory rehabilitation.
- Describe the activities, roles, and internal and external products that take part in the proposed enriched process.

Table 1.1 Specific objectives traceability matrix.

Objective number	Description	How was it accomplished?
1	Attend speech therapy sessions to observe and gather requirements to develop applications to support a rehabilitation program.	
2	Elaborate applications in compliance with the gathered requirements in order to acquire experience in their development.	
3	Compare software development process models from the literature.	
4	Choose a process model and review its characteristics.	
5	Model the use context, in terms of platform, user, and environment.	
6	Elaborate a metamodel of serious games for auditory rehabilitation.	
7	Identify concepts about game design, which may be useful for the realization of this project.	
8	Incorporate concepts on game design, rehabilitation, and HCI techniques or tools to the stages of the chosen process model in order to enrich it.	
9	Develop serious games to support auditory rehabilitation by following the enriched process model.	
10	Verify the proposal (enriched process model) and validate the developed products.	
11	Generate publications of national and international impact reporting advances and results obtained during the performance of this thesis work.	

Limitations:

- The explanation of the proposed process will be performed on terms of what to do, and not on how to do it, this is, following a descriptive approach.
- The verification of the process will be driven with support of developers and engineering students with different skills and profiles.
- For the validation of the products, only those serious games approved by the speech therapist are going to be considered, and also, it is important to point out that a reduced and variable number of patients are available to participate using the developed applications.
- The observation, requirements gathering, testing, and evaluation of concepts are going to be realized in the “Centro de Atención Integral a la Salud” (integral health-care center) of the Universidad Autónoma de Aguascalientes due to availability.

1.1. Research Approach

For the realization of this thesis work, a scientific research procedure is employed. Scientific research may be classified according to the object of study, the invested time, the nature of the gathered information for solving the problem, according to data gathering techniques, and according to the general objective among others [Sanc11].

According to the object of study, this work may be classified as applied research, since the resultant experience will be applied in field studies solving practical problems, this is, the experience and serious games developed are going to be tested and implemented for their evaluation with end users.

Because of the nature of the gathered information for solving the problem, this work may be classified as: quantitative research, since numerical and statistical information will be reported and interpreted on it, for instance time invested in the development of a serious game, time invested by the user in the usage of a serious game, points scored from a user in a play session, etc.; qualitative research, since non numerical information will be gathered and interpreted on it, like the characteristics of the enriched process, the roles involved, and the tools provided to promote the development of serious games; and exploratory research, since the main goal pursued with this project is to provide with tools and guidelines to address the problem of developing serious games to support auditory rehabilitation in short time and with limited human resources.

In regard to the extension of the study, this project may be classified as case study research, since it consists on the observation and study of a specified group of subjects and the elaborated tools and developed products, are tested within the same group.

And finally, in respect to the general objective, this thesis work may be classified as descriptive research, since it implies that process models and game design elements are observed, analyzed, and they form the baseline in which the proposal is based.

1.2. Structure of the Document

The remainder of the thesis work is organized in chapters that in turn contain sections. This first chapter, Introduction, provides a general panorama of the problem that is intended to solve, and about concepts that are important to be understood since they'll be used in further chapters. Also, on it, the research question, associated hypothesis, related general objective, and aligned specific objectives, are described.

The second chapter contains the state of the art, which is the result of the research that was carried out in order to recognize concepts on rehabilitation, meaning by this speech therapy, occupational therapy, and physical therapy; model driven development; process models, including linear sequential, prototype construction, incremental, spiral, component based, and the tailored process models, as well as a comparison of them; patterns; evaluation; serious games design elements; serious games for rehabilitation; and finally describing experiences on the development of serious games for acquiring skills on those projects that may later on work as baseline for the research.

Third chapter is aimed on the description of SEGA-ARM, a metamodel that was designed considering the experience gained with the former development of applications for rehabilitation and elements that were found in the literature, in compliance to a HCI approach, and expressed in accordance to a UML notation. This metamodel may be used as a support tool for stages in the process model that is described in the following chapter.

The fourth chapter presents the main proposal on this work, this is the enrichment of a generic software development process model with speech rehabilitation concepts, HCI based tools, and game design elements found on the literature. On its sections, a workflow of the application of the process is presented, the general view of the process is explained, and finally each stage involved on the process model is detailed in terms of the actors, activities, and products that take part on it.

Verification of the proposed process and validation of one of the developed products are described in the fifth chapter, along with the explanation of the experiments that were carried out to this effect.

Finally, the sixth chapter is dedicated to provide the conclusions to the realization of this thesis work reporting how the objectives were accomplished, describing if there are others to be addressed, and mentioning if the established hypothesis was proved, along with the future work.

2. State of the Art

The current chapter introduces concepts, methods, models and tools that are considered to be relevant for the purpose of understanding, designing, developing and testing applications for the rehabilitation of several types of disabilities, as well as the analysis of related works in order to set up guidelines for the proposal.

It is important to point out that even though this thesis is aimed on software engineering for serious games, it is necessary to define and understand concepts and practices for auditory rehabilitation to provide a complete background on the scope of the work.

2.1. Concepts on Rehabilitation

Rehabilitation, as defined by Brandt & Pope, is the process by which physical, sensory, and mental capacities are restored or developed in (and for) people with disabling conditions [Bran97]. When it comes to rehabilitation for auditory disabilities, it is covered mainly by speech therapy and reinforced by occupational and physical therapy, each one, branches of rehabilitation contributing with specific techniques for achieving a common goal. The success of a rehabilitation program depends on various factors: timing, patient diagnosis and treatment planning [Rego10].

2.1.1. Speech Therapy

Speech therapy, also known as speech-language pathology, is the rehabilitative treatment of physical and/or cognitive disorders resulting on the difficulty to perform verbal communication. Speech therapists assess, diagnose, treat and help to prevent disorders related to speech, language and other elements of communication [McDo12]. The ultimate goal of these specialists is to help patients to develop or recover reliable communication and other skills. Most speech-language pathologists work with determined age groups, such as children or elderly and focus on the treatment of certain communication problems, such as those resulting from strokes or hearing loss [BLS14c].

Speech therapy professionals' duties while diagnosing patients typically are the following:

- Communicate with patients to evaluate their speech and language skills.
- Use standardized tests or observe a patient while completing basic reading and vocalizing tasks to determine communication problems.

- Identify, propose and carry out an individualized treatment plan.

While treating patients, speech therapists typically do the following:

- Teach patients how to make sounds and improve their voices when possible.
- Teach alternative communication methods, when making sounds or improve voices is not possible.
- Improve patients' writing and reading skills.
- Involve patients' relatives on the treatment.

2.1.2. Occupational Therapy

Occupational therapy is defined by the American Occupational Therapy Association (AOTA) as the "use of purposeful activity with individuals who are limited by physical injury or illness, psychosocial dysfunction, developmental or learning disabilities, poverty and cultural differences or the aging process in order to maximize independence, prevent disability and maintain health" [AOTA81]. In general terms, occupational therapy consists on using selected activities, such as cooking, cleaning, game playing, and music, in order to gain or regain abilities [Reed88]. Occupational therapists' goal is to improve basic motor functions and cognitive and emotional abilities, as well as, compensate for loss of a function [McDo12].

Occupational therapy professionals' carry out the following tasks [BLS14a]:

- Observe patients doing tasks, ask questions and review their medical history in order to evaluate their conditions and needs.
- Propose a treatment plan for patients according to the types of activities and specific goals to be accomplished.
- Demonstrate exercises and help people with different disabilities to perform different tasks.
- Evaluate the patient's context based on the needs they have, and identify potential improvements.
- Educate a patient's family and employer about how to accommodate and care for the patient.
- Recommend special equipment and instruct patients on its use.

2.1.3. Physical Therapy

Physical therapy is defined in the Mosby's Medical Nursing and Allied Health Dictionary, as "the treatment of disorders with physical agents and methods to assist in rehabilitating patients and in restoring normal function after illness or injury" [Glan90]. Physical therapists help patients to improve their movement and provide a treatment to face chronic conditions or injuries.

When working with their patients', physical therapists typically do the following [BLS14b]:

- Review patients' medical history.
- Diagnose patients' dysfunctional movements by observing them stand or walk among other methods.
- Propose a plan for patients' rehabilitation considering their goals.
- Use exercises and equipment as treatment for improving patients' conditions.
- Evaluate patients' progress and modify the treatment plan according to the obtained results.
- Educate patients and their families about the recovery progress.

2.2. Model-Driven Development

Researchers on computer sciences had been working on mechanisms to raise the level of abstraction with which developers write programs [Atki03]. In this sense, the Model-Driven Development (MDD) approach, allows developers to model necessary functionalities and architectural characteristics of the systems, instead of requiring them to specify every single detail of them using a determined programming language [Mill01] [Fran03].

MDD is an Object Management Group (http://www.omg.org) initiative which main goal, is to improve developers' productivity and reduce time-to-market, meanwhile the challenge on it involved is in defining processes to transform the generated models to code for specific platforms [Send03].

In order to be considered as a Model-Driven Approach, MDD has, among other requirements, to be supported by a modeling language [Vand05], which is usually but not limited to the Unified Modeling Language (UML) [Jaco99].

The concept of MDD is closely related to Model Driven Architecture that consists on a paradigm that incorporates a standard for the establishment of the components that integrate an Interactive system development methodology, these components are:

- Models. A set of models to describe the elements that form a User Interface (UI): tasks, data modeling, user profiles, graphic presentation and behavior.
- Language. For specifying the models in some way the computer might process a User Interface Definition Language (UIDL) is required. That type of language allows designers and developers to exchange, communicate and share fragments of the specification so the software tools can operate on them. The modeling bases are independent to the UIDL, so the proposed solution might be implemented through different languages. The revision and comparison of several UIDLs has been carried out and is available for its consideration [Guer09].
- Software. A methodology must be supported by software tools and its interoperability has to be ensured at least theoretically.
- Approach. This refers to the paradigm to be used for giving an order to the UI development methodology steps. The process of design begins with tasks models developed under a gradual approach for ending in the definition of the UI [Cupp06]. An approach is based on the Cameleon Reference Framework [Calv03] which consider four development phases:

  1) Tasks and concepts (T&C): describe user tasks; they are concepts referent to data models that are required for doing these tasks. Task modeling is achieved often using notations such as CTTE [Pate01]. An example of a CTTE diagram is shown in Figure 2.1.

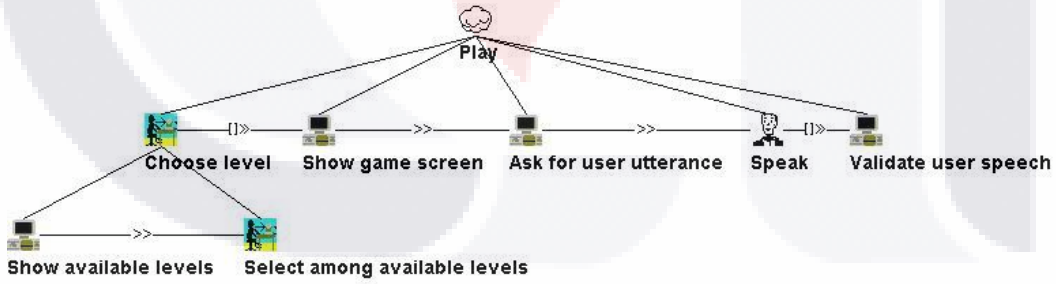


Figure 2.1 Example of a tasks model using CTTE notation [Pate01].

  2) Abstract User Interface (AUI): define abstract containers and interaction individual components. Tasks are associated to containers for its execution or to individual objects for its manipulation. An AUI is considered as an abstraction of a Concrete UI with respect to the interaction modality. At this level, the UI is

composed mainly by the definition of the system inputs and outputs but does not define the modality to be used (graphical, vocal, tactile). AUI modeling may be achieved by using slide editors, painting software or state of the art applications such as IdealXML [Mont07]. Examples of AUIs are shown in Figure 2.2.

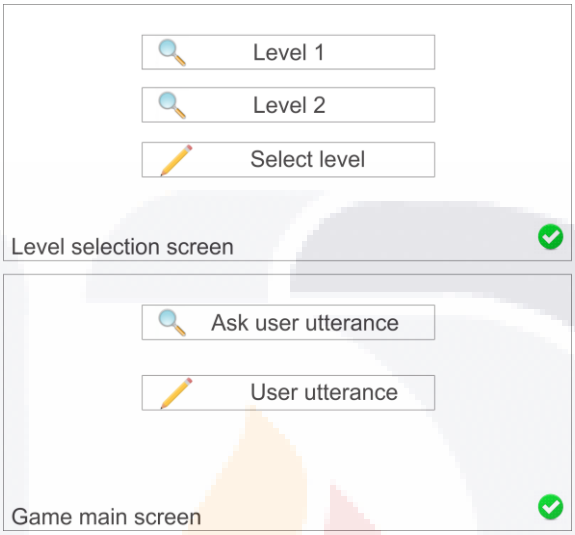


Figure 2.2 Example of Abstract User Interfaces using IdealXML notation [Mont07].

3) Concrete User Interface (CUI): the CUI defines an interaction modality and is composed by elements that describe it, Concrete Interaction Objects (CIOs) for defining the design widgets and the navigation through the interface. The CUI is computing platform independent and although it makes explicit the aspect and behavior of the UI, it is still a model only working for a particular environment. A CUI may also be considered as a reification of an AUI in the superior level and an abstraction of the Final UI with respect to the platform. Graphic design applications may be used in order to create CUI models. Figure 2.3 presents examples of this type of interfaces.

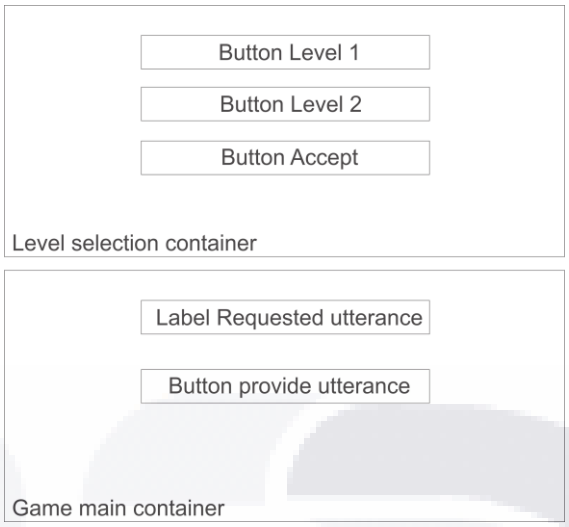


Figure 2.3 Example of Concrete User Interfaces.

4) Final User Interface (FUI): corresponds to the operational elements, this means, the runtime UI over a determined computing platform. This type of interfaces is one of the results of the codification stage of a Software Development Process. Figure 2.4 presents examples of CUIs.

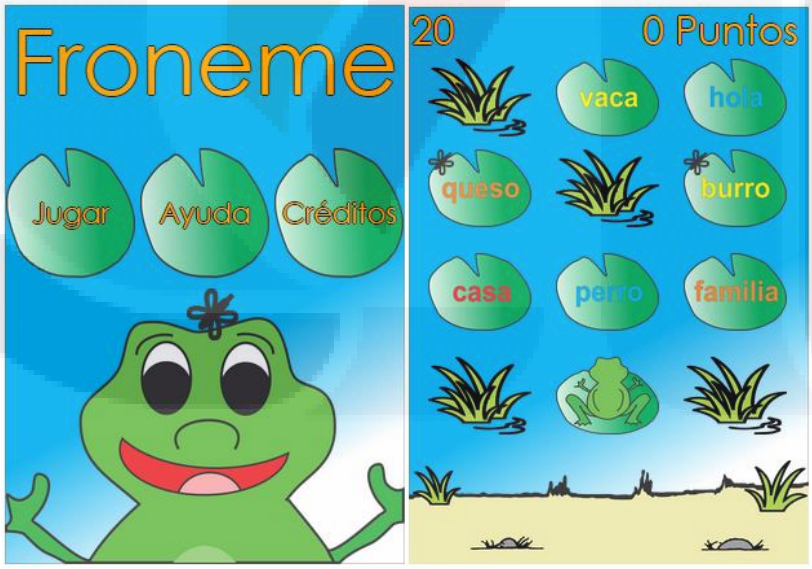


Figure 2.4 Example of Final User Interfaces.

In order to support the modeling of the above seen levels, there are methods and transformation rules for different use contexts. The transformational development of the UI

finds its motivations in the concept of heterogeneity of information systems. For this case, heterogeneity refers to the variety of use contexts for those which the UI was designed for. This heterogeneity makes a stand on the necessity of abstracting the pertinent details from specific contexts. It is possible to obtain specific representations from these abstractions. The advantage of accessing to those representations is the ability for reasoning about a unique model (task model) and to obtain many different UIs.

The main advantages that are encountered on the usage of MDD are [Stef07] [Uhl08]:

- Simplicity: Complex software products may be represented as an easy to read collection of models and documents.
- Agility: The different abstraction levels that may be represented using models empower project evolution through different process stages.
- Customizability: Modeling notation elements may be renamed and restructured as needed to ease the representation of systems.
- Consistency: Traceability and semantic consistency are guaranteed, from the earliest stages of prototyping until the final execution.
- Verification: By defining properties that must be satisfied, automatic mechanisms and model checking techniques may be provided for verifying the produced models and its modifications.
- Executability: During the different levels of abstraction that are covered, execution code may be generated. This code can be as abstract as text descriptions, or as concrete as actual runtime implementations.

2.3. Process Models

Software applications are complex products, difficult to develop and test. This complexity leads on the necessity of counting on the definition of software process models that describe activities, techniques and methods to support development [Fugg00]. As soon as the term software process was coined, workshops and events were carried out for the enrichment of the discipline, and even standardization organizations such as ISO and IEEE started important efforts centered on it. In this sense, and for the purposes followed by this work, it is relevant to mention ISO 12207 (System and software engineering - Software life cycle processes), that can be used [ISO08]:

- By an organization, to establish an environment of desired processes

- By a project, to select, structure and employ the elements of an established set of life cycle processes to provide products and services.
- By an acquirer and a supplier, to help develop an agreement concerning processes and activities.
- By organizations and assessors, to perform assessments that may be used to support organizational process improvement.

It is important to point out that this International Standard includes clauses that define the requirements for the system life cycle processes and requirements for specific software life cycle processes, along with annexes containing examples and guides for its implementation. Processes within that standard are described in terms of their following attributes: title, purpose, outcomes, activities, and tasks. This form of description is defined by ISO 24774 (System and software engineering - Life cycle management – Guidelines for process definition) [ISO10].

ISO 12207 also provides the definition of software life cycle models, introducing the concept of stages. According to this standard, the life of a software product can be modeled by a life cycle model consisting of a sequence of stages that may iterate and/or overlap, as appropriate to the project's scope. For the description of each stage, it is necessary to declare its purpose and outcomes.

In general terms, a software process involves three main stages: definition, development and maintenance. The definition stage is aimed to describe *what* is going to be developed, meaning by this, the identification of the information to be processed, the desired performance and functionality, system behavior, user interfaces, and criteria to evaluate correctness of the system. Three main tasks are carried out during this first software design stage, information engineering, software project planning, and requirement analysis [Pres05] [Boeh88] [Feil93].

The development phase focuses on *how* a software product is going to be elaborated, this is, definition of data structures, software architecture, procedural details, the manner in which models are going to be translated into code, and how tests are going to be carried out. The main tasks to consider during this stage are, software design, codification, and software testing [Pres05] [Hump89].

The maintenance stage is aimed to describe the *changes* that occur during mistakes correction, adaptation, and improvements. Thus, four basic types of changes are considered as part of this phase, correction, adaptation, improvement, and prevention. The

main tasks to be carried out during the maintenance stage are, Software project control and tracking, quality assurance, usage of metrics, and risk management [Pres05] [Boeh91]. The following subsections present some of the process models structures that may be found on the literature.

2.3.1. Linear Sequential Model

The linear sequential software process model, also known as cascade model, consists on a series of stages related one to another. The linear model is the oldest approach and the most commonly used. Often, this process model is described as a sequence of the following phases: Requirements gathering and analysis, design, codification, testing, and maintenance.

The main disadvantage encountered on the application of this approach, is that the client or user rarely exposes all of his/her needs. Thus, its usage is recommended only when all of the requirements are known. The structure of this process model is presented in Figure 2.5. Strictly speaking, follow this process model, implies that not feedback may be provided by the client once the development is started. Also, it is important to point out that for long term projects, the first version of the software products won't be available soon [Pres05] [Carm95].

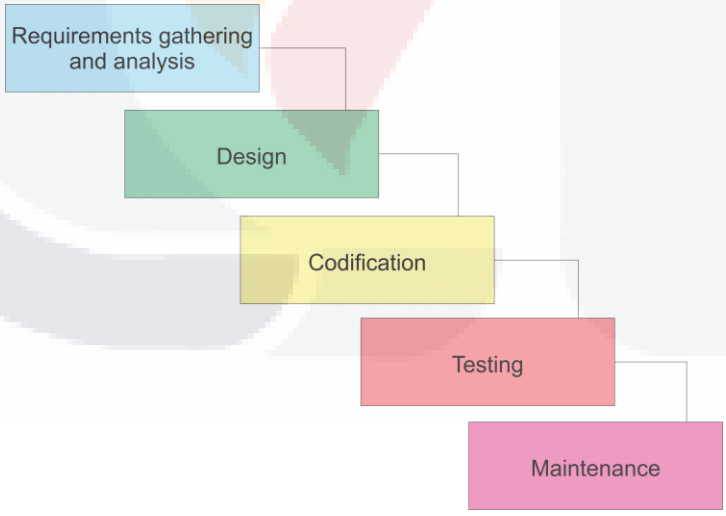


Figure 2.5 Structure of the Linear Sequential Model.

2.3.2. Prototype Construction Model

Unlike the linear sequential model, when instead of well-defined requirements, only a general vision of the objectives to achieve with a software application is provided by the client or user, a prototype construction model may be followed. This process model, is mainly conformed by three iterative stages, the first one in which the client exposes the goals that are pursued with the software tool to be developed, the second one in which the analysis and development team builds a working prototype to be presented to the client during the third phase in which changes and improvements are requested. This structure is illustrated in Figure 2.6. The prototype construction model's main advantage is that it supports the involvement of the client on the software development, carrying the risk of presenting low quality functional prototypes [Pres05].

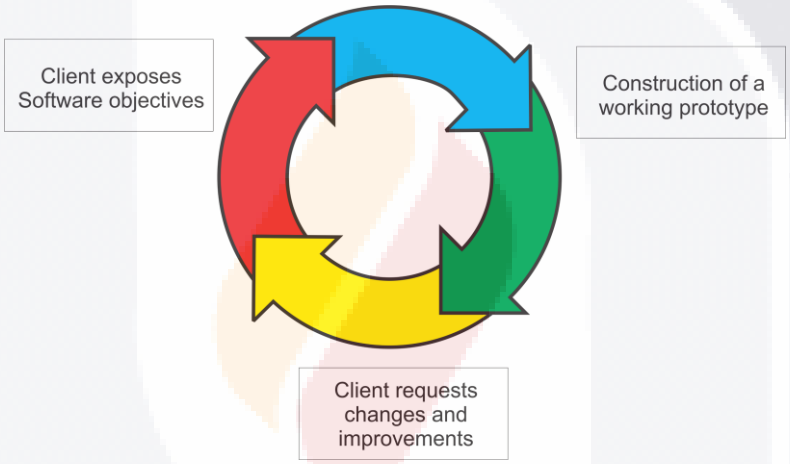


Figure 2.6 Structure of the Prototype Construction Model.

2.3.3. Incremental Model

The incremental software process model is the result of a combination between the linear sequential model stages, and the generation of partial products and feedback that are present on the prototype construction model. The main advantage of the incremental model when compared to the prototype construction one is that in this one, each version that is presented to the client or user, is a fully operational product that has passed through a testing phase, allowing ensuring its quality.

When the due date for delivering the software product may be updated and the client or user can be involved on the process, this approach may be applied. The structure and flow of the incremental model are shown in Figure 2.7 [Pres05] [Larm03].

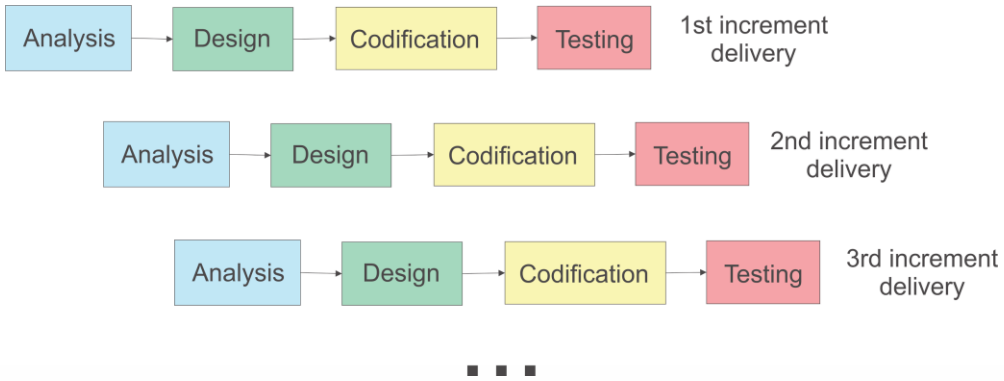


Figure 2.7 Structure of the Incremental Model.

2.3.4. Spiral Model

Originally proposed by [Boeh88], the spiral model is an evolutionary incremental approach that combines the iterative nature of the prototypes construction model with systematic aspects of the sequential one. This process model consists on a series of stages that allow the production of functional versions of the desired product. Early deliveries may be presented as low level (paper) prototypes, while the last ones should be more complete versions of the designed system.

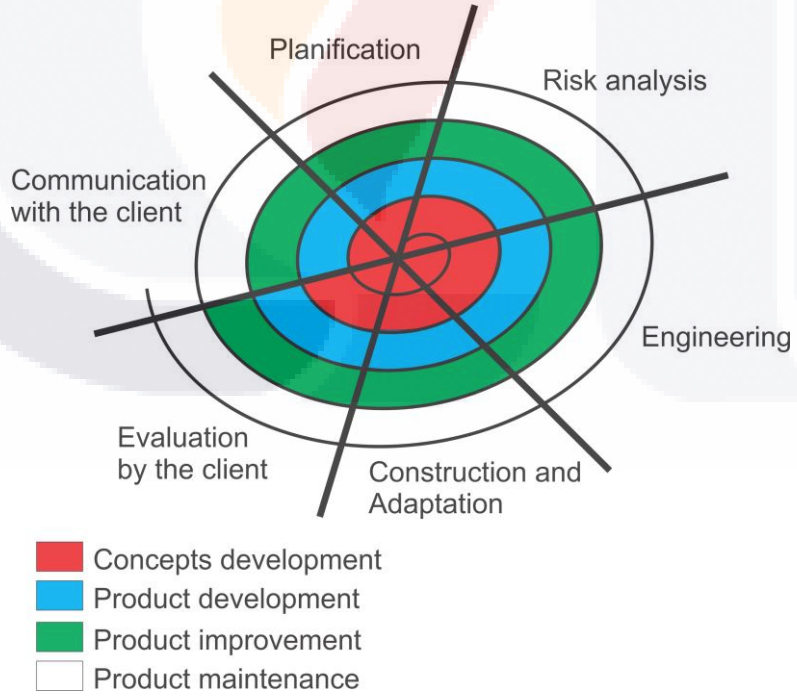


Figure 2.8 Structure of the Spiral Model.

Often, the spiral model is divided into task areas such as communication with the client, planification, risk analysis, engineering, construction and adaptation, and evaluation by the client. It is recommended to follow this process model approach for identifying and reducing risks, and for applying well defined steps in a more real, iterative manner. The main disadvantage of the implementation of this model is that it requires that the design team counts on high skills for determining risks. Figure 2.8 presents the basic structure of the spiral process model [Pres05] [Will88].

2.3.5. Component Based Development Model

Object Oriented Programming (OOP) provides characteristics that promote the definition of a component based development model [Pres05] [Ning96]. If well designed and correctly implemented, the objects created under this paradigm may be reused for different applications and systems. This process model is very similar to the spiral or the incremental ones, with the difference that in this case, software applications are configured from existing components. In this process model, software engineering begins with the identification of candidate classes, by analyzing the requirements of the application that is to be developed.

The usage of this approach, supposes the creation of a repository of classes or objects. The main goal that is pursued with the implementation of this process model is to decrease the time invested on codification tasks, by increasing the time spent on analysis and configuration. In order to document the objects and classes that are available for reutilization, modeling languages such as UML [Jaco99] are employed.

2.3.6. Tailored Process Models

The creation of standards such as ISO 12207 [ISO08] and ISO 13407 [ISO99] among others, for process models, resulted on the definition of compatible tailored process models for specific domains, and for particular working environments according to available roles and resources.

For instance, in [Smit04], an ISO 13407 compatible process model for the development of cross-cultural websites is presented as a series stages with activities, detailing tasks to be carried out as part of each of them. The described tasks are designed according to the domain of use of the system to be developed.

Moreover, in [Viei09] a process for the design of context-sensitive systems (CSS) is proposed, defining for it roles, activities and tasks, by using the SPEM notation [OMG08]. The activities that were described by the authors refer to the specific domain of CSS development.

2.3.7. Comparison of Process Models

In previous subsections, an overview of several process models is provided along with their main characteristics. Table 2.1 presents a tabular comparison of them, as a brief resume of the whole section. It is important to notice that component based and tailored process models are not considered during this comparison since, although they are relevant to the aim of this work, they do not lay on an specific structure and may be adapted to the particular characteristics of the project they are applied to.

Table 2.1 Comparison of Software Development Process Models.

Process model approach	Linear Sequential	Prototype Construction	Incremental	Spiral
Basic stages	Requirements gathering and analysis Design Codification Testing Maintenance	Client exposes Software objectives Construction of a working prototype Client requests changes and improvements	Analysis Design Codification Testing	Communication with the client Planification Risk analysis Engineering Construction and adaptation Evaluation by the client
Recommended usage	When all of the project requirements are defined	When only a general view of the software objectives is provided and the Client needs to track the development	When the due date of the project may change and the User can take part on the development process	When it is necessary to deliver completely functional prototypes. And the development team is skilled on analyzing and

		process		reducing risks
Scalability	Not supported	According to Client requests, after presenting a partial prototype	Supported by creating a new delivery	Supported by executing a new complete loop
Most demanding stages	Requirements gathering and analysis	Construction of a working prototype	Analysis and design	Planification and Communication with the Client
Partial products generation	Not supported	Presentation of partial prototypes without quality assurance	Fully implemented prototypes meeting most requirements	Presentation of partial prototypes with quality assurance
Client or User inclusion	Only during the requirements gathering and analysis stage	While exposing software objectives and for reviewing and suggesting changes and improvements	During analysis and testing stages	During the communication with the Client and the evaluation stages
Reutilization of documents and/or code	Not supported due to specificity of the software product and documentation for the current project	Hard to be achieved since the final product is result of customization through several presentations and improvements	Hardly, unless early stages of a project may work for another one	Only if on early stages of the process it is considered. In that case, during the planification stage it has to be determined

2.4. Patterns

A problem that occurs often and a solution that is given to it, is defined as a pattern [Alex77]. There are specific patterns for a determined domain, like architectural patterns, social behavioral patterns, and sewing patterns to name a few, but in the scope of this work, the ones of interest are mainly design and interaction patterns.

According to "The Gang of Four"[1], a design pattern is formed by four main elements [Gamm94]:

1. *The name of the pattern:* allowing description of the problem with its solution and consequences. As soon as this is specified, the design vocabulary is enriched, giving the opportunity of a more abstract design.
2. *The problem:* describing when to apply a determined pattern, and explains the problem itself and its context. Sometimes, common particular design problems are described along with a series of constraints to take into account while using the defined pattern.
3. *The solution:* describing design elements, relationships, responsibilities and collaborations. The solution is not a specific implementation nor design description but a template that may be applied to similar problems. The pattern provides an abstract description of a design problem and a general solution to it in terms of arrangement of elements (e.g. objects and classes).
4. *The consequences:* results, advantages and disadvantages of applying the defined pattern. The definition of consequences allows to evaluate and to understand costs and benefits of the given pattern. Often, consequences on software development are referred to the balance of space and time, but may also represent language or implementation issues.

Following these principles, in [Gamm94] a list of 23 design patterns was proposed, grouping them according to their purpose, as creational, structural and behavioral patterns. Based on this proposal, several works had been developed addressing extension, specification and description of design patterns for a particular domain or problem [Heer06] [Gay05] [Land03].

For the domain of interaction, specific patterns for web, mobile and desktop applications had been proposed as a manner to give designers and developers a guide for the solution

[1] A name given to the group of authors of the book Design Patterns [Gamm94], integrated by Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides.

of common problems. For instance, a collection of interaction patterns for web sites development was provided by Van Welie in [Van03], using for their definition a pattern language consisting of name, problem, solution, when to use, application, rationale, and more usage examples.

Taking into account the definition of a pattern, it may be possible to find and describe patterns for specific domains rather than software design, for achieving the goal of understanding a common problem, and provide a solution. Such is the case of image analysis, in which shape, color, and texture patterns are identified in order to allow recognition of objects, resulting on the implementation of algorithms for facial recognition, border detection and object tracking, to mention a few [Sama92] [Gols90] [Yilm06].

Likewise, in domains like the educational one, learning patterns are a valuable tool for characterizing learning styles, and therefore define strategies to allow better understanding and acquisition of information by the students [Verm04].

Similarly, for the domain of rehabilitation, the proposal of patterns to describe common conditions or impairments, along with known solutions or strategies adopted to attend them, may support the design of therapy programs. Examples of works addressing the definition and analysis of patterns for physical and cognitive rehabilitations are [Hara00] [Cicc84]. Works addressing the definition of patterns for supporting auditory rehabilitation were not found.

2.5. Evaluation

Once Software process activities are defined and performed, it is necessary to drive an evaluation in order to determine if the process successfully allowed the development of a product, and to confirm that the developed products meet the requirements that were gathered on the earlier stages. Those evaluations are in fewer words, V&V: verification of the process and validation of the product, and may be performed at the very end of the execution of the process or at each stage, in order to check the different internal and external products (i.e. documents, components, designs, and code, among others) [Kung08].

On one hand, verification consists on checking that the developed application meets the specified requirements, and is related to software quality, while on the other, validation is a sub process that allows to check whether the application specification is aligned to the users' needs, and is related to user satisfaction [Kung08] [Pare08].

Verification involves activities such as testing and inspection, and may be supported by the usage of checklists and traceability matrices, while validation is a more subjective process that consists mainly on make assessments of how well the developed application satisfies the real needs of the users, and it includes activities for modelling requirements and evaluation with users, among others [Wall89] [Kung08].

Since the beginning of a software project, it is necessary to establish metrics [Fent00] for carrying out both, validation and verification, taking into account the nature of the project itself, the products that are expected to be developed, the desired level of quality, and the available resources for this measure. Examples of metrics for evaluating a software process are: invested time, generated documentation, reusability of the product, and satisfaction of requirements rate, among others; while for measuring the quality of software products, usability, user experience and functional testing may be carried out.

2.6. Serious Games

In several contexts such as healthcare, military and educational, games have been applied as a way to improve skills and for training professionals [Arna13] [McDo06] [Gros07]. This type of games are called serious games [Abt87] [Mich05] [Breu10], and may be either digital or not. Despite the fact that the concept of serious games itself, might be considered as an oxymoron since “Games are inherently fun and not serious” [Newm04], many works have been conducted in the field addressing the development of serious games, always considering fun as a main component of them [Ritt10].

To define a serious game, [Zyda05] begins by defining a game as a physical or mental challenge, that is played according to specific rules, with the objective of amusing or entertaining the participant. Then, continues to build the definition of a videogame as a mental challenge that is played through a computer according to certain rules for entertainment, fun or to achieve a goal. And finally, provides the definition of a serious game as a mental challenge, played through a computer according to specific rules, which uses entertainment mechanisms to achieve training or capacitation goals.

The interest on the usage of serious games for rehabilitation lies on the fact that is has been demonstrated that they increase motivation towards therapy sessions, which represents a major problem due to the repetitive nature of exercises [Rego10].

Note that in the context of this particular work, when serious games are mentioned, it is done as a reference to videogames for the purpose of rehabilitation.

Concerning serious games design, an effort has been realized by several authors, in order to abstract components that lead to the definition of a guide on what to be aware of when developing a game with these characteristics [Aldr09] [Dete11]. However there is not yet a consensus on the components or elements that should be considered when performing design and implementation, Table 2.2 presents some of the design elements that are usually taken into account for the development of games [Aldr09] [Dete11] [Tayl02] [Sale04] [Sanc09].

Table 2.2 Game design elements in the literature [Aldr09] [Dete11] [Tayl02] [Sale04] [Sanc09].

Game Design Element	Description
Characters	The definition of characters allows user engagement to the game, and provides the possibility to define actions. Helps improving the player experience. Considers both, player and non-player characters.
Narrative description	The rationale of the game described in a narrative way, promotes the immersion of the player, helps to define the needed resources, characters, game mechanics and challenges.
Challenges	The definition of challenges to be faced during the gameplay, gives the player the opportunity to try his/her skills and to compete either with a partner or with the game itself.
Type of view	Determines the level of immersion that is decided to use in the game. The most used point-of-view perspectives for gameplay are: first-person, third-person, third-person trailing, overhead, and three-fourths isometric.
Constraints	Constraints determine characteristics of the challenges (e.g., limited time or resources, competitive and sequential activities).
Rewards	Establishing a reward system (e.g., points, badges, levels, and leaderboards) allows improving the player experience and motivating him/her to achieve a goal.
Punishments	As with rewards, the establishment of punishments, such as a

	decrement of points, and losing a life in the game, motivates the user to achieve a determined goal.
Interactive elements	The definition of interactive elements supports the immersion of the player on a virtual world. Their definition helps to design player tasks.
Feedback	For each action performed by the player, the game provides a reaction. In this way, the user is aware of the consequences of his/her acts on the game and an appearance of continuous dialog is given to him/her.

Moreover, Longstreet & Cooper [Long12] proposed a conceptual guide for development of serious games to support higher education, their conceptual definition includes most of the design elements reported on Table 2.2, but a structural notation was not provided and the relationship between the involved elements is only explained in a narrative manner, making its reutilization complicated.

2.7. Serious Games for Rehabilitation

Several authors had been working addressing the problem of development of serious games for rehabilitation [Scho11] [Burk09] [Lang11], however, as mentioned in the Introduction chapter, the main goal that is pursued with this work is not only to develop a tool to support the process of auditory rehabilitation, but to provide a guidance for developers and designers to ease the implementation of that type of software applications. Therefore, analysis on the already existent serious games and its development is only carried out in order to abstract and determine characteristics that this proposal must consider.

Most of the serious games that were found on the literature are aimed to provide a solution or auxiliary tool for physical rehabilitation. In this regard, the proposal of [Burk09], attends upper limb rehabilitation for patients that had suffered a stroke, by delivering a series of serious games for different platforms, starting from the premise that these games, may support the rehabilitation process by increasing the motivation of the patients towards the therapy activities. For the development of the serious games, it is no reported that the authors followed a determined methodology nor software process, but they did consider the inclusion of game design elements such as those that were reviewed in Section 2.6 i.e.

Challenges design, scoring mechanisms establishment, simulation environment definition, punishments and rewards description, and feedback providence. In this proposal, evaluation was performed via usability testing, reporting a positive feedback by the users towards the usage of the given serious games.

Moreover, the work in [Scho11], reports the development of a serious game for supporting the rehabilitation of patients with chronic pain of the lower back and neck. To this purpose, the authors use hardware for tracking the patient's movements and bio signal data, and Unity as game engine. In this proposal, some of the game design elements that were mentioned in Section 2.6 are used (Actors identification, feedback providence, rewards and punishments description, actions description, and narrative description), but it isn't reported that a process model was followed for the implementation of the serious game. An evaluation was driven in terms of user experience, throwing that the users found it enjoyable to use the serious game, and evolution of the rehabilitation program, getting positive results after 4 weeks of usage.

Also, in the proposal by [Lang11], a software tool was developed with the aim of supporting the process of physical impairment, specifically for balance training of adults. The proposed solution relies on the usage of hardware for tracking the user's movement, and software to work as a game engine to enable the development of a tool for its particular objectives. Although it is not explicitly said that a process model was followed for the development of the application, it was declared that the product was elaborated as a prototype which may mean that a prototype construction process model was considered but not software engineering documentation was generated. The evaluation section was focused on the contribution of the developed serious game towards the rehabilitation process, concluding that valuable positive input was provided by users and therapists, but usability test must be performed to determine the feasibility of the application of this type of tools on rehabilitation programs.

When it comes to auditory rehabilitation, there are also proposals reported on the literature [Loai13] [Lv15], as well as others available for their download through mobile application stores or websites. Table 2.3 presents a comparison of some of the found applications in terms of if it is reported that a process model was followed for their development and the design game elements from Section 2.6 that are on them encountered.

Table 2.3 Comparison of found applications in terms of compliance to a development process, and of game design elements usage.

Characteristic	Speech essentials	Apraxia - Early intervention 1	Naming therapy Lite	Phonological Awareness: Rhyme Time Picture Match-up
Reports to be developed under a process model	No	No	No	No
Available by December, 2015 at	Android Play Store	Android Play Store	Apple App Store	http://www.quia.com/mc/334099.html
Characters	A robot as non-player character	None	None	None
Narrative description	None	None	None	None
Challenges	Find words in a word search puzzle, memory game with images and sound	Recognize words, images and geometric shapes	Guess the name of a determine object that is presented in an image	Find and relate words that rhyme from a set
Type of view	Table top view	Table top view	Table top view	Table top view
Constraints	Not declared	Figures are supposed to be set according to geometric shapes	The user is supposed to name the objects that are presented in	Not declared

			images. The correctness is evaluated by the therapist, not by the app	
Rewards	Score mechanism	Score mechanism, reward sounds and images	Reward sounds	Prompt to tell one the user finished the game
Punishments	None	None	Punishment sounds	None
Interactive elements	Cards with letters and images	Cards with words, and images that play sounds once they are touched	Images describing objects, buttons to play sounds whether describing the objects or giving clues about their names	Cards with words and images
Feedback	Sound, and score mechanism	Sound, and score mechanism	Sound	Images to tell if the selection either is correct or incorrect

As it may be seen in the comparison, the available applications, as well as those found on the literature, as a general fact do not report to were developed by following a process model. This may result in lack of documentation, difficult reutilization of resources for further implementations, and uneasy testing. It was also found, that most of the applications certainly address game design elements that were described in Section 2.6, but none includes all of them, especially the narrative description, actors, and punishments. This leads to conclude, that most of them are not actually serious games to

support rehabilitation, but applications for rehabilitation that include some gamification [Dete11] characteristics.

2.8. Towards the Enrichment of a Process for Developing Serious Games to Support Auditory Rehabilitation

As a first step to enrich a process with concepts, tools, and definitions, it was decided to assist to speech therapy sessions in order to observe and understand the necessities of the potential users towards an application to reinforce what is practiced with the specialist. Then, as a result of the observation and through the interview with the speech therapist, requirements were gathered for designing and developing applications and gain in this way experience in the tasks involved.

In this exercise for the identification of activities, 8 engineering students participated for a period of a month, by analyzing a series of requirements, interpreting them, and then following different approaches according to their skills, developing 3 software applications out of 5 that were required, and which screenshots are shown in Figure 2.9.

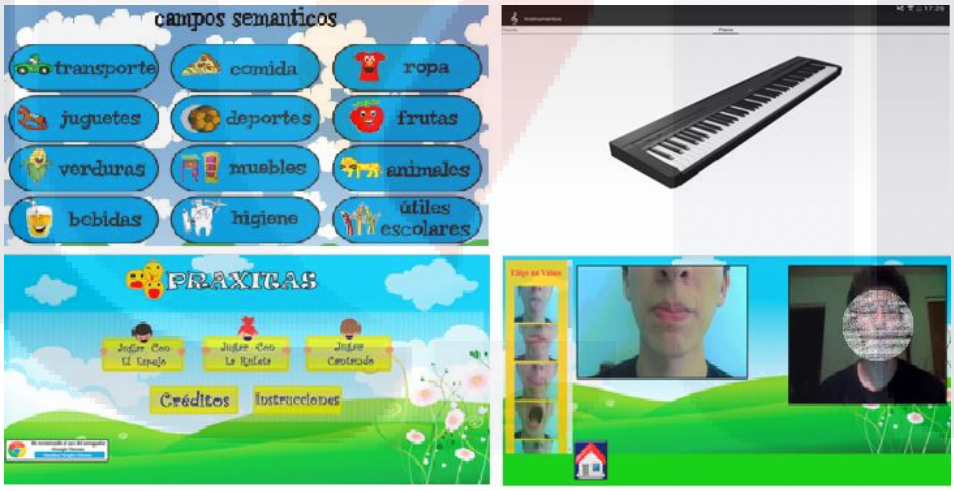


Figure 2.9 Screenshots of the developed applications as a first approach.

From the observation of the followed processes and delivered products, it was possible to notice that even though the instructions were to develop serious games, none of the produced applications included the definition of challenges, the description of actors, the establishment of scoring mechanisms, nor rewards to be given to the user. For this, it may

be consider as necessary to provide the development team with guidelines for adding game design elements for successfully meeting the requirements.

Also, during the experiment, it was possible to see that in absence of a clear definition of minimum required documentation, some of the developers invested most of the time trying to find out useful generic software development process models, techniques, methodologies and tools, while some others skipped early design stages since they reported to think that the given time to accomplish the whole project was very short, resulting this on poor project planification, and later, on projects that weren't finished.

Hence, it was possible to conclude from this, that if well the usage of a generic software development process model worked fine for achieving at least in a partial way the objectives of some projects, a process model enriched with HCI, rehabilitation, and serious games design concepts, may be needed for accomplishing them completely. In compliance to this idea, Chapters 3 and 4 present the proposal of this work along with its description.

3. SEGA-ARM: A Metamodel for the Design of Serious Games to Support Auditory Rehabilitation

One of the objectives that were presented on the introduction chapter, consists on proposing a definition of modeling elements along with their relationships and rules in order to enable the creation of semantic models, that may later result on the design of serious games with the purpose of supporting the process of auditory rehabilitation, meaning by this, a metamodel [Koch03] for the design of serious games to support auditory rehabilitation. Thus, in this chapter, a metamodel (SEGA-ARM) is proposed, considering for its definition concepts related to context, auditory rehabilitation, planned therapy, performed therapy, and serious games design elements.

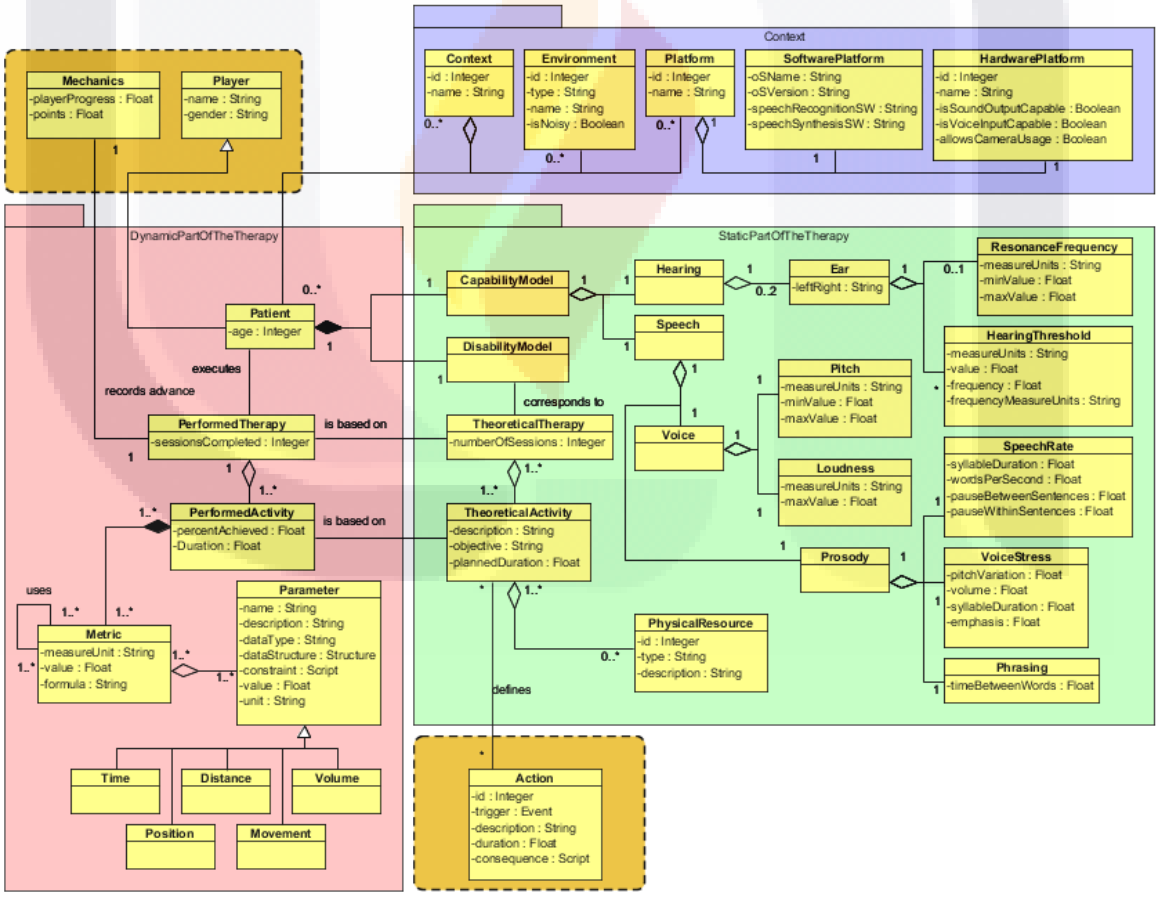


Figure 3.1 General view of the SEGA-ARM metamodel.

The proposed metamodel, shown in Figure 3.1, is divided for its better comprehension and usage into three packages, presented in Sections 3.1, 3.2 and 3.3, and an extension introduced in Section 3.4, each one of them identified by a different color and name, and designed to contain elements that support the design of serious games for auditory rehabilitation. In the rest of this chapter, SEGA-ARM is described in detail and explained through a case study of a serious game design for supporting the rehabilitation of a patient with an auditory disability.

The case study consists on a scenario in which the design of a serious game is required in order to support the therapy sessions conducted by a speech therapist for a six year old male patient diagnosed with deep bilateral hypoacusia who has recently received a cochlear implant. The therapist has already performed tests over the patient to determine his exact condition and counts on a full record describing his capabilities towards hearing and speech skills. The intention with the required game is to extend and complement the therapy sessions that the patient already receives with sessions conducted by himself in his home as those suggested in [Ittl12]. The language therapist wants the game to reinforce one of the therapy activities that the patient finds to be repetitive since it has to be performed in almost every single session.

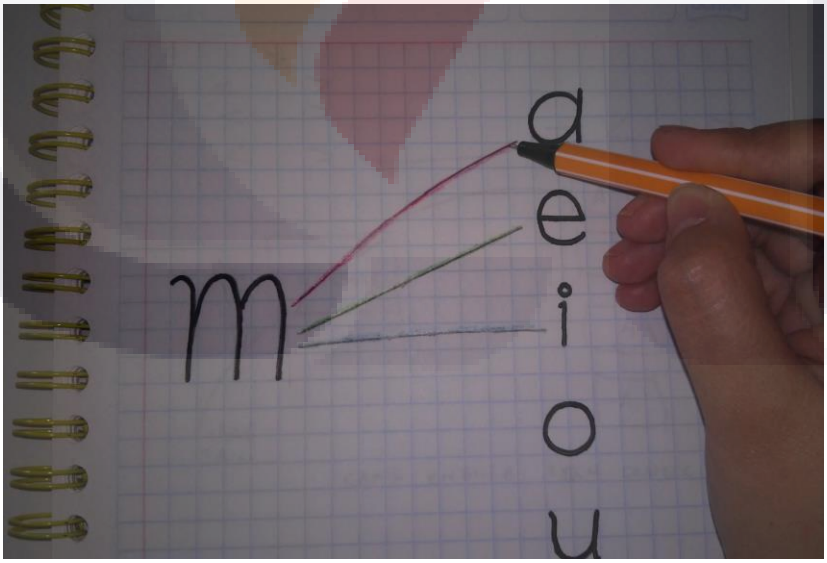


Figure 3.2 Example of an exercise to be considered as part of a case study.

The exercise consists on that the therapist chooses a series of phonemes sharing a consonant (e.g., ma, me, mi, mo, and mu), write them down on a notebook as shown in

Figure 3.2, putting the consonant on one column of the page and the vowels in another next to it, then asks the patient to pronounce each one properly while connecting the consonant with the vowels with lines, and then repeat this task several times and with different phonemes. This activity is designed to be performed in approximately ten minutes and also considers phoneme visual recognition by the patient. It is also known by the therapist that the patient has a 10.1" tablet with Android 4.2 OS. The information that the specialist possesses has been structured in terms of the here proposed metamodel as a way to facilitate the communication between the speech therapist and the game development team.

3.1. The Context Package

Shown in Figure 3.3, the context package was designed to understand the ***Context*** [Calv03] in which the user interacts with the system, and to provide the developers with a brief description of relevant characteristics to make decisions about what interactive modality to implement, and resources selection among others. The ***Context*** is identified by an id and a descriptive name, and formed by an ***Environment***, a ***Platform*** and a user (***Patient***).

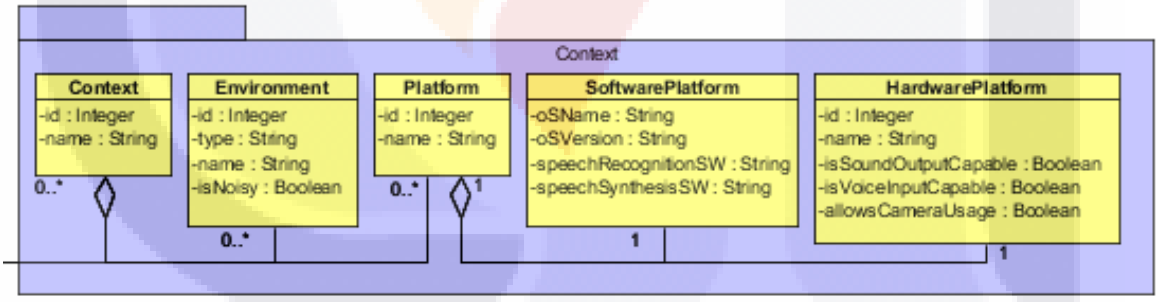


Figure 3.3 Package of SEGA-ARM for expressing the context.

The ***Environment*** describes the physical place in which the interaction occurs. For the specific domain of this work, it is necessary to point out if it is noisy or not, and to give it a name, an id and a type (i.e., exterior or interior).

The ***Platform*** definition is carried out to determine the characteristics of the device that is going to host the application, and is formed by a ***SoftwarePlatform*** and a ***HardwarePlatform***. The ***SoftwarePlatform*** represents those characteristics relative to the

software in the hosting platform i.e. operating system name and version, as well as the speech recognition/synthesis software on it installed, while the ***HardwarePlatform*** class defines characteristics regarding the hardware of the hosting platform and relevant to the domain, such as the possibility of using sound outputs, voice inputs or a camera.

In our example, the context in which the therapy sessions take place is defined by a therapist's office as a non-noisy, interior environment, has not platform and the user role on it is taken by the patient, meanwhile, the context of use of the application that is intended to be developed, is composed by the patient's home as non-noisy interior environment, a 10.1" tablet with Android 4.2 (Jelly Bean) operating system, audio output, voice input and camera usage enabled, with android speech recognition/synthesis software as platform, and the patient as user. Note that even though the characteristics of the context must be taken into account by the serious game's designer; the implemented solution may work under other similar contexts as well.

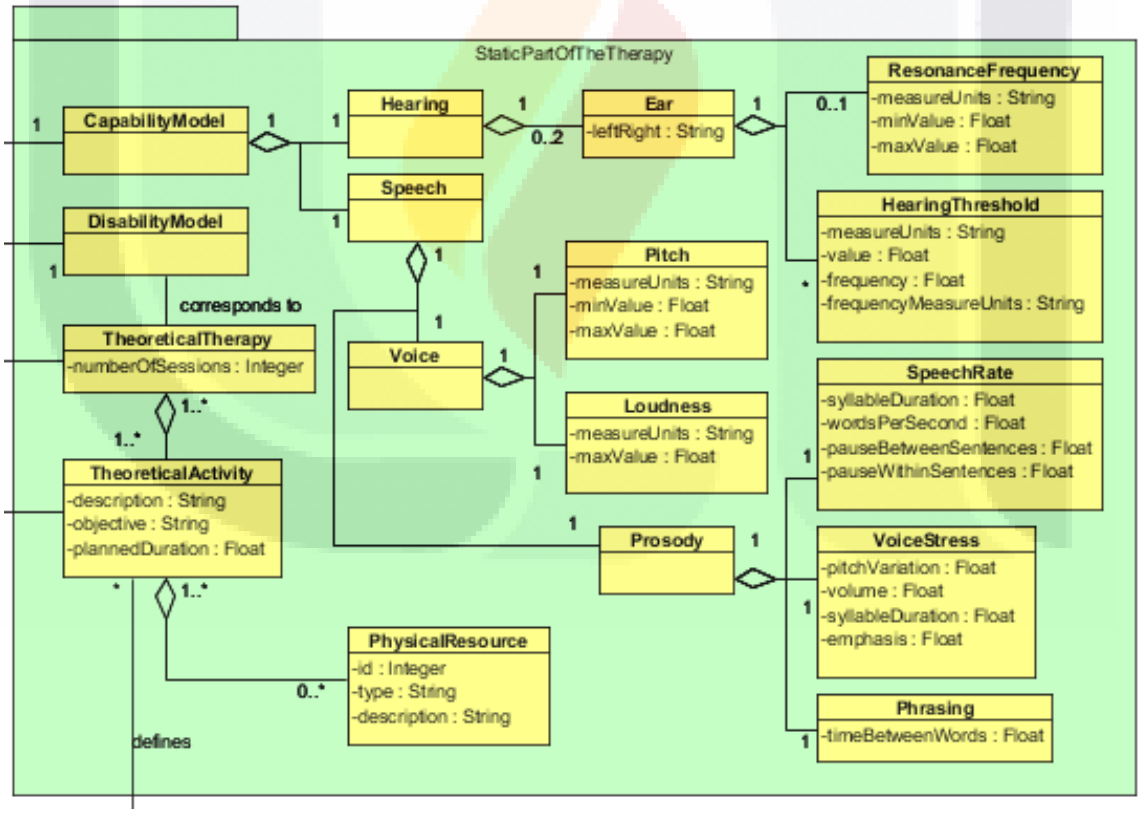


Figure 3.4 The static part of the therapy package of the SEGA-ARM metamodel.

3.2. The Static Part of the Therapy Package

The purpose that is pursued with the design of this package that is presented in Figure 3.4 is to provide the speech therapist with a semiformal notation to express the characteristics of a therapy plan, keeping that information structured in a way that results understandable for serious games designers. The definition of this part of the model was inspired mainly by the work done in [Kakl13] for modeling a Virtual User, extended to allow the expression of a full auditory therapy session program, and enriched with concepts found out to be relevant for patients of auditory rehabilitation on the literature [Chin12] [Lend07] and through the attendance to speech therapy sessions.

Abstracted from the full capability model [Kakl13] presented in Figure 3.5 that describes physical, cognitive and behavioral user characteristics, for the specific domain of this work, the ***CapabilityModel*** presented as part of this paper, only considers the elements related to speech and hearing.

The only parameter in the ***Hearing*** container is the ***Ear*** that in turn acts as one for the values related to each ear of the user: ***ResonanceFrequency*** and ***HearingThreshold***. The ***Speech*** container includes ***Voice*** and ***Prosody*** elements. ***Voice*** elements include ***Pitch*** and ***Loudness***, each one with its measure units and values; while ***Prosody*** is formed by ***SpeechRate***, ***VoiceStress*** and ***Phrasing***.

A ***DisabilityModel*** [Kakl13], that is presented in Figure 3.6, allows the description of all the disabilities of the user as well as the affected by them tasks. For each disability, a name and a type (e.g. motor, auditory, and vocal) must be provided, and it is related to affected tasks. For a determined ***DisabilityModel***, a ***TheoreticalTherapy*** may be proposed.

A ***TheoreticalTherapy*** is a full rehabilitation plan and consists of a certain number of sessions and a series of theoretical activities. A ***TheoreticalActivity*** is in turn, a series of actions with a common objective to be performed by the patient either using or not ***PhysicalResources*** (e.g., a mirror, sticks or cards). The planned duration, objective and description are attributes of a ***TheoreticalActivity***.

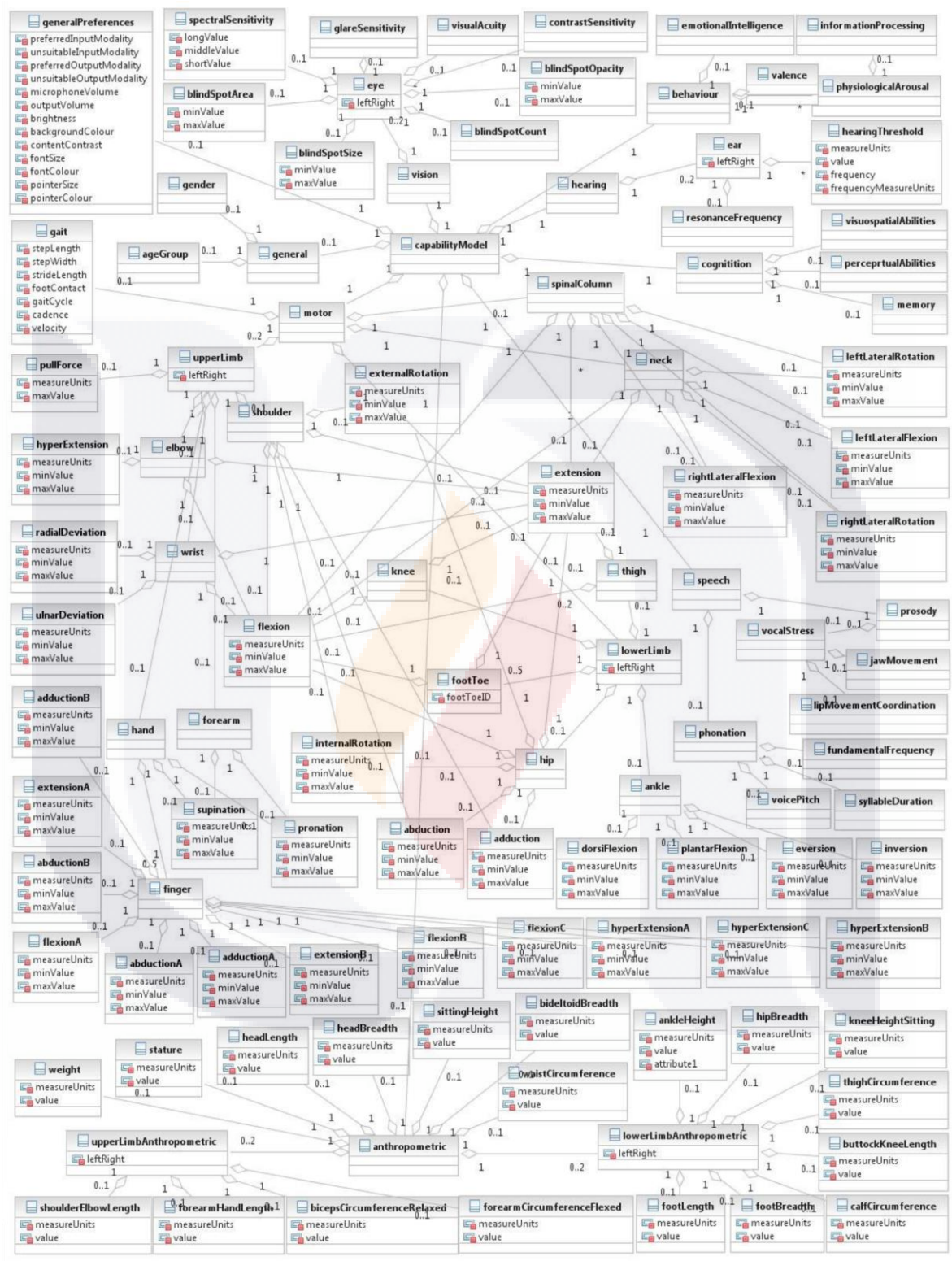


Figure 3.5 Full capability model describing physical, cognitive and behavioral user characteristics from [Kakl13]

For our case study, the disability and capability models correspond to the diagnosis that was realized by the specialist in speech therapy. The disability that was encountered is deep hypoacusia, and the affected tasks by it, are hearing and speaking. For the capability model, in the hearing branch, the patient presents hearing thresholds of more than 90 dB nHL for frequencies of 1000, 2,000 and 4,000Hz in both ears; and a resonance frequency between 800 and 1,200Hz. For the speech branch, the patient's voice was measured, getting as values for the pitch 250-300Hz, and 73dB for the maximum loudness.

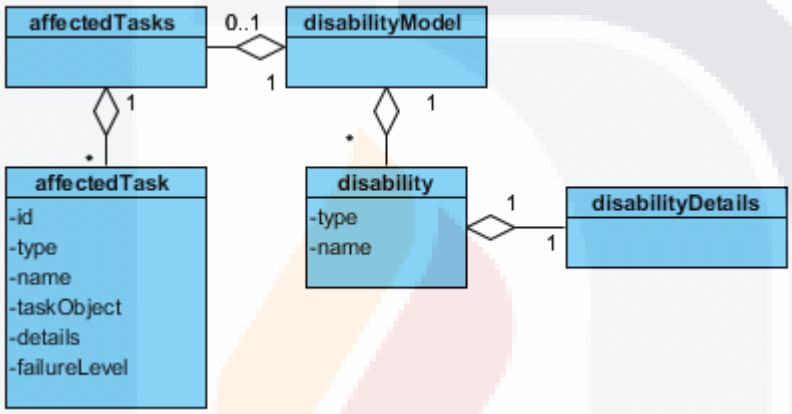


Figure 3.6 Disability model from [Kakl13] .

The prosody was also analyzed, obtaining for it typical results for a patient of the age and condition as the previously described. Corresponding to the identified disability, a rehabilitation program was designed comprising approximately 300 hours, to be provided in half an hour, two times a week sessions. One of the activities to be performed by the patient consists of repeating a series of phonemes in a determined order. The therapist mentions a phoneme and expects the patient to repeat it. If a phoneme is not correctly pronounced, the therapist reinforces its pronunciation and motivates the user to try again. The objective of the task is to improve the patient's pronunciation and to get him/her used to repeat the sounds that he/she listens to. For this particular activity, only a notebook and a color pencils are used.

3.3. The Dynamic Part of the Therapy Package

In order to represent the current progress and performance of the ***Patient*** towards the therapy program, and to provide the development team with a structure to understand how to evaluate patient's activities, the dynamic part of the therapy package, show in Figure 3.7, was created. This package contains then the ***Patient*** class, with an age attribute and related to a ***PerformedTherapy*** which reports the number of completed sessions and groups objects of type ***PerformedActivity***.

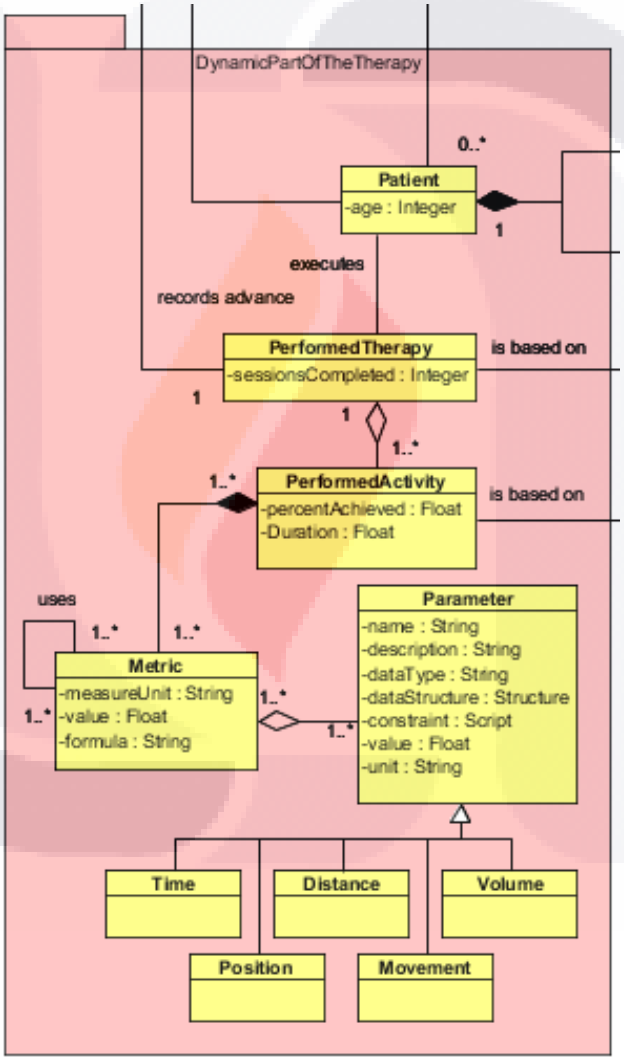


Figure 3.7 Dynamic part of the therapy package of the SEGA-ARM metamodel.

A ***PerformedActivity*** is based on a ***TheoreticalActivity*** and its main goal is to keep a record of the advance presented by the patient while performing a given activity. In order to allow the evaluation of a ***PerformedActivity***, a ***Metric*** class was also defined. Metrics allow the use of a formula in order to compute a value and to express it on a determined measure unit (e.g., centimeters, seconds and decibels). For the evaluation of a formula, a series of ***Parameters*** i.e., criteria to be used while evaluating, may be defined. There could be nested metrics since some formulas may need some others to be previously computed in order to take values from them. Metrics in this domain may be useful for measuring among others, tongue movement (i.e., tongue tip position, tongue vibration, and tongue position), lips movement (upper lip movement and lower lip movement), pronounced phonemes and blow action.

In the context of the case study, the performed therapy and performed activity classes keep track on how the patient is evolving and advancing through the therapy program and their involved values are constantly updated.

The metric to be used for the specific theoretical activity previously described, consists on evaluating how a phoneme is pronounced by the patient, recognize it and compare it to an expected phoneme passed as parameter according to a specific tolerance, using for this purpose the formula represented on Figure 3.8 where the pronounced by the patient phoneme (pp) is compared to the expected phoneme (ep) and then, if the distance between them is smaller than the tolerance value (t), the utterance from the patient is accepted as correct.

$$Recognition = distance(ep, pp) - t$$

Figure 3.8 Metric for evaluating if an utterance is recognized.

The process of definition of metrics may be supported by the creation of rehabilitation patterns that describe common activities. Examples of this type of patterns described by following the notation of design patterns mentioned in Section 2.4, are provided in Table 3.1 and Table 3.2.

Table 3.1 Facial praxia pattern description.

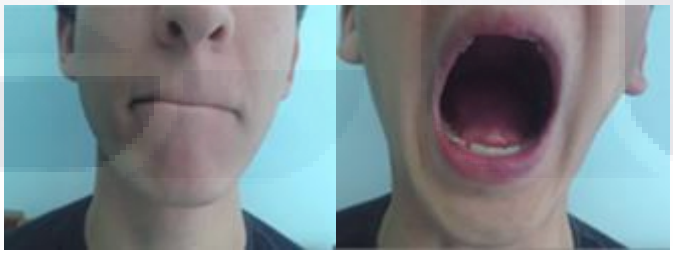
Pattern	Facial praxia
Context	Speech therapy session
Problem	The patient presents lack in facial mobility that is required for emitting sounds and words.
Solution	Muscular exercises are designed with which the patient may recover mobility and agility.
Purpose	Provide the patients with exercises for acquiring basic skills for emitting sounds and pronounce words.
Motivation	In the observed therapy sessions, different activities are carried out with the purpose of helping the patients to develop skills. One of these activities, consists on the repetition of muscle exercises called praxias, and pursues the goal of improving the patient's flexibility and ability for emitting sounds and pronouncing words.
Applicability	Use the facial praxia pattern for: - Proposing muscular exercises imitating the movements that are realized while emitting sounds. - Presenting a series of movements to a patient to allow him/her to develop skill on the disposition of facial muscles.
Examples	
Actors involved	Speech therapist, and patient
Consequences	Development of skills for articulation and emission of sounds and words.

Table 3.2 Blowing activity pattern description.

Pattern	Blowing activity
Context	Speech therapy session
Problem	The patient cannot successfully emit sounds and require exercises to help him/her to moderate the air emission and facial muscle disposition for this effect.
Solution	Exercises with which the patient may reinforce his/her capability to manage the air into his/her mouth, increasing the ability to emit sounds and pronounce words.
Purpose	That the patient develops the skill for modulating the air emission that is needed for phoneme pronunciation.
Motivation	Some phonemes are mainly differenced by the speed and the way in which the air is expelled from the mouth. This task may be controlled by exercising the disposition of facial muscles, and practicing how to inflate balloons, moving light balls, turn off candles, and blowing whistles.
Applicability	Use this pattern for: - Providing the user with an exercise to improve the management of the air that is expelled from his/her mouth. - Showing the user the differences when pronouncing phonemes.
Examples	 
Actors	Speech therapist, patient.
Consequences	Development of the skill of air emission and phoneme pronunciation.

3.4. The Serious Games Extension

In attention to the aim of this work, an extension, presented in Figure 3.9, to the therapy metamodel for considering serious games design elements was elaborated. For the proposal of this extension, the design elements introduced in Section 2.6 are taken into account. The intention on the definition of this extension is to provide developers with a tool for passing from a narrative description of a game to a semiformal structure that may be related to the one of the core metamodel. A ***SeriousGame*** is composed by ***Mechanics*** to record the player progress and to manage the punctuations, one or many instances of the ***Character*** class to define players as well as non-player characters (***NPC***), and ***SimulationGameElements*** such challenges, rules, scenarios and interactive objects.

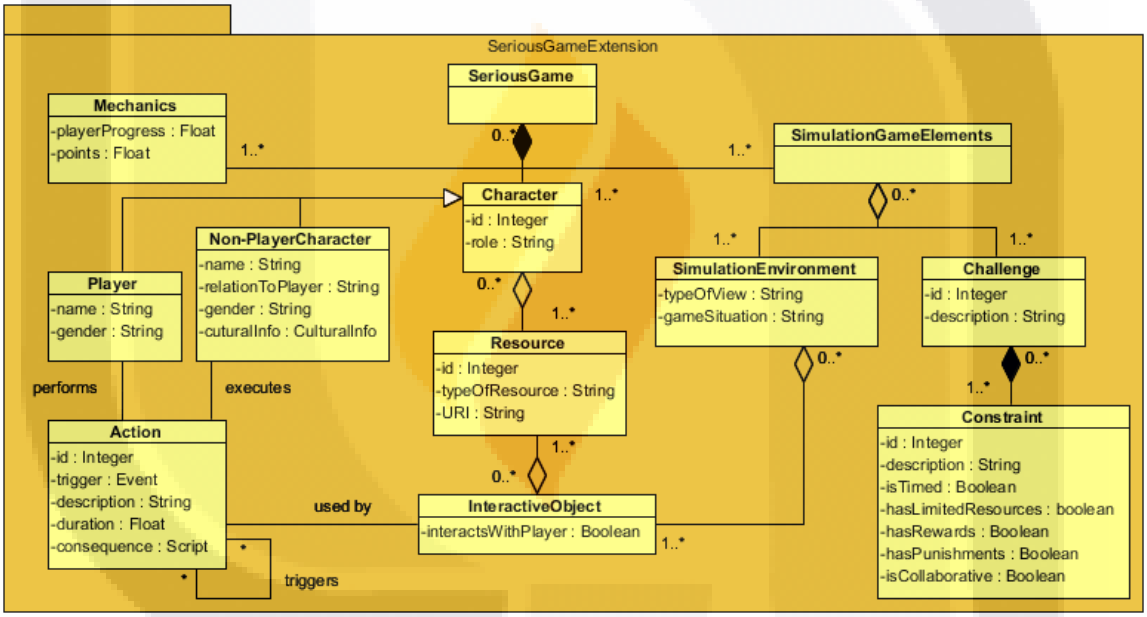


Figure 3.9 The serious game extension package of the SEGA-ARM metamodel.

A ***Character*** object represents an entity that realizes actions. These instances can be either a ***Player*** or a ***Non-PlayerCharacter***, and need ***Resources*** in order to be properly displayed. For the definition of ***NPC***s, it is necessary to give them a name, decide the relationship that they maintain towards the ***Player***, and when possible, provide their gender and cultural information e.g. language, nationality, degree among others [Long12]. In order to handle events on the game and describe their consequences, it was necessary to define the ***Action*** class. An ***Action*** is represented by an event (e.g. key press, click,

tap, recognized word or movement, or scripts) that triggers it, along with a description, duration, and a script to execute as consequence. Actions are performed by ***Players*** or executed by ***NPC***s and may whether use or not an ***InteractiveObject*** that are in turn, objects that belong to the ***SimulationEnvironment*** and are created to support the game interaction, for instance, the scenario, the floor, and colliding objects.

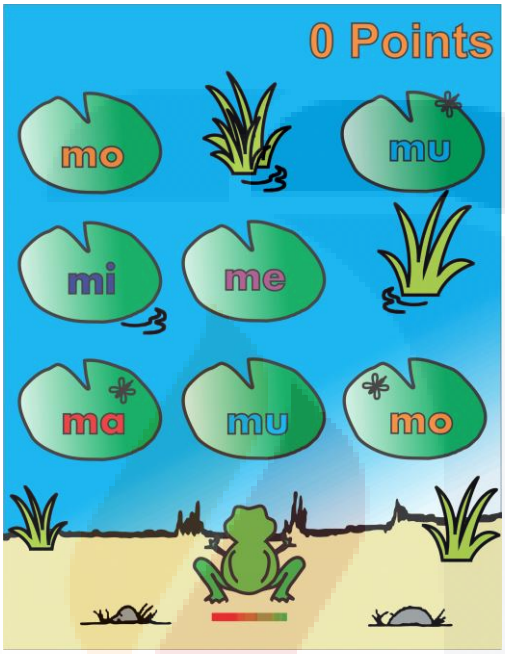


Figure 3.10 Result of the User Interface for the Froneme the Frog serious game.

The ***SimulationEnvironment*** description contemplates the definition of a type of view (e.g., first person, third person, or isometric) [Tayl02], and helps to hold the game situation i.e. level, state, and maximum amount of errors [Sale04] in a narrative manner. Finally for this package, a series of challenges has to be designed to represent the goals that are pursued while playing the game. A ***Challenge*** consists of a description and is composed by at least one ***Constraint***. Constraints may include one or several conditions (viz. time, limited resources, rewards, punishments, and mandatory collaboration).

For the given case study, considering the description provided by the speech therapist for the context, theoretical and dynamic packages, the serious game designer proposes to elaborate a game with the characteristics that are presented in Table 3.3 and which resultant user interface is shown in Figure 3.10.

Table 3.3 Characteristics of the designed serious game Froneme the Frog

Game name	Froneme the frog.
Narrative description	Froneme the frog is in the edge of a river looking for food to eat. In order to go find it, he has to jump between leaves avoiding to fall on the water. To perform a jump, the player has to identify the phoneme associated to the destination leaf and pronounce it. If the player is unable to provide a correct utterance, Froneme falls down into the water and therefore the game is over.
Player	The patient represented by a frog.
Interactive objects	Mosquitoes representing Froneme's food; leaves to allow Froneme jump from one to another identified by a phoneme; water that has to be avoided by Froneme.
Player actions	Eat mosquitoes by landing on the leaves that they are on, jump from one leaf to another by pronouncing phonemes, and fall down on the water if not correct utterance is provided on the given time interval.
Simulation environment	The game is designed with an isometric type of view, and the following game situations: Froneme in the border of the river (initial state of the game), Froneme on a free leaf, Froneme on a leaf occupied by a mosquito (eating), and Froneme falling on the water.
Challenges	The challenges involved in the game are the following: jump from a leaf to another pronouncing a correct phoneme, jump from a leaf to another within a determined time interval (time limit) to earn points (reward), or let Froneme to fall into the water (punishment); choose a leaf to jump on from a series of provided options (limited resources) and if there is a mosquito on the selected one, earn a bigger amount of points (reward).
Mechanics	For each time the player takes Froneme to a valid leaf, 10 points are going to be added to the budget, If there is a mosquito on the occupied leaf, ten extra points should be added. The time available for producing the phoneme is five seconds. Time and punctuations are recorded and reported.

It is important to point out that the definition of some of the game design elements was achieved through the establishment of relationships between the therapy packages and the serious games extension (e.g. ***TheoreticalActivity*** and the ***Action***). Next section emphasizes this characteristic of SEGA-ARM along with others that had not been yet addressed.

3.5. Discussion on the Metamodel

As it may be seen in Figure 3.1, there are relationships between the therapy packages and some of the serious games design elements extension classes. These connections allow the developers to know where to extract information that is supposed to be represented in the game, and how to present it. Punctually, there are relationships between: the ***Patient*** (Dynamic part of the therapy) and the ***Player*** (Serious games) to remark the fact that the patient has to be considered as the player and to get some information from his/her profile; the ***PerformedTherapy*** (Dynamic part of the therapy) and the ***Mechanics*** (Serious games), to track the advance of the player through the game and give a feedback for the therapist; and the ***TheoreticalActivity*** (Theoretical part of the therapy) and the ***Action*** (Serious games) in order to understand the tasks that the patient is supposed to perform while playing the serious game. It is also important to point out that three levels were considered while modeling the therapy: (1) the patient (capabilities and disabilities), (2) the therapy program (theoretical therapy), and (3) the involved activities to meet the plan (theoretical activity). Each one of those three levels has equivalence on the dynamic part of the therapy in order to keep track of the advance and performance of the patient through the therapy.

When compared to the proposal in [Long12], this metamodel has the advantage of providing a semiformal notation which may be used in a methodological process as a conceptual guide for achieving the goal of expressing and therefore implementing serious games for a specific set of therapy activities. Along with its description, it also allows identifying the actor that is supposed to fulfill the different classes and attributes, and provides a definition of the relationships that exist among them. However, in this paper the use of knowledge bases and taxonomies is not considered resulting on the lack of a repository of activities to provide to the developers.

One of the aimed advantages of this proposal is reutilization, and in order to demonstrate it, a second instantiation of serious game, called Roberto the robot, was realized in which

the very same definitions of the described case study for the context and for the static and dynamic parts of the therapy packages are taken into account, varying only the serious game extension elements and therefore creating a different serious game but for the same rehabilitation purpose.

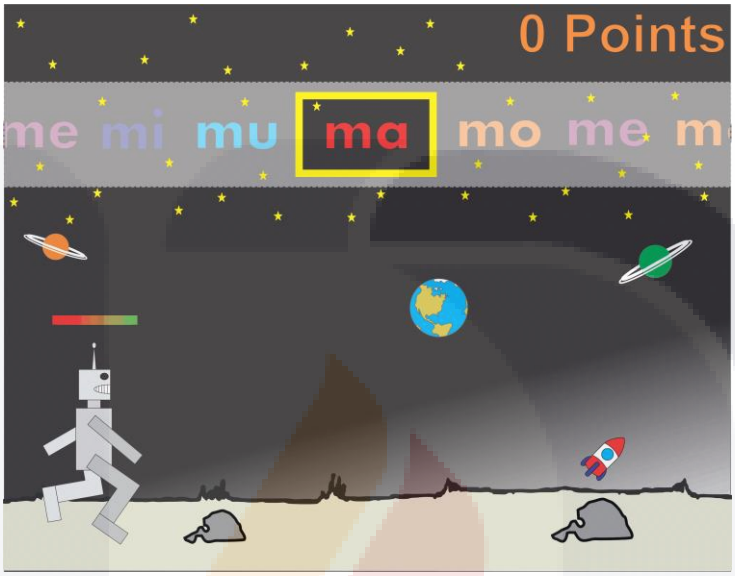


Figure 3.11 User Interface of the Roberto the Robot serious game.

For this second game, a character representing the player called Roberto the robot is set on a scenario representing the moon's surface looking for pieces to assemble a rocket to return to the Earth. In order to find them, he has to run and jump over rocks avoiding hitting them. To perform a jump, the player has to pronounce a required phoneme. If the player is unable to provide a correct utterance, Roberto hits a rock and finally after 5 impacts, falls down.

The interactive objects are: small rockets representing rocket pieces, and rocks that have to be avoided by the player. The player actions are: to take rocket pieces by reaching them, to jump over rocks, to hit rocks if no correct utterance is provided on a given interval. The game was designed with a third person type of view, and the following game situations: Roberto on the surface of the moon, Roberto jumping over a rock, and Roberto reaching a rocket piece.

The challenges involved in the game are the following: jump over rocks pronouncing a correct phoneme (limited resources) within a determined time interval (time limit) to earn points (reward), or hit rocks and fall (punishment); and if a rocket piece is reached earn a

bigger amount of points (reward). As mechanics of the game, for each time Roberto jumps a rock, 10 points are added to the budget, if a rocket piece is reached, ten extra points are added. The time available for producing the phoneme is five seconds. On the development of this game, shown in Figure 3.11, also interface elements from the Froneme the Frog game were reused.

It is important to note as another advantage of the here proposed metamodel that since the packages on it included keep a certain level of independence towards the serious games extension, it is possible to use them, and to define new extensions in order to design other types of software solutions for auditory rehabilitation (e.g., diagnosis, tracking, and advising systems). Likewise, by developing models for other domains than auditory rehabilitation such as educational and military, it may be possible to relate them to the extension here provided to enable the development of serious games for those ambits.

4. A Process for Development of Serious Games to Support Auditory Rehabilitation

The result of the revision of the available literature, along with the experience obtained by attending speech therapy sessions, lead to abstract and structure characteristics to develop applications to support the rehabilitation process. After developing 3 applications in collaboration with degree students from different universities in Mexico, and reporting the process that was followed for their production [Cesp14a] [Cesp14b] [Cesp14c], the characteristics of these processes were considered and enriched with aspects of rehabilitation, serious games design, and HCI, as established in the Introduction in order to propose an enrichment of a process aimed on the production of serious games to support auditory rehabilitation.

A software process definition may include the description of activities, roles, workflow, internal and external products, methodologies, tools and metrics for each stage or for the whole thing. The aim of this work is to define for the whole process: a workflow model, a software development life cycle model, roles, and products; while for the stages on it involved, to define: tools, products, documents, and roles.

The remainder of this chapter is organized as follows: Section 4.1 introduces the proposed process, explaining its foundations, the additions that were made to it, a workflow that may be followed, the stages that are on it involved, and the expected products to be generated. Sections 4.2 through 4.5 describe in detail the stages that are involved in the proposal, by exposing their internal products, as well as the tools and/or methodologies to be used to achieve the expected goals. Due to the nature of the here proposed process, Section 4.6 is focused on integration of the generated components and maintenance of the products. Finally, in Section 4.7 a discussion takes place in terms of the characteristics of the proposal, and its advantages and disadvantages.

4.1. General View of the Process

For the proposal, the characteristics of a component based process model [Press05] [Ning96] were took as base and enriched with game design elements, HCI aspects and rehabilitation concepts in order to support the development of serious games to assist speech therapy programs. The main reason for choosing this process model as template

is that it promotes reutilization of generated components, making it easier to develop applications lowering the required effort. In this sense, for small projects, less human resources are needed and time invested for developing decreases. Meanwhile, a bigger effort is required while analyzing requirements, organizing a repository of components and in configuration and integration tasks.

Also, the component based process model, offers UML compliance which makes it easier to develop applications considering aspects that had been already set up in this notation and that were presented in Chapter 3. Finally, this process model, as most of the others, includes stages: requirements analysis, design, codification, and testing.

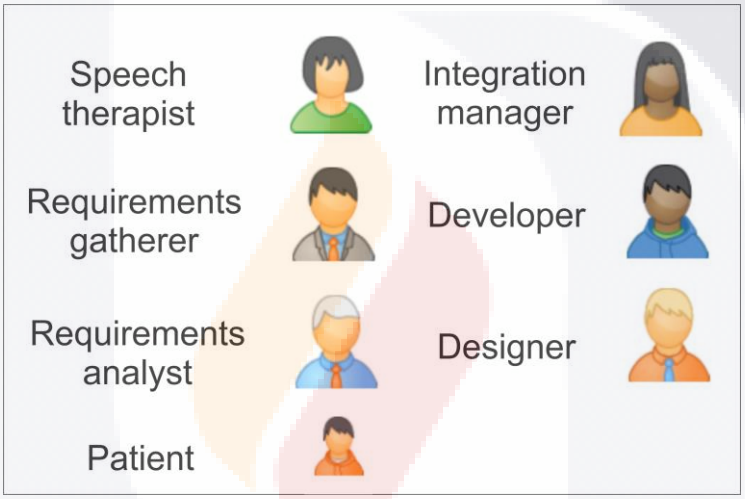


Figure 4.1 Identified roles involved in the proposed process model.

The proposal is first presented from a workflow approach. To this effect, it is important to remember that the proposed process is aimed to support a rehabilitation program, and therefore, it begins from the observation of therapy sessions, in which characteristics, flow and tasks of certain activities are presented to a resource in the development team who is playing the role of *requirements gatherer*. Next, through an interview with the *speech therapist*, the observation is complemented, the provided instruments to document the context of usage and the static and dynamic part of the therapy are filled and functional and nonfunctional requirements are gathered and detailed. Later, the *requirement analyst*, processes the information, an *integration manager* evaluates if it is possible to execute integration from components on a repository, and then, a requirements analysis document is elaborated.

Once the *speech therapist* approves the requirements definition, it is possible to start a design phase. In this stage, a *designer* takes the requirements document, along with the filled instruments, and if needed with the documentation of the to be integrated modules, and model a serious game, taking as reference the provided game elements list, and the serious game extension from SEGA-ARM (Chapter 3), and generates a new document, containing at least a conceptual model, user interface prototypes, and a requirements traceability matrix. This document presented to the *speech therapist* for his/her approval.

With the design document released, a *developer* must perform codification, but before it, another document should be drafted, including at least a codification standard, description of the integration from existent components, and once more a requirement traceability matrix. In this stage, metrics for evaluation of the progress of the patient and to gather statistics of the usage of the applications are included.

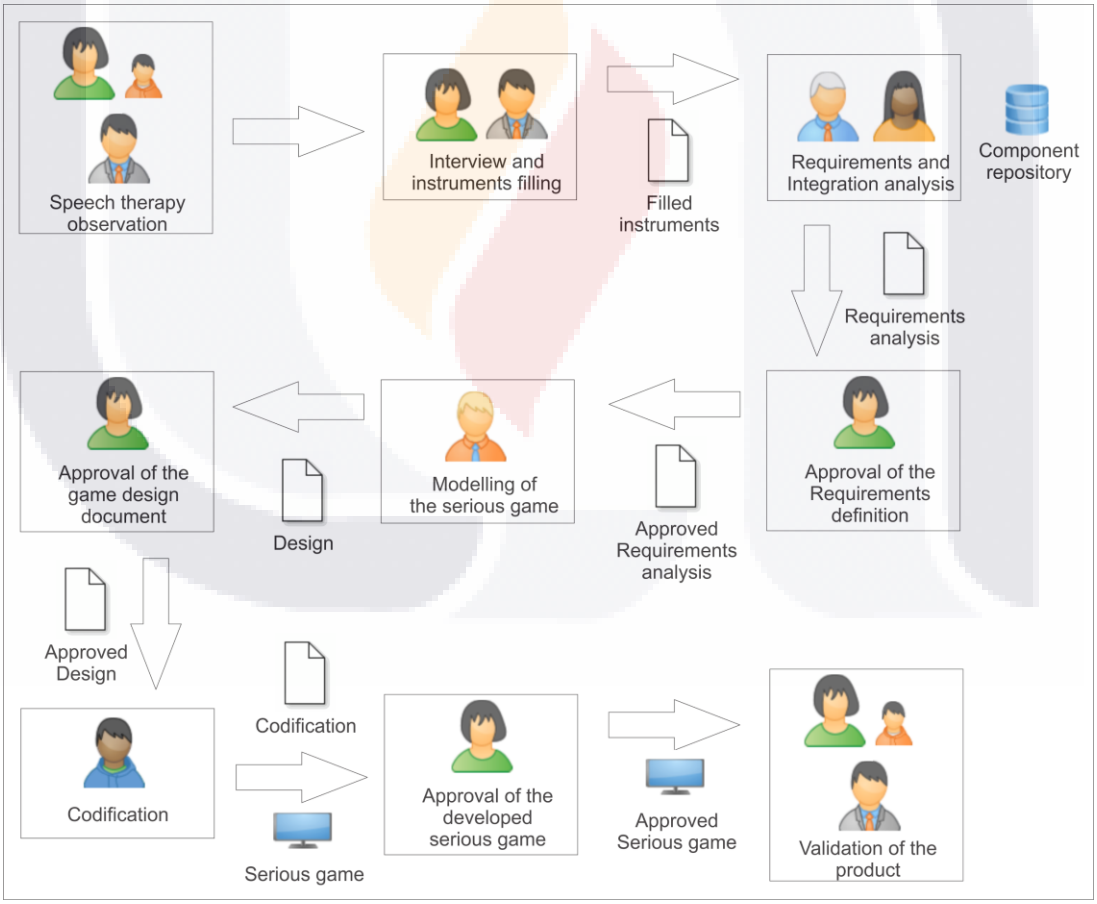


Figure 4.2 Workflow model of the implementation of the proposed process model.

Finally, the *speech therapist* approves the implementation and agrees to use it as part of the therapy sessions; tests may be performed in terms of usability, user experience, and performance. Also, once the *patient* start using the provided game, statistical data may be gathered to validate it. It is important to point out that it is necessary that each one of the specified roles exist, but one person may play two or more of them. The identified roles that take place on the application of the process are presented in Figure 4.1 along with a graphic representation of them to be used in the illustration of the workflow of the proposed process that is shown in Figure 4.2.

From the workflow description, it is possible to identify the existence of at least four stages: requirements gathering and analysis, design, codification, and testing; that are reinforced with integration and maintenance activities. Also, it is possible to find products that are supposed to pass through stages and the actors that are on them involved. These products, and actors, along with specific tasks to be performed on each stage, are described in Sections 4.2 through 4.5.

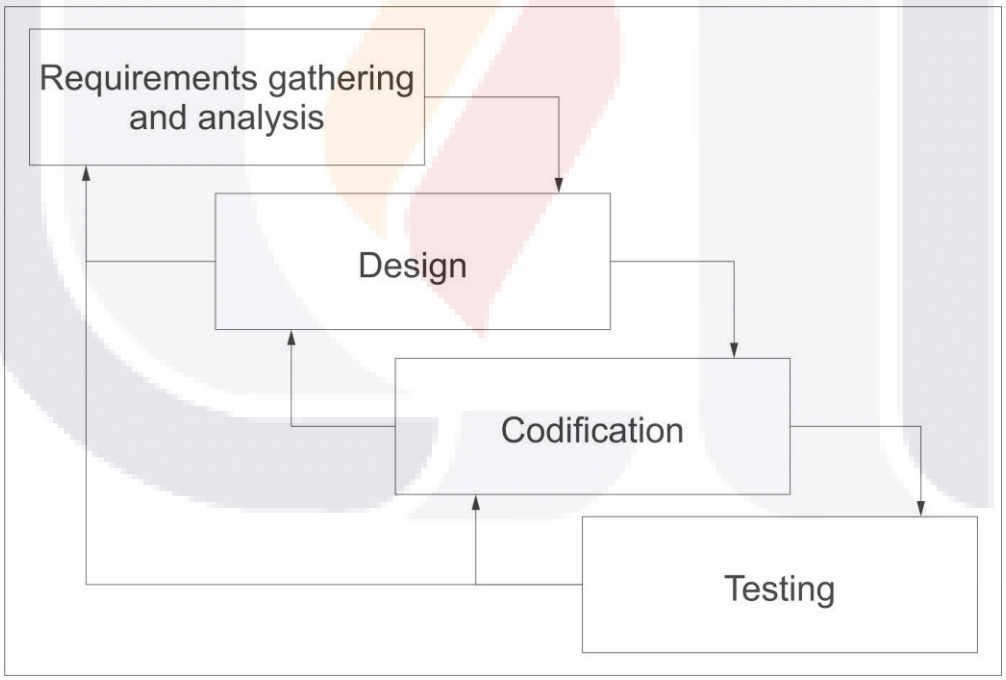


Figure 4.3 Structure of the process model to promote the development of serious games to support auditory rehabilitation.

In the workflow model, it was also mentioned that for advancing through the project, several speech therapist approvals are required, making it necessary to consider that once

an stage is being performed, it may be possible to return to others in order to correct or rework elaborated products. The structure of a process including and relating these stages is shown in Figure 4.3.

4.2. Requirements Gathering and Analysis Stage

The objective that is pursued with the definition of this stage, is to provide the development team with a phase to understand the rehabilitation domain for which the project is being carried out, and to collect characteristics of the to be developed serious game, by getting involved on therapy sessions, and by asking the speech therapist about his/her needs.

For allowing the accomplishment of these objectives, it is necessary to involve several human resources and to define the tasks that ought to be carried out by them:

- The patient, who has to perform therapy activities as usually, does for his/her rehabilitation program.
- The therapist, to participate in the to be observed therapy sessions, to participate in the interview and define the requirements for the project, to help filling the instruments for analyzing the context, the static part of the therapy, and the dynamic part of the therapy, to evaluate and approve the generated document, and therefore to enable the pass to the next stage.
- The integration manager, to analyze the gathered requirements, and to propose, if there are, components from a repository that may be reutilized in the project.
- The requirements gatherer, to observe the therapy sessions driven in the therapist's office, to perform the interview, to fill the instruments for analyzing the context, the static part of the therapy, and the dynamic part of the therapy, to deliver a list of functional and nonfunctional requirements.
- The requirements analyst, to review the filled instruments, and the functional and nonfunctional requirements, organize the information, the integration proposal, and to elaborate the requirements analysis document.

This stage, as may be seen in the tasks to be performed by the therapist and the requirements gatherer, involves the usage of instruments for analyzing the context, the static part of the therapy, and the dynamic part of the therapy. These instruments, which are presented in Appendix C, were elaborated in compliance to SEGA-ARM (Chapter 3), to work as a guide and to allow organization of concepts that are essential, for later carrying out the design phase.

The internal products to be created on this stage are: report of the observation, report of the interview, matrix of the identified functional and nonfunctional requirements, filled SEGA-ARM instruments, proposal of integration of components, and speech therapist approval. While on the other hand, the only external product is the requirements analysis document.

4.3. Design Stage

This stage is to consider the approved requirements analysis document, and model the serious game to be developed, in terms of a conceptual model (e.g. classes' diagram), user interfaces prototypes, and according to a game design elements list and the SEGA-ARM serious game extension.

This phase supposes the need of actors performing specific activities as follows:

- The Designer: to review the requirements analysis document, to elaborate a conceptual model for expressing the components to be developed, and the application in general, to model the inclusion of the to be integrated components, to design the serious game in compliance to the SEGA-ARM serious game extension, to design user interfaces for the game, to update the requirements traceability matrix, and to elaborate the design document concentrating all that is created and updated during this phase.
- The therapist: to review, and to approve the design document.

For achieving the objectives of this stage, a SEGA-ARM compliant instrument for expressing the serious game design elements, which is presented in Appendix C, is provided.

The internal products to be elaborated as part of this stage are: the requirements traceability matrix updated, and the prototypes of user interfaces. The only external product that is produced in this phase is the approved design document.

4.4. Codification Stage

This stage is one of the most important of the process, since on it the main objective is to translate the elaborated models, documents, and prototypes into code for a serious game to support auditory rehabilitation. Other goals involved in this phase are: to integrate the already existent components, to update the requirements traceability matrix, to elaborate

the codification document including a codification standard, and to submit the serious game for the speech therapist approval.

The actors that are involved in this stage and their tasks are described next:

- The developer: to review and analyze the design document, to review and analyze the integration proposal, to carry out the codification of the serious game according to the documentation, to define a codification standard, to update the requirements traceability matrix, and to elaborate the codification document.
- The therapist: to approve the developed serious game, and to evaluate if the application satisfies his/her necessities towards the therapy program in which it is going to be included.
- The integration manager: to supervise the integration of existing components according to the integration proposal, and to document and add the developed components to a repository.

The internal products to be elaborated are: the codification standard, the updated requirements traceability matrix, and the documentation for describing the components that are going to be added to the repository. The external products are in turn: the codification document, and the developed serious game to support auditory rehabilitation.

4.5. Testing Stage

This stage has as objectives: to implement the developed serious game as part of a therapy program, to design and apply an evaluation to determine if the product satisfies correctly the needs of the user, to get improvement opportunities, and to gather statistical data from the usage of the application for each user in order to determine how much does the application impact on the performed therapy. This stage may imply, but is not limited to, usability testing, user experience experiments, quality evaluation, and functional testing.

The enrolled actors and their activities are listed next:

- Patient: to be instructed in the usage of the application, to use the developed serious game, to provide feedback (which may result in usability evaluation through the think aloud technique [Niel02]), and to participate in user experience and usability experiments.
- Speech therapist: to instruct the patient in the usage of the serious game, to include the developed serious game as part of the rehabilitation program, to

participate in user experience and usability experiments, and to provide feedback (that may result on an usability evaluation through the expert opinion technique [Mayh99]).

- Requirements gatherer: to participate in the patient instruction in the usage of the serious game, to gather statistical data from the usage of the application (e.g. elapsed time, and achieved punctuation), and to report any found issues during the realization of this stage, to elaborate if necessary, change requests.

During this stage, the internal products that are to be created are: experiment designs, issues reports, feedback from the user and from the speech therapist, and statistics of the usage of the serious game; while the external products are change requests, and an analysis of the statistical data that was gathered from the usage of the serious game.

4.6. Integration and Maintenance on the Process

For the approach of the proposal is a component based one, it is required to consider integration and maintenance tasks into it. Integration in this case, is addressed by defining the usage of a repository of documented software components, analyzing the pertinence of their inclusion in a current project, and performing its addition to the code; and by assuring that every component created in the current work, is well documented, tested, and added to the repository, in order to make it available for its implementation in further affairs. This integration must be supervised by a member of the development team with the role of integration manager.

From the structure of the general view of the proposal, presented in Section 4.1, it is possible to see that it follows an iterative approach. This means, that once the project is developed, according to the results of tests and experiments, analysis tasks must be carried out, in order to identify issues that may be attended in further iterations to improve the performance of the application, and enhance the interaction with the user.

Moreover, maintenance also includes the support that may be provided to the speech therapist, as well as to the user to instruct them on the operation of the developed serious game, and to gather and report statistical information about its usage. These tasks are mainly accomplished by the requirements gatherer.

4.7. Discussion on the Process

The proposal of this work as it may be seen in the rest of this Chapter is an enriched software process model to develop serious games to support auditory rehabilitation. The concepts that are added to it are related to rehabilitation, game design elements, and HCI, as it was intended to accomplish the objectives that were defined in the Introduction.

This process is designed in compliance to a component based approach, and an iterative paradigm. For its application, 7 roles had been identified and described, from which, 5 must be carried out by the development team (formed by 1 or more persons), meanwhile the two others correspond to the patient (user), and the speech therapist. For each one of these roles, activities had been settled, focusing mainly in describing what to do, but not on how to do it, safe those that are related to the usage of the instruments to express the context, the static part of the therapy, the dynamic part of the therapy; as well as the ones about game design, that may be supported by the serious games extension, and the list of components to be considered for the design of games. The found activities grouped by the actor that is supposed to perform them, are shown in Table 4.1.

Table 4.1 Actors involved in the proposed process and their responsibilities.

Actors	Responsibilities
Patient	- To perform therapy activities as usually does for his/her rehabilitation program. - To be instructed in the usage of the application. - To use the developed serious game. - To provide feedback. - To participate in user experience and usability experiments.
Speech therapist	- To participate in the to be observed therapy sessions. - To participate in the interview and define the requirements for the project. - To help filling the instruments for analyzing the context, the static part of the therapy, and the dynamic part of the therapy. - To evaluate and approve the requirements document, and therefore to enable the pass to the design stage. - To review, and to approve the design document. - To approve the developed serious game,

	- To evaluate if the application satisfies his/her necessities towards the therapy program in which it is going to be included. - To instruct the patient in the usage of the serious game. - To include the developed serious game as part of the rehabilitation program. - To participate in user experience and usability experiments. - To provide feedback.
Requirements gatherer	- To observe the therapy sessions driven in the therapist's office. - To perform the interview, to fill the instruments for analyzing the context, the static part of the therapy, and the dynamic part of the therapy. - To deliver a list of functional and nonfunctional requirements. - To participate in the patient instruction in the usage of the serious game. - To gather statistical data from the usage of the application (e.g. elapsed time, and achieved punctuation). - To report any found issues during the realization of the testing stage. - To elaborate if necessary, change requests.
Requirements analyst	- To review the filled instruments, and the functional and nonfunctional requirements. - To organize the information, and the integration proposal. - To elaborate the requirements analysis document.
Integration manager	- To analyze the gathered requirements. - To propose, if there are, components from a repository that may be reutilized in the project. - To supervise the integration of existing components according to the integration proposal. - To document and add the developed components to a repository.
Designer	- To review the requirements analysis document. - To elaborate a conceptual model for expressing the components to be developed, and the application in general. - To model the inclusion of the to be integrated components. - To design the serious game in compliance to the SEGA-ARM serious game extension. - To design user interfaces for the game.

	- To update the requirements traceability matrix. - To elaborate the design document concentrating all that is created and updated during the design phase.
Developer	- To review and analyze the design document. - To review and analyze the integration proposal. - To carry out the codification of the serious game according to the documentation. - To define a codification standard - To update the requirements traceability matrix. - To elaborate the codification document.

Along with the activities and actors that take part in the proposal, a series of internal and external products are defined. Except for those that consider the usage of the SEGA-ARM model and instruments, they are only identified, but it is out of the aim of this work to describe in detail how to carry them out. The items that are to be produced are presented in Table 4.2, grouped by each one of the stages that are considered as part of the proposed process.

Table 4.2 Internal and external products per stage.

Stage	Products
Requirements gathering and analysis	- Report of the observation. - Report of the interview - Matrix of the identified functional and nonfunctional requirements. - Filled SEGA-ARM instruments. - Proposal of integration of components. - Speech therapist approval of the requirements analysis document. - Requirements analysis document.
Design	- The requirements traceability matrix updated. - The prototypes of user interfaces. - Speech therapist approval of the design document. - The approved design document.
Codification	- The codification standard.

	- The updated requirements traceability matrix. - The documentation for describing the components that are going to be added to the repository. - The codification document. - The developed serious game to support auditory rehabilitation.
Testing	- Experiment designs. - Issues reports. - Feedback from the user and from the speech therapist. - Statistics of the usage of the serious game. - Change requests. - Analysis of the statistical data that was gathered from the usage of the serious game.

Once the process has been described along with the activities, roles, and products on it involved, it is necessary verifying that it successfully accomplish the goals that are introduced in the first chapter of this thesis work, and to validate products developed under its application. Next chapter is dedicated to this effect.

5. Verification and Validation

In regard to the objectives that were defined in the Introduction, the present chapter is aimed to present the evaluation that was carried out over the proposed process and one of the products that was developed following it. The rest of the chapter is divided into sections, the first one for presenting the verification of the process, the second one for describing the validation of the product, and the last one for discussing the obtained results.

5.1. Verification of the Process

In order to verify the correctness of the proposed process, an experiment was designed, in which four development teams, each developing one project, were evaluated. To perform the evaluation, two of the teams were given with instruments for the analysis of: the context, the static part of the therapy, and the dynamic part of the therapy; as well as with a list of activities to carry out, a list of game design elements to be taken into account, a workflow model to be followed, and the structure of the software process, proposed in chapter 3, while the two others were provided with generic software engineering material (books, formats, papers, and references).

Also, each team was advised by a speech therapist for the definition of game requirements and had the following objectives: Develop a serious game for supporting auditory rehabilitation, according to what the speech therapist requested; elaborate documentation expressing requirements identification, design, and codification; and to use a mechanism to evaluate the user progress while using the game.

A series of metrics was defined in order to measure and allow comparison of the performed projects. The first one is about how many, and which documents are generated during the process. This metric is to serve as an evidence of the application of every carried out process stage and to demonstrate that internal and external products were generated. The elaborated documents for the applications that were developed following the here proposed process, are included as Appendix A and Appendix B. It is important to point out that the documentation of all of the projects was carried out in Spanish, since it is the mother language of the participants.

The second metric that was applied, is the time that was invested for the project, since it may be a relevant factor to point out, because in conjunction with the next one, determines the applied effort for developing the product (invested time * team size) [Gepp05].

In third place, the team size was considered, only to point out that for this experiment, all of the teams were of the same size (2 people), and therefore, there wasn't influence of the applied human resources on the result, and remained constant, making it unnecessary to calculate the effort. As a qualitative metric, the profile of the development team members is considered in terms of academic background, in order to show that it was not a determining factor for the success of a project.

Finally, it was counted how many requirements were identified, how many of them were accomplished, and then, the rate of completeness of the application was calculated as the accomplished requirements over the identified ones. Table 5.1 presents the obtained results for each of the considered projects on terms of the applied metrics, as well as final observations describing their characteristics.

Table 5.1 Comparison of applications that were implemented following the proposed process and others without following it.

Criteria	Museum visit (Without process)	Sound discrimination (Without process)	Safari hunter (With process)	Froneme the frog (With process)
Generated documents	● Partial delivery of a Software requirements document	● Requirements description ● Tasks models ● Abstract User Interfaces models ● Concrete User Interfaces ● Final User Interfaces	● Software requirements definition document ● Design document ● Codification document ● Context format ● Static therapy format ● Dynamic therapy format	● Software requirements definition document ● Design document ● Codification document ● Context format ● Static therapy format ● Dynamic therapy format

			● Games design elements description	● Games design elements description
Time invested	1 month	1 months	1 month	1 month
Team size	2	2	2	2
Team members profile	● Student of the last semester of Computational systems engineering. ● Speech therapist.	● Student of the fifth semester of Computational systems engineering. ● Speech therapist.	● Student of the fifth semester of mechatronics engineering. ● Speech therapist.	● Software developer. ● Speech therapist.
Found requirements	6	6	14	12
Requirements accomplished	0	4	14	12
Completeness (Accomplished /Found)	0%	67%	100%	100%
Observations	Not successfully developed product, unable to be presented to the speech therapist for evaluation.	Mobile app well enough developed with sound usage, submitted to therapist approval, and pending to be maintained according to comments, in order to be tested	Mobile serious game successfully implemented with speech recognition enabled, submitted to speech therapist approval and pending to be uploaded to Google Play store.	Web oriented serious game successfully developed with speech recognition usage, uploaded to a private server and used by patients as complement of their

		by patients as complement to their therapy program.		rehabilitation process.

The following section presents the validation that was carried out on one of the products that were developed following the proposed process, while section 5.3 contains the discussion about the results that were obtained on these processes of verification and validation.

5.2. Validation of a Product

For validating the proposal of this work, one of the products that were developed following the process and using the provided tools was considered. The main reason for its selection is because it was used by actual auditory rehabilitation patients as part of their therapy program, and data of this interaction was gathered. The application that was considered for this effect, was entitled “Froneme the Frog” and its documentation may be found in Appendix B. The experiment that was driven is described in the following paragraphs.

As the last steps of the workflow model suggest, once the application was developed, it was presented to the speech therapist to request for approval. When that approval is granted, the application is ready to be available for the users. In this case, the “Froneme the Frog” application was implemented for being used in a web context, using a web browser, so its address was given to 4 of the patients that were enrolled to speech therapy at the *Clínica de Atención Integral a la Salud* of the Universidad Autónoma de Aguascalientes, two girls and two boys in the ages between 7 and 8 years. And their performance on the application during 2 weeks was tracked.

Aligned to the objectives that were presented in the introduction chapter, the concepts to be measured are: number of sessions, points obtained per session, words pronounced per session, time elapsed per session, points over time ratio, and words per minute. The results for each user, were gathered in a database, and later were graphed for their analysis.

For the first user, it was reported that 12 playing sessions were performed, adding 0.82 hours to his therapy, increasing then it in a 41%. During the whole experience, the patient

pronounced a total of 195 words in 2972 seconds, having as best turn in terms of total utterances 33 words, while in terms of words per minute rate, one were 31 words were pronounced in 5.25 minutes.

Figure 5.1 Statistical data gathered from the first patient.

Also, it is important to notice that for this patient, the trend was to increase punctuation, elapsed time, and quantity of words, denoting improvement in the performance of the player. Figure 5.1 shows the obtained results for the first patient.

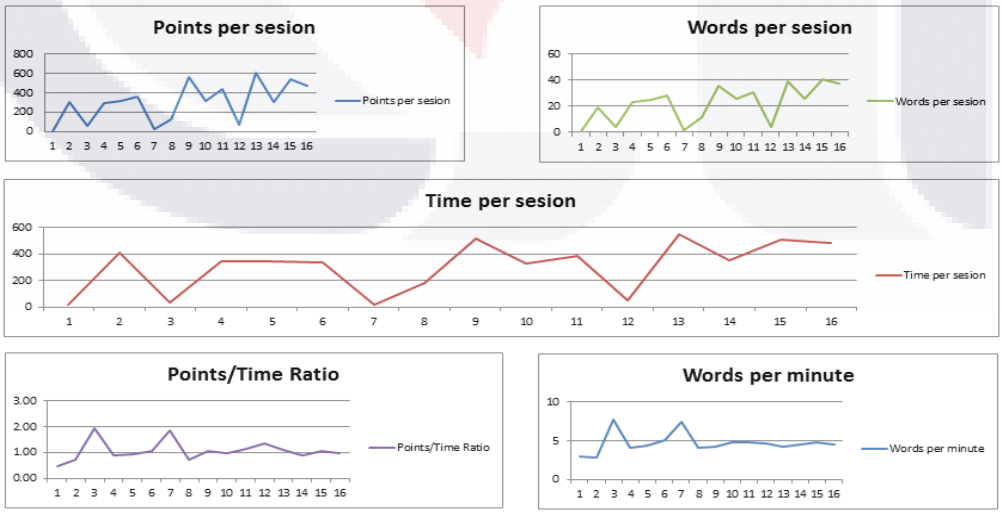


Figure 5.2 Statistical data gathered from the second patient.

The second patient used the application 16 times, adding to her therapy a total of 1.35 hours, meaning by this an increment of 67% in regard to her regular treatment. During the two weeks of the experiment, she pronounced correctly a total of 354 words. In terms of elapsed time, her best result was a session in which 547 seconds were invested, while in terms of pronounced words, the best session reported that 41 words were uttered. The graphed results for this user are presented in Figure 5.2, on them, it is also possible to see that the performance of this user kept a regular trend, this is, that the results of points over time ratio and words per minute, do not present dramatic decrements or increments.

The third user in turn, carried out 20 play sessions, by which 1.81 hours were added to the therapy, representing an increment of 90% with respect to her regular therapy program. In terms of invested time, the best result was one in which 572 seconds were employed, while with respect to number of words uttered, it was another one in which 41 words were pronounced. This last result was also the best one in regard to the words per minute rate, with a value of 4.44. Figure 5.3 shows the graphic results for this patient, were it is possible to notice that the results were slowly improving.

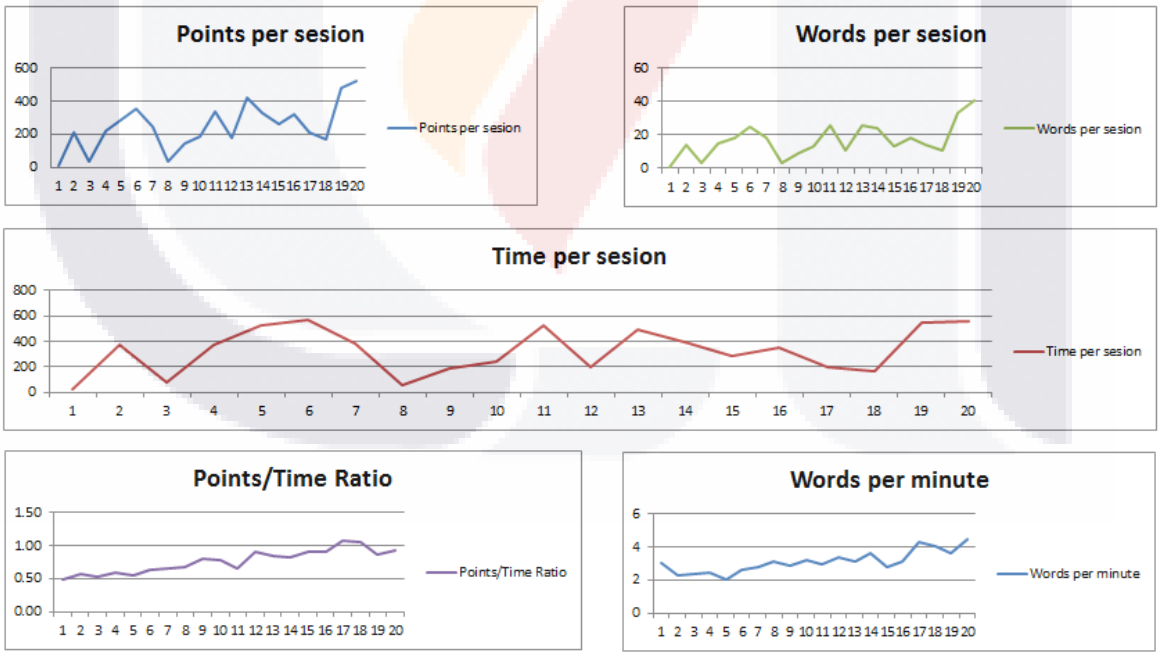


Figure 5.3 Statistical data gathered from the third patient.

For the fourth patient, results from 17 sessions were gathered. This experience meant an addition of 1.53 hours his therapy, representing an increment of 75% with respect to the regular therapy program. In regard to the number of uttered words, the best result consisted on a total of 58 utterances, while in terms of invested time, the best result was another in which 578 seconds were employed. Moreover, when it comes to the words per minute rate, the best result was a value of 6.15. Figure 5.4 shows the graphic results for this patient, were it is possible to notice that the performance of this user kept a regular trend, this is, that the results of points over time ratio and words per minute, do not present dramatic decrements nor increments.

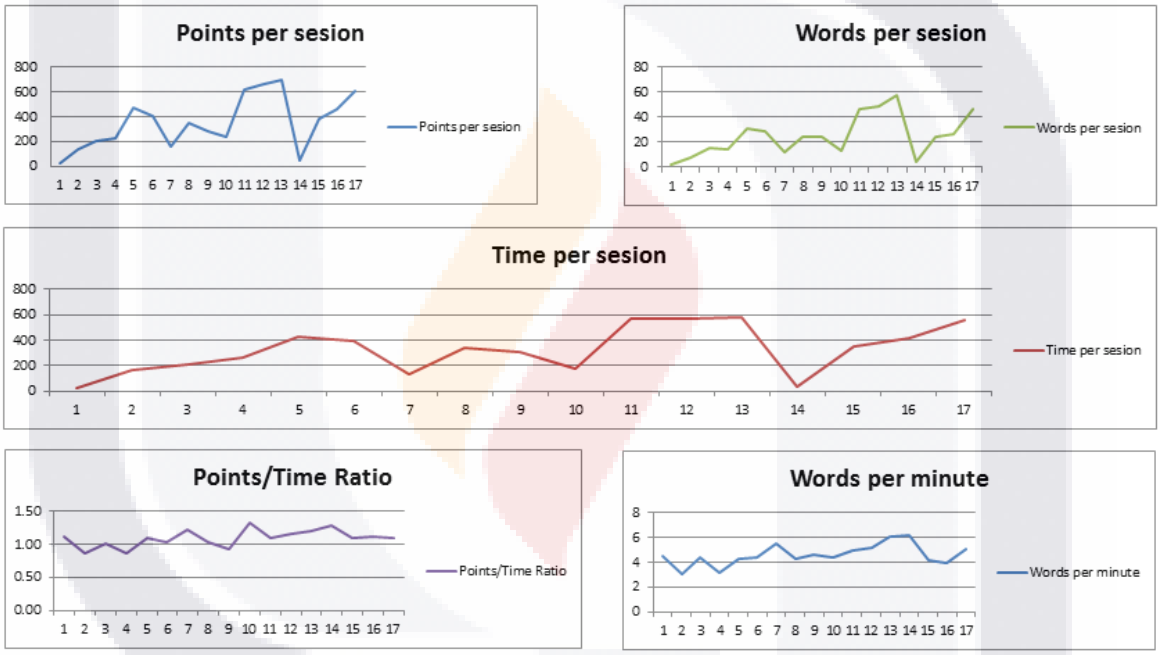


Figure 5.4 Statistical data gathered from the fourth patient.

The following section presents a discussion about the results that were reported for the verification and validation that were carried out.

5.3. Discussion on Verification and Validation

The verification that was carried out over the process, reported that more documentation was generated in the projects that followed the proposed process than those that did not. Some of these documents such as the instruments for describing the context, the static part of the therapy and the dynamic part of the therapy, allow reutilization of the products

that are generated in the early stages of the process (e.g. requirements, models, and designed components). Also, it was possible to note that the applied effort (relation of the human resources and the time invested) for the development of the projects was uniform, meaning by this that the advantages encountered on the projects that were created with or without process are not affected by this factor.

The characteristics of the teams that participated, were distributed so that the experience of the developers wouldn't impact in the success of the projects, even, one of the persons that acted as software engineer / developer on a project that followed the proposed process, was not closely related to those disciplines, and accused to have little experience in development, while the others had average to good programming skills and were aware of software engineering practices.

In regard to project completeness, both of the projects that were elaborated by following the proposed process got to be developed in a 100%, when comparing the reached requirements and those that were gathered; meanwhile for the other projects, one of them reached a 67% of accomplished requirements, and the remaining hadn't managed to be implemented. Finally, one of the projects (developed following the process) was reviewed and authorized to be used as complement of therapy programs; two others (one with and one without process) were checked and fed back by the speech therapist and are actually pending to be included as part of therapy programs; lastly, one of the projects (implemented without following the proposed process) was not successfully developed and later abandoned.

On the other hand, for the teams that followed the process, it was necessary to explain in detail the usage of instruments, to track the performance of each stage, leading this to assume that for further application of the proposed process, a process of capacitation should be considered, while planning the different activities to be carried out. Also, when working with experienced developers or software engineers, the implantation of a new process may be difficult to be done, since they may already have established methods and practices, and could present a certain degree of resistance to change.

When it comes to the validation of one of the elaborated products as a mechanism to evaluate the process, it was possible to see that the goal of increasing the time per week that a patient practices got accomplished, in some cases up to a 90%. Also, from the obtained results, it may be seen, that the more experience the users (patients) got at

playing the game, the better were their results in terms of uttered words, elapsed time, and words per minute rate.

6. Conclusions and Future Work

As conclusion to this thesis work, in this chapter, a summary of each one of the other chapters is provided, the contributions are enlisted, the traceability matrix of specific objectives that was introduced in the first chapter is updated, and then, it is mentioned if the hypothesis was proved or refuted. Finally, the future work is described.

This first chapter provided a general panorama of the problem that is intended to solve, and about concepts that are important to be understood. On it, a research question was settled: "What elements, disciplines, or techniques may enrich a software development process for promoting the development of serious games to support auditory rehabilitation, in short time and with limited human resources involved?", to then formulate a related hypothesis: "The stages of a software development process model may be enriched with concepts on serious games design, rehabilitation, and HCI techniques or tools, in order to promote the production of serious games to support auditory rehabilitation in short time and with limited human resources involved", and finally to describe general and specific objectives.

The second chapter contains the state of the art, on it were addressed concepts on rehabilitation, model driven development, process models, verification and validation, serious games design elements, serious games for rehabilitation, and finally describing experiences on the development of serious games for acquiring skills on those projects that later on worked as baseline for the performed research.

Third chapter was aimed on the description of SEGA-ARM, a metamodel that was designed considering the experience gained with the former development of applications for rehabilitation and elements that were found in the literature, in compliance to a HCI approach, and expressed in accordance to a UML notation. This metamodel includes the expression of the user capabilities and disabilities, of the dynamic and static part of the therapy, of the context of use, and of the serious games design elements; and its usage is recommended for requirements gathering and analysis, and design stages of the proposal in chapter four.

The main proposal of this work was presented in the fourth chapter, this is, the enrichment of a generic software development process model with speech rehabilitation concepts, HCI based tools, and game design elements that were found on the literature. On its sections,

a workflow of the application of the process was presented, the general view of the process was explained, and finally each stage involved on the process model was detailed in terms of the actors, activities, and products that take part on it.

Verification of the proposed enriched process and validation of one of the developed products in compliance to it were described in the fifth chapter, along with the explanation of the experiments that were carried out to this effect.

The contributions that were delivered from this thesis work are:

- A state of the art reporting the result of the performed research for better understanding concepts on rehabilitation, model driven development, software development process models, and patterns.
- A comparison of software development process models that may be useful when deciding which one to implement in a project.
- A metamodel to promote the design of serious games to support auditory rehabilitation, allowing expressing the context of use, the static part of the therapy, the dynamic part of the therapy, and serious games design elements.
- The definition of activities, roles, and tools that take part of a software development process model enriched with concepts of rehabilitation, serious games design, and tools of a HCI approach.
- 3 applications and 3 serious games for supporting auditory rehabilitation produced during the different carried out experiments.
- Statistical data that was gathered from the interaction of patients with one of the produced serious games.

In the Introduction chapter, a specific objectives traceability matrix was created in order to make it easier to control and verify the accomplishment of the settled goals. That matrix was updated and the resultant one is presented in Table 6.1 Updated specific objectives traceability matrix..

Table 6.1 Updated specific objectives traceability matrix.

Objective number	Description	How was it accomplished?
1	Attend speech therapy sessions to observe and gather requirements to	Speech therapy sessions were observed at the “Centro de Atención

	develop applications to support a rehabilitation program.	Integral a la Salud" of the Universidad Autónoma de Aguascalientes in Mexico, and also interviews with the speech therapist were performed for gathering requirements to develop applications in compliance to the therapy objectives.
2	Elaborate applications in compliance with the gathered requirements in order to acquire experience in their development.	In the state of the art chapter, it is described that 8 engineering students participated in an experiment in which 3 applications were developed based on the requirements that were gathered from the observation of therapy sessions.
3	Compare software development process models from the literature.	Section 2.3.7 presents a comparison carried out among different process models in terms of the involved stages, recommended usage, scalability, productivity, and other characteristics.
4	Choose a process model and review its characteristics.	In the fourth chapter, it was told that a component based approach was considered to this effect.
5	Model the use context, in terms of platform, user, and environment.	In the third chapter, the context is modeled as part of the SEGA-ARM metamodel.
6	Elaborate a metamodel of serious games for auditory rehabilitation.	The SEGA-ARM metamodel is provided for accomplishing this objective.
7	Identify concepts about game design, which may be useful for the realization of this project.	In the state of the art, analysis is performed over game design elements, which are later listed, described, and finally structured as part of the SEGA-ARM metamodel.

8	Incorporate concepts on game design, rehabilitation, and HCI techniques or tools to the stages of the chosen process model in order to enrich it.	In the fourth chapter, the stages that take part on the enriched process are described.
9	Develop serious games to support auditory rehabilitation by following the enriched process model.	In the verification and validation report chapter, two serious games are described, and moreover one of them was used as part of a therapy program. The documentation for these projects is included in Appendix A and Appendix B.
10	Verify the proposal (enriched process model) and validate the developed products.	The enriched process was verified, comparing the projects that were developed under its guidelines against others that were not. The validation was carried out evaluating the interaction of end users with one of the developed serious games.
11	Generate publications of national and international impact reporting advances and results obtained during the performance of this thesis work.	During the realization of this thesis work, 7 research papers were published in national and international conferences reporting advances in the research. Two of them are included in Appendix D.

As it may be seen from the Table above, all of the specific objectives were reached, and therefore the general one was too. Taking this into account and by analyzing the results that were gathered from the verification and validation process, it is possible to say that the hypothesis:

> *H1.- The stages of a software development process model may be enriched with concepts on serious games design, rehabilitation, and Human-Computer Interaction techniques or tools, in order to promote the production of serious games to support auditory rehabilitation in short time and with limited human resources involved.*

was **proved**.

If well all of the objectives and goals of the present work were accomplished, there's still work to be carried out. As future work, along with the description of the activities that take part on the stages of the proposed process, tools, methodologies, technologies, and sub processes may be detailed in order to improve the results in terms of quality of the products. Along with this extension of the work, the proposal must be evaluated in different contexts and for a broader group of more experienced developers for gathering feedback of possible improvements.

Glossary of terms

Serious game:

A mental challenge played through a computer according to specific rules, which uses entertainment mechanisms to achieve training or capacitation goals.

Software development process model:

Process that is being selected for the development of the project depending on the project's aims and goals.

Rehabilitation:

Process by which physical, sensory, and mental capacities are restored or developed in (and for) people with disabling conditions.

Metamodel:

Definition of modeling elements along with their relationships and rules in order to enable the creation of semantic models, that may later result on the design of Software tools.

References

A

[Abt87]

Abt, C. C. (1987). Serious games. University Press of America.

[Aldr09]

Aldrich, C. (2009). The complete guide to simulations and serious games: how the most valuable content will be created in the age beyond Gutenberg to Google. John Wiley & Sons.

[Alex77]

Alexander, C., Ishikawa, S., & Silverstein, M. (1977). Pattern languages. Center for Environmental Structure, 2.

[AOTA81]

American Occupational Therapy Association Representative Assembly minutes. (1981). American journal of Occupational Therapy, 35, 792-802

[Arna13]

Arnab, S., Dunwell, I., & Debattista, K. (2013). Serious games for healthcare: Applications and implications. Medical Information Science Reference.

[Atki03]

Atkinson, C., & Kühne, T. (2003). Model-driven development: a metamodeling foundation. *Software, IEEE*, *20*(5), 36-41.

B

[BLS14a]

Bureau of Labor Statistics, U.S. Department of Labor, Occupational Outlook Handbook, 2014-15 Edition, Occupational Therapists, on the Internet at http://www.bls.gov/ooh/healthcare/occupational-therapists.htm (visited November 03, 2014).

[BLS14b]

Bureau of Labor Statistics, U.S. Department of Labor, Occupational Outlook Handbook, 2014-15 Edition, Physical Therapists, on the Internet at

http://www.bls.gov/ooh/healthcare/physical-therapists.htm (visited November 04, 2014).

[BLS14c]

Bureau of Labor Statistics, U.S. Department of Labor, Occupational Outlook Handbook, 2014-15 Edition, Speech-Language Pathologists, on the Internet at http://www.bls.gov/ooh/healthcare/speech-language-pathologists.htm (visited November 03, 2014).

[Boeh88]

Boehm, B. W. (1988). A Spiral Model of Software Development and Enhancement. Computer, (5), 61-72.

[Boeh91]

Boehm, B. W. (1991). Software risk management: principles and practices.Software, IEEE, 8(1), 32-41.

[Bran97]

Brandt Jr, E. N., & Pope, A. M. (Eds.). (1997). *Enabling America: Assessing the Role of Rehabilitation Science and Engineering*. National Academies Press.

[Breu10]

Breuer, J. S., & Bente, G. (2010). Why so serious? On the relation of serious games and learning. Eludamos. Journal for Computer Game Culture, 4(1), 7-24.

[Burd00]

Burdea, G., Popescu, V., Hentz, V., & Colbert, K. (2000). Virtual reality-based orthopedic telerehabilitation. Rehabilitation Engineering, IEEE Transactions on, 8(3), 430-432.

[Burk09]

Burke, J. W., McNeill, M. D. J., Charles, D. K., Morrow, P. J., Crosbie, J. H., & McDonough, S. M. (2009, March). Serious games for upper limb rehabilitation following stroke. In Games and Virtual Worlds for Serious Applications, 2009. VS-GAMES'09. Conference in (pp. 103-110). IEEE.

C

[Calv03]

Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., & Vanderdonckt, J. (2003). A unifying reference framework for multi-target user interfaces. Interacting with Computers, 15(3), 289-308.

[Carm95]

Carmel, E., & Becker, S. (1995). A process model for packaged software development. Engineering Management, IEEE Transactions on, 42(1), 50-61.

[Casa08]

Casas, Roberto, et al. (2008). User modelling in ambient intelligence for elderly and disabled people. Springer Berlin Heidelberg.

[Cesp14a]

Céspedes-Hernández, D., Valencia-Fernández, R., González-Flores, J., Álvarez Rodríguez, F., Camacho-González, Y., Félix, V., Muñoz-Arteaga, J., (2014). Desarrollo de un Videojuego Serio para Ejercitación de Praxias en Edad Infantil. En E. Archundia, M. León, et al. (Ed.), Aportaciones de Redes Innovadoras en Tecnología Educativa (pp. 3-13). Puebla, México: Benemérita Universidad Autónoma de Puebla, Dirección de Fomento Editorial.

[Cesp14b]

Céspedes- Hernández, D., Álvarez-Rodríguez, F., Muñoz-Arteaga, J., González-Calleros, J., Rodríguez-Vizzuett, L., (2014). Diseño de un Videojuego para Terapia de Lenguaje en Pacientes con Hipoacusia Bilateral Profunda con Implante Coclear. En M. Prieto, S. Pech, et al. (Ed). Technologies and Learning: Innovations and Experiences (pp. 226-230) Florida, Estados Unidos: Humboldt International University.

[Cesp14c]

Céspedes-Hernández, D., Zavala-García, D., Álvarez Rodríguez, F., González-Calleros, J., Rodríguez-Vizzuett, L., Guerrero-Bueno, R., (2014). Diseño y Desarrollo de un Videojuego Serio para Soporte a Terapias de Lenguaje Basado en Campos Semánticos. En A. García, F. Álvarez, et al. (Ed). Avances en las Tecnologías de la Información y Comunicaciones (pp. 504-511). Ciudad de México, México: Alfa Omega.

[Chin12]

Chin, S. B., Bergeson, T. R., & Phan, J. (2012). Speech intelligibility and prosody production in children with cochlear implants. Journal of communication disorders, 45(5), 355-366.

[Cicc84]

Ciccone, D. S., & Grzesiak, R. C. (1984). Cognitive dimensions of chronic pain.Social Science & Medicine, 19(12), 1339-1345.

[Cupp06]

Cuppens, E., Raymaekers, C., & Coninx, K. (2006). A model-based design process for interactive virtual environments. In Interactive Systems. Design, Specification, and Verification (pp. 225-236). Springer Berlin Heidelberg.

D

[Dete11]

Deterding, S., et al. (2011, September). From game design elements to gamefulness: defining gamification. In Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments (pp. 9-15). ACM.

F

[Feil93]

Feiler, P. H., & Humphrey, W. S. (1993, February). Software process development and enactment: Concepts and definitions. In Software Process, 1993. Continuous Software Process Improvement, Second International Conference on the (pp. 28-40). IEEE.

[Fent00]

Fenton, N. E., & Neil, M. (2000, May). Software metrics: roadmap. InProceedings of the Conference on the Future of Software Engineering (pp. 357-370). ACM.

[Fugg00]

Fuggetta, A. (2000, May). Software process: a roadmap. In Proceedings of the Conference on the Future of Software Engineering (pp. 25-34). ACM.

[Fran03]

Frankel, D.S. (2003). Model Driven Architecture: Applying MDA to Enterprise Computing, OMG Press.

G

[Gamm94]

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design patterns: elements of reusable object-oriented software. Pearson Education.

[Gay05]

Gay, D., Levis, P., & Culler, D. (2005). Software design patterns for TinyOS. ACM SIGPLAN Notices, 40(7), 40-49.

[Gepp05]

Geppert, B., Mockus, A., & Röbler, F. (2005, September). Refactoring for Changeability: A way to go?. In Software Metrics, 2005. 11th IEEE International Symposium (pp. 10-pp). IEEE.

[Glan90]

Glanze, W. D., Anderson, K. N., & Anderson, L. E. (1990). Mosby's Medical Nursing and Allied Health Dictionary. The Cv Mosby Co., USA, 630, 1134-1204.

[Gols90]

Golston, J. E., Moss, R. H., & Stoecker, W. V. (1990). Boundary detection in skin tumor images: an overall approach and a radial search algorithm. Pattern Recognition, 23(11), 1235-1247.

[Gros07]

Gros, B. (2007). Digital games in education: The design of games-based learning environments. Journal of Research on Technology in Education, 40(1), 23-38.

[Guer09]

Guerrero-García, J., González-Calleros, J. M., Vanderdonckt, J., & Muñoz-Arteaga, J. (2009, November). A theoretical survey of user interface description languages: Preliminary results. In Web Congress, 2009. LA-WEB'09. Latin American (pp. 36-43). IEEE.

H

[Hara00]

Harada, N. D., Chun, A., Chiu, V., & Pakalniskis, A. (2000). Patterns of rehabilitation utilization after hip fracture in acute hospitals and skilled nursing facilities. Medical care, 1119-1130.

[Heer06]

Heer, J., & Agrawala, M. (2006). Software design patterns for information visualization. Visualization and Computer Graphics, IEEE Transactions on, 12(5), 853-860.

[Hump89]

Humphrey, W. S., & Kellner, M. I. (1989, May). Software process modeling: principles of entity process models. In Proceedings of the 11th international conference on Software engineering (pp. 331-342). ACM.

I

[ISO08]

ISO, I. IEC 12207: 2008 Systems and software engineering-Software life cycle processes., 2008. International Organization for Standardization.

[ISO10]

ISO, I. IEC 24774: 2010 Systems and software engineering-Life cycle management-Guidelines for process definition.,2010. International Organization for Standardization.

[ISO99]

ISO, I. 13407: 1999 Human-centred Design Processes for Interactive Systems.,1999. International Organization for Standardization.

[Ittl12]

Ittleman, M. A. (2012). The Teaching of Talking: Learn to Do Expert Speech Therapy at Home with Children and Adults. Morgan James Publishing.

J

[Jaco99]

Jacobson, I., Booch, G., Rumbaugh, J., Rumbaugh, J., & Booch, G. (1999).The unified software development process (Vol. 1). Reading: Addison-wesley.

K

[Kakl13]

Kaklanis, N., A Coupled User and Task Modelling Methodology for Accessible Product Design. PhD dissertation, University of Surrey. March 2013.

[Kiku07]

Kikuchi, T., et al. (2007, June). Quasi-3-DOF rehabilitation system for upper limbs: its force-feedback mechanism and software for rehabilitation. In Rehabilitation Robotics, 2007. ICORR 2007. IEEE 10th International Conference on (pp. 24-27). IEEE.

[Koch03]

Koch, N., & Kraus, A. (2003). Towards a common metamodel for the development of web applications. In Web Engineering (pp. 497-506). Springer Berlin Heidelberg.

[Kuip13]

Kuipers, D. A., et al. (2013). Design for Transfer. In Serious Games Development and Applications (pp. 239-246). Springer Berlin Heidelberg.

[Kung08]

Kung, D., & Zhu, H. (2008). Software verification and validation. Wiley Encyclopedia of Computer Science and Engineering.

L

[Land03]

Landay, J. A., & Borriello, G. (2003). Design patterns for ubiquitous computing. Computer, 36(8), 93-95.

[Lang11]

Lange, B., Chang, C. Y., Suma, E., Newman, B., Rizzo, A. S., & Bolas, M. (2011, August). Development and evaluation of low cost game-based balance rehabilitation tool using the Microsoft Kinect sensor. In Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE(pp. 1831-1834). IEEE.

[Larm03]

Larman, C., & Basili, V. R. (2003). Iterative and incremental development: A brief history. Computer, (6), 47-56.

[Lend07]

Lenden, J. M., & Flipsen, P. (2007). Prosody and voice characteristics of children with cochlear implants. Journal of Communication Disorders, 40(1), 66-81.

[Loai13]

Loaiza, D., Oviedo, C., Castillo, A., Portilla, A., Alvarez, G., Linares, D., ... & Ivarez, G. (2013, September). A video game prototype for speech rehabilitation. In Games and Virtual Worlds for Serious Applications (VS-GAMES), 2013 5th International Conference on (pp. 1-4). IEEE.

[Long12]

Longstreet, C. S., & Cooper, K. (2012, July). A meta-model for developing simulation games in higher education and professional development training. In Computer Games (CGAMES), 2012 17th International Conference on (pp. 39-44). IEEE.

[Lv15]

Lv, Z., Esteve, C., Chirivella, J., & Gagliardo, P. (2015). A game based assistive tool for rehabilitation of dysphonic patients. arXiv preprint arXiv:1504.01030.

M

[Mayh99]

Mayhew, D. J. (1999). Business: Strategic development of the usability engineering function. interactions, 6(5), 27-34.

[McDo12]

McDonnell, A., & Akruvala, S. D. (2012). Rehabilitative Therapists in Geriatric Health Services. *Managing Geriatric Health Services*, 45.

[McDo06]

McDowell, P., Darken, R., Sullivan, J., & Johnson, E. (2006). Delta3D: a complete open source game and simulation engine for building military training systems. The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology, 3(3), 143-154.

[Mich05]

Michael, D. R., & Chen, S. L. (2005). Serious games: Games that educate, train, and inform. Muska & Lipman/Premier-Trade.

[Mill01]

Miller, J., & Mukerji, J. (2001). Model driven architecture (mda). Object Management Group, Draft Specification ormsc/2001-07-01.

[Mont07]

Montero, F., & López-Jaquero, V. (2007). IdealXML: an interaction design tool. In Computer-Aided Design Of User Interfaces V (pp. 245-252). Springer Netherlands.

N

[Niel02]

Nielsen, J., Clemmensen, T., & Yssing, C. (2002, October). Getting access to what goes on in people's heads?: reflections on the think-aloud technique. InProceedings of the second Nordic conference on Human-computer interaction(pp. 101-110). ACM.

[Ning96]

Ning, J. Q. (1996, August). A component-based software development model. In compsac (p. 0389). IEEE.

O

[OMG08]

OMG. "Software & Systems Process Engineering Meta-Model (SPEM) Specification v.2.0", 2008. In: http://www.omg.org/technology/documents/formal/spem.htm, Accessed in 01/2009.

P

[Pare08]

Parés, M. E., Rosales, J. J., & Colomina, I. (2008, January). Yet another IMU simulator: validation and applications. In EUROCOW" International Calibration and Orientation Workshop".

[Pate01]

Paternò, F., Mori, G., & Galiberti, R. (2001, March). CTTE: an environment for analysis and development of task models of cooperative applications. In CHI'01 Extended Abstracts on Human Factors in Computing Systems (pp. 21-22). ACM.

[Pres05]

Pressman, R. S. (2005). Software engineering: a practitioner's approach. Palgrave Macmillan.

R

[Reed88]

Reed, K. L. (1988). Occupational therapy articles in serial publications: an analysis of sources. Bulletin of the Medical Library Association, 76(2), 125.

[Rego10]

Rego, P., Moreira, P. M., & Reis, L. P. (2010, June). Serious games for rehabilitation: A survey and a classification towards a taxonomy. In Information Systems and Technologies (CISTI), 2010 5th Iberian Conference on (pp. 1-6). IEEE.

[Ritt10]

Ritterfeld, U., Cody, M., & Vorderer, P. (Eds.). (2010). Serious games: Mechanisms and effects. Routledge.

S

[Sale04]

Salen, K., & Zimmerman, E. (2004). Rules of play: Game design fundamentals. MIT press.

[Sama92]

Samal, A., & Iyengar, P. A. (1992). Automatic recognition and analysis of human faces and facial expressions: A survey. Pattern recognition, 25(1), 65-77.

[Sanc09]

Sánchez, J. L. G., Zea, N. P., & Gutiérrez, F. L. (2009). From usability to playability: Introduction to player-centred video game development process. In Human Centered Design (pp. 65-74). Springer Berlin Heidelberg.

[Sanc11]

Sanca Tinta, M. D. (2011). Tipos de investigación científica. Revista de Actualización Clínica Investiga, 12, 621.

[Scho11]

Schönauer, C., Pintaric, T., Kaufmann, H., Jansen-Kosterink, S., & Vollenbroek-Hutten, M. (2011, June). Chronic pain rehabilitation with a serious game using multimodal input. In Virtual Rehabilitation (ICVR), 2011 International Conference on (pp. 1-8). IEEE.

[Send03]

Sendall, S., & Kozaczynski, W. (2003). *Model transformation the heart and soul of model-driven software development* (No. LGL-REPORT-2003-007).

[Smit04]

Smith, A., Dunckley, L., French, T., Minocha, S., & Chang, Y. (2004). A process model for developing usable cross-cultural websites. Interacting with computers, 16(1), 63-91.

[Stef07]

Steffen, B., Margaria, T., Nagel, R., Jörges, S., & Kubczak, C. (2007). Model-driven development with the jABC. In Hardware and Software, Verification and Testing (pp. 92-108). Springer Berlin Heidelberg.

T

[Tayl02]

Taylor, L. N. (2002). Video games: Perspective, point-of-view, and immersion (Doctoral dissertation, University of Florida).

U

[Uhl08]

Uhl, A. (2008). Model-driven development in the enterprise. Software, IEEE,25(1), 46-49.

V

[Van03]

Van Welie, M., & Van der Veer, G. C. (2003, September). Pattern languages in interaction design: Structure and organization. In Proceedings of interact (Vol. 3, pp. 1-5).

[Vand05]

Vanderdonckt, J. (2005, January). A MDA-compliant environment for developing user interfaces of information systems. In Advanced Information Systems Engineering (pp. 16-31). Springer Berlin Heidelberg.

[Verm04]

Vermunt, J. D., & Vermetten, Y. J. (2004). Patterns in student learning: Relationships between learning strategies, conceptions of learning, and learning orientations. Educational psychology review, 16(4), 359-384.

[Vent82]

Ventry, I. M., & Weinstein, B. E. (1982). The hearing handicap inventory for the elderly: a new tool. Ear and hearing, 3(3), 128-134.

[Viei09]

Vieira, V., Tedesco, P., & Salgado, A. C. (2009, April). A process for the design of context-sensitive systems. In Computer Supported Cooperative Work in Design, 2009. CSCWD 2009. 13th International Conference on (pp. 143-148). IEEE.

W

[Wall89]

Wallace, D. R., & Fujii, R. U. (1989). Software verification and validation: an overview. IEEE Software, (3), 10-17.

[Will88]

Williams, L. G. (1988, April). Software process modeling: a behavioral approach. In Proceedings of the 10th international conference on Software engineering (pp. 174-186). IEEE Computer Society Press.

Y

[Yilm06]

Yilmaz, A., Javed, O., & Shah, M. (2006). Object tracking: A survey. Acm computing surveys (CSUR), 38(4), 13.

Z

[Zyda05]

Zyda, M. (2005). From visual simulation to virtual reality to games. Computer, 38(9), 25-32.

Appendix A

Project: Desarrollo de un juego serio que ayuda a la mejora de la pronunciación y reconocimiento de palabras mediante reconocimiento de voz.

1 Introducción

Esta investigación tiene como fin desarrollar un juego serio el cual sea capaz de ayudar a los niños con discapacidades auditivas y de lenguaje por medio de actividades didácticas y entretenidas, dando como resultado una actividad de terapia la cual pueda ser llevada a cabo en cualquier lugar y día de la semana.

Siendo que las terapias se llevan a cabo mínimo una hora por semana, de manera que se pierde seguimiento por casi siete días y es responsabilidad de los familiares ayudar al paciente a practicar las actividades indicadas por el terapeuta. (Secretaria de Salud, 2010)

Sin embargo no siempre es posible brindar este tipo de atención y apoyo dando como resultado un mínimo progreso. De manera que el paciente deja de practicar, facilitando que olvide o no recuerde las habilidades adquiridas durante esa hora de terapia.

De manera que un juego serio el cual el paciente pueda utilizar por sí mismo, daría paso a la práctica continua sin sustituir la terapia, si no complementándola reduciendo el tiempo de rehabilitación. Aprovechando que en la actualidad el 87% de la población en México cuenta con un dispositivo móvil inteligente es factible llegar a los pacientes por medio de una aplicación móvil. (ITU, 2015)

1.1 Propósito

Ayudar a la rehabilitación de niños con discapacidades auditivas y de lenguaje por medio del desarrollo de un juego serio entretenido y divertido capaz de realizar actividades terapéuticas que mejoren la calidad de vida de los pacientes.

1.2 Glosario

Disfasia: Retraso en la adquisición del lenguaje, una gran pobreza de vocabulario, grandes dificultades gramaticales y falta de espontaneidad a nivel de expresión verbal. (Ronda, J. A. .2008)

Hipoacusia: Incapacidad total o parcial para escuchar sonidos en uno o ambos oídos. (MediPlus, 2014)

Juegos Serios: Son aquellos juegos que se usan para educar, entrenar e informar. (Michael, D. & Chen, S. 2006)

Rehabilitación: Conjunto de acciones y programas dirigidos a la utilización del potencial máximo de crecimiento personal de un individuo, que le permita superar o disminuir desventajas adquiridas a causa de su enfermedad en los principales aspectos de su vida diaria. (Herrera, A. et al, 1998)

Terapia de lenguaje: Tratamiento que se lleva a cabo para resolver las dificultades para expresar ideas. (Nelson, A. 2008)

Terapia del habla: Tratamiento que se lleva a cabo para resolver problemas con la producción de sonidos. (Nelson, A. 2008)

1.3 Referencias

Secretaría de Salud. (2010). Hipoacusia Neurosensorial Bilateral e Implante Coclear. México.

ITU. (2015). Mobile-cellular subscriptions. Julio 15, 2015, de ITU Sitio web: http://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx

Rondal, J. A. (1982). El desarrollo del lenguaje. Barcelona: Médica y Técnica.

MediPlus. (2014). Hipoacusia. Julio 15, 2015, de MediPlus Sitio web: http://www.nlm.nih.gov/medlineplus/spanish/ency/article/003044.htm

Michael, D. & Chen, S. (2006). Serious Games. Games that educate, train and infoms. Canadá: Thonsom.

Herrera, A. & Gamiochipi, J.A. (1998). Rehabilitación temprana en pacientes hospitalizados en unidades psiquiátricas. Psiquis - Hospital Psiquiátrico Fray Bernardino Alvarez, 7, 128-141.

Nelson, A. (2008). La terapia del lenguaje y del habla. Julio 12, 2015, de KidsHealth.org Sitio web: http://kidshealth.org/parent/en_espanol/crecimiento/speech_therapy_esp.html#

2 Descripción general del proyecto

2.1 Objetivos particulares

- Uso del reconocimiento de voz para analizar que la pronunciación del paciente sea correcta.
- Síntesis de voz de manera que el paciente escuche la pronunciación correcta de la palabra a aprender.
- Generar un banco de palabras que sean fáciles de analizar por el software y que las pueda utilizar el paciente en su vida cotidiana.
- Crear graficas decentes que logre captar la atención del paciente.
- Desarrollar varios niveles con distintas dificultades para que el paciente logre progresar periódicamente. Y sea un mayor reto conforme avanza en el juego.

2.2 Objetivo general

Llevar a cabo el diseño y desarrollo de un juego serio, con el que el paciente logre practicar y aprender como pronunciar palabras de manera correcta por medio de una actividad que no sea monótona ni aburrida.

Permitiéndole practicar a cualquier hora y momento sin necesidad de asistir a un consultorio o presencia del terapeuta. Logrando que el paciente haga terapia por más tiempo.

2.3 Características de los usuarios

Los usuarios a los que está orientado el software son pacientes con discapacidades auditivas (hipoacusia), habla o lenguaje (Disfasia, Afasia, etc.) imposibilitándoles la correcta pronunciación de palabras.

De manera que tienen problemas de fluidez en el habla, variación en la tonalidad y volumen, así como problemas al distinguir palabras o escuchar determinadas frecuencias.

3 Requisitos del juego serio

3.1 Requisitos funcionales

RF01 - Análisis de la correcta pronunciación de la palabra.

RF02 - Reproducción de sonidos con la correcta pronunciación de las palabras.

RF03 - Contadores de tiempo.

RF04 - Acumuladores de puntaje.

RF05 - Banco de palabras que sirvan como terapia.

3.2 Requisitos no funcionales

RN01 - Imágenes de los objetos.

RN02 - Menú inicial.

RN03 - Iniciar sesión con una cuenta personalizada del usuario.

RN04 - Animaciones de los personajes.

RN05 - Visualización del puntaje y el tiempo que le resta para pronunciar la palabra.

RN06 - Uso de colores brillantes para atraer la atención de los niños.

RN07 - Personajes con vestimenta del momento.

RN08 - Tabla de puntajes de todos los usuarios.

4 Apéndices

4.1 Reportes de asistencia a sesiones de terapia de lenguaje

4.2 Instanciación del metamodelo de contexto, terapia estática y terapia dinámica

4.3 Instrumento de contexto

La aplicación deberá ser utilizada en lugares con poco ruido para su correcto funcionamiento, como en el consultorio durante la terapia o en casa del paciente.

4.4 Instrumento de terapia estática

4.5 Instrumento de terapia dinámica

El paciente será un cazador el cual tendrá que alimentar a los animales para evitar que lo ataquen, para ello será necesario pronunciar el nombre correcto del animal que se encuentra hambriento lo más rápido posible, si desconoce el nombre del animal podrá tocar al animal y se reproducirá un sonido con la correcta pronunciación.

4.6 Matriz de trazabilidad de requerimientos

RF01	Análisis de la correcta pronunciación de la palabra.	
RF02	Reproducción de sonidos con la correcta pronunciación de las palabras.	
RF03	Contadores de tiempo.	
RF04	Acumuladores de puntaje.	
RF05	Banco de palabras que sirvan como terapia.	
RN01	Imágenes de los objetos.	

RN02	Menú inicial.	
RN03	Iniciar sesión con una cuenta personalizada del usuario.	
RN04	Animaciones de los personajes.	
RN05	Visualización del puntaje y el tiempo que le resta para pronunciar la palabra.	
RN06	Uso de colores brillantes para atraer la atención de los niños.	
RN07	Personajes con vestimenta del momento.	
RN08	Tabla de puntajes de todos los usuarios.	

Historia

Un cazador intenta sobrevivir en la selva del ataque de los animales salvajes, de manera que este les disparará con su pistola de agua para alejarlos. Para ello el jugador debe pronunciar de manera correcta el nombre de los animales y así poder activar el arma.

Funcionamiento

El juego inicia cargando la puntuación, el número de la ronda en que se encuentra y el botón para pausar el juego.

De manera que el jeep safari entra por el lado izquierdo de la pantalla para dejar al cazador en medio de la pantalla.

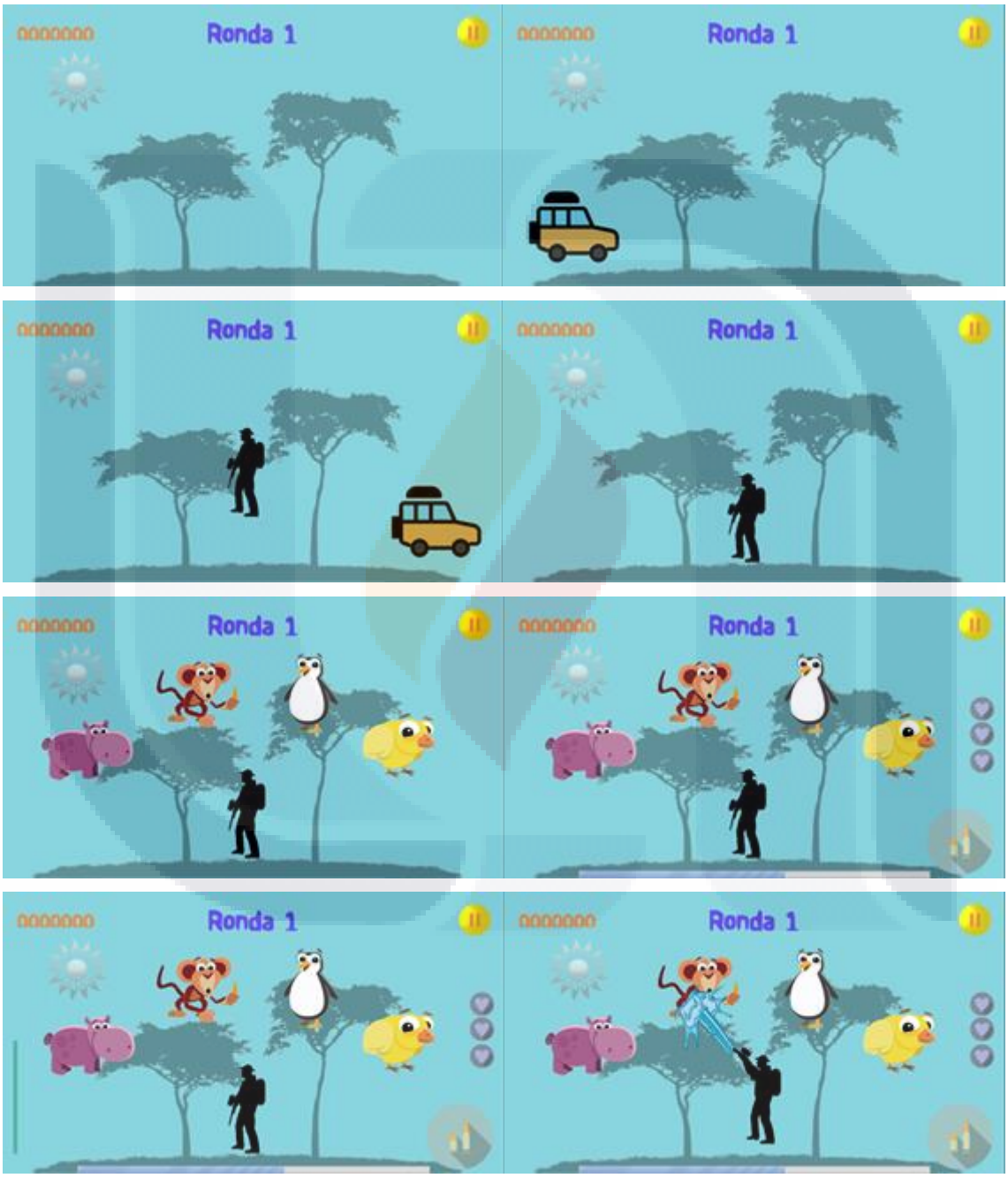
Una vez esté en su posición se agregan el número de vidas y la barra del tiempo.

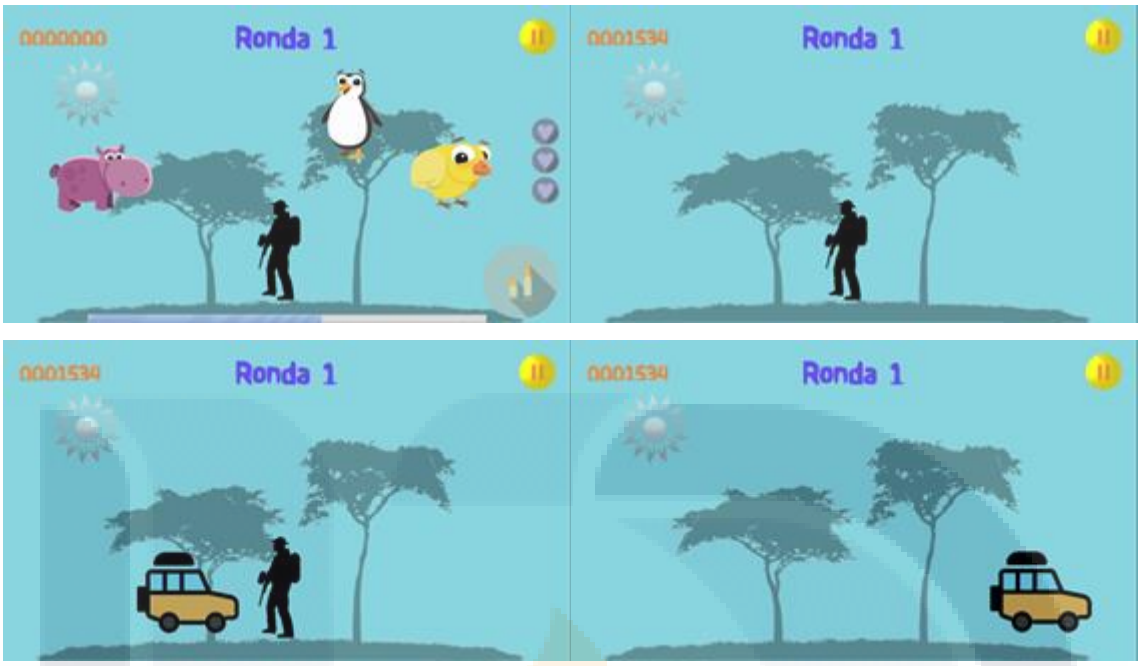
Se procede a un conteo de 3 segundo para avisar al jugador que está a punto de iniciar, al momento que este llega a su fin se habilita el botón que activa el reconocimiento de voz y comienza a disminuir la barra de tiempo.

El jugador debe presionar el botón de disparo para activar el reconocimiento de voz, mientras este habla se mostrará un barra que retroalimentará al usuario para indicarle que se le está escuchando.

Si la palabra que pronuncio es la adecuada entonces el cazador apunta al animal y le dispara un chorro de agua.

Guión Gráfico





Elementos de diseño de juego

Personajes:

Primarios: Cazador.

Secundarios: Animales de los cuales algunos serán inofensivos y otros no tanto.

Descripción Narrativa: El jugador será un cazador el cual se encontrará en medio de la pantalla. Se aparecen varios animales al mismo tiempo, quienes intentarán comer al cazador, de manera que tendrá que decir el nombre del animal al que desea disparar lo más rápido posible antes de que el animal ataque al cazador.

Retos: Sobrevivir al ataque del o de los animales que lo atacan dentro de un determinado tiempo.

Vistas de juego: De arriba y tercera persona de arrastre.

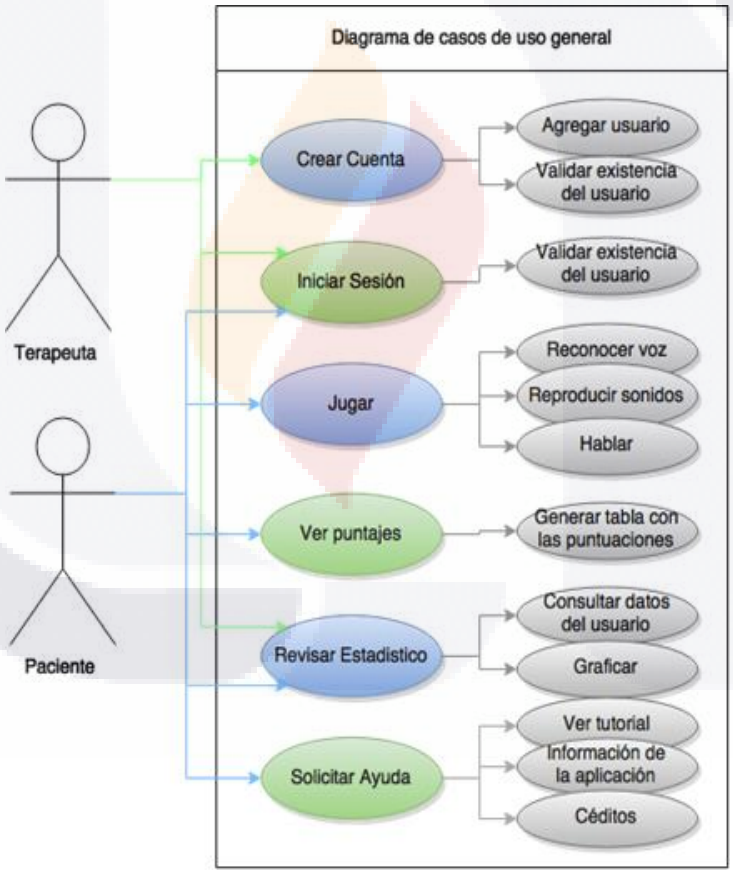
Restricciones: Serán rondas que durarán un determinado tiempo, el cual dependerá del nivel.

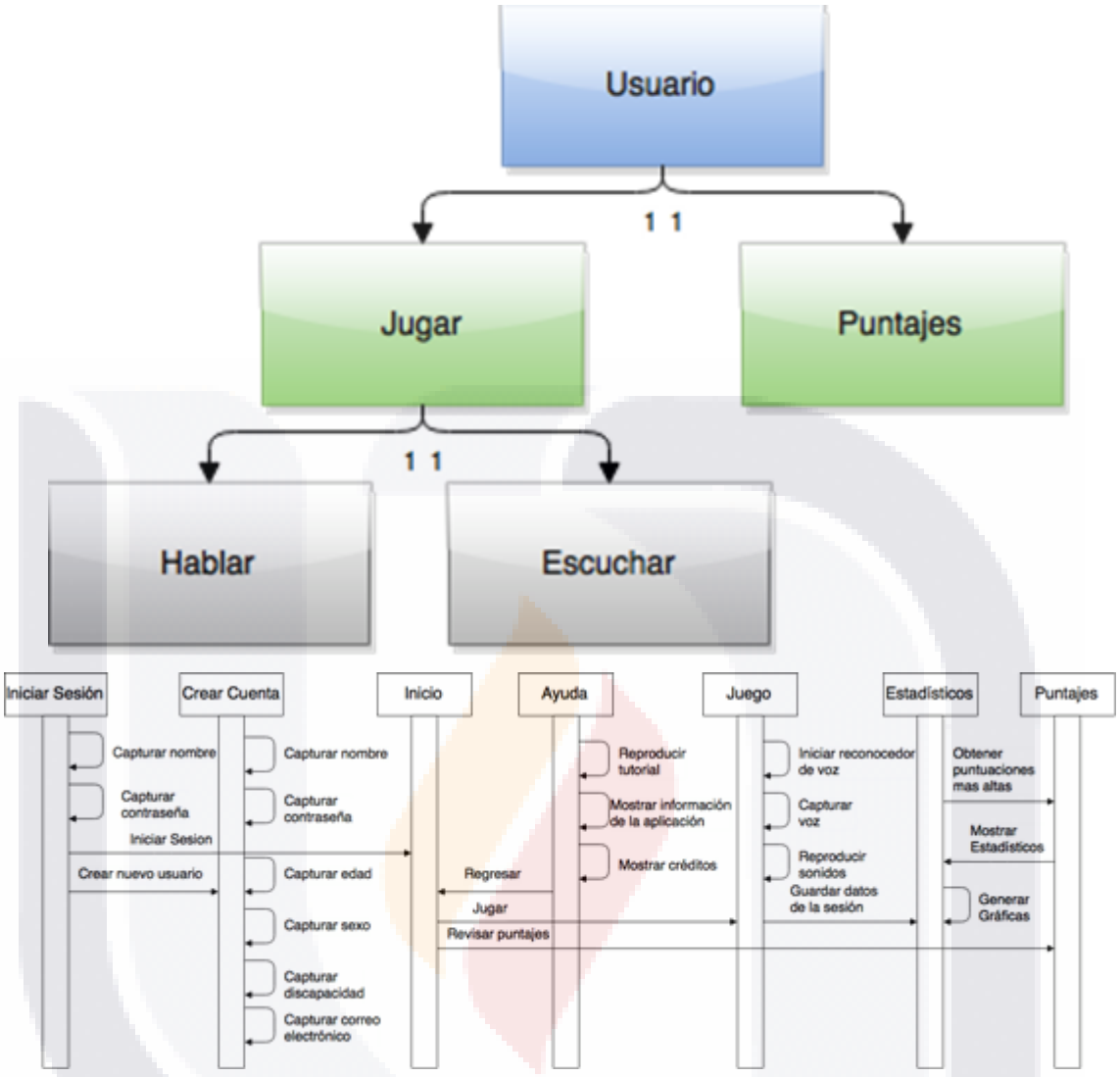
Recompensas y castigos: Tendrá un número determinado de vidas que perderá conforme es atacado y conforme vaya sobreviviendo el tiempo de que dure la ronda será menor, ganando puntos entre más rápido responda más punto obtendrá.

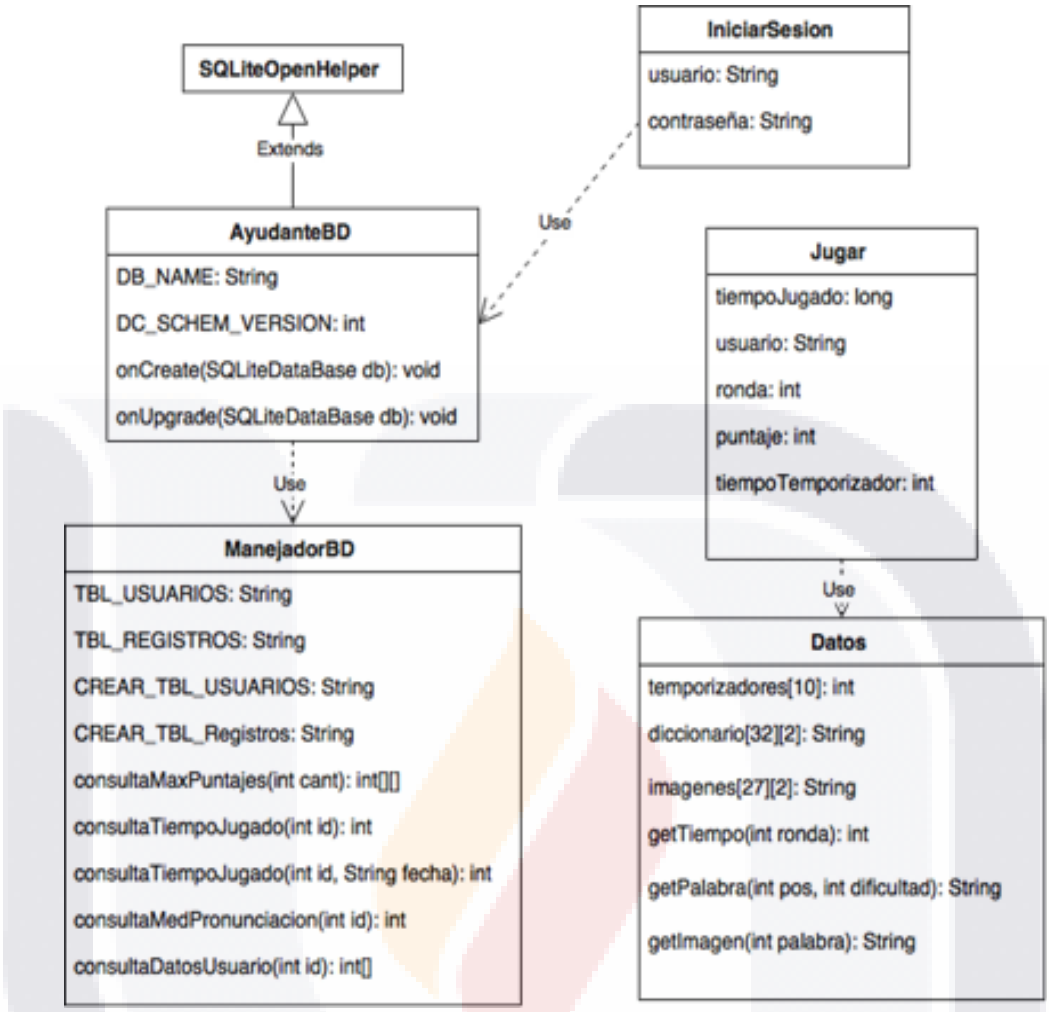
Elementos interactivos: El cazador irá sobre un carro de safari que avanzará cada que sube de nivel.

Retroalimentación: Al inicio de la ronda se activará el reconocimiento de voz y le avisará si está escuchando, y aparecerá una medalla cada vez que acierte.

Diagramas conceptuales

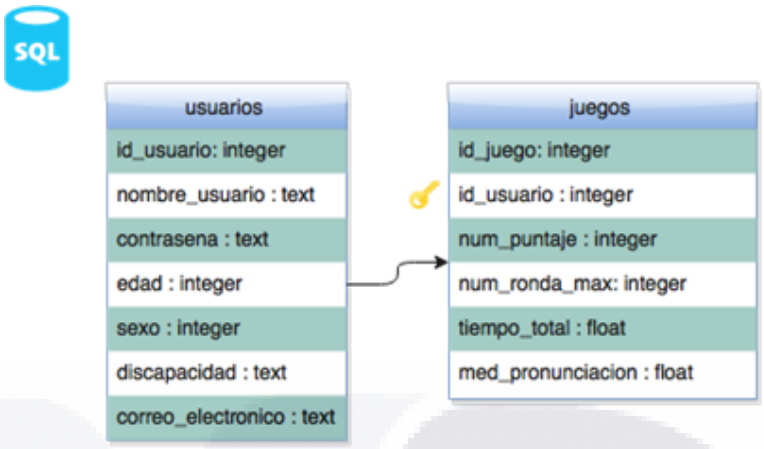






Modelado de Datos

Para la información que se manejara lo más conveniente es una base de datos en la cual se registren los usuarios, los datos del juego y las palabras del diccionario.



Interfaces de Usuario

Las interfaces deberán ser sencillas y fáciles de acceder, siendo que los usuarios principales serán niños, así que entre menos pestañas y el uso de iconos fáciles de entender es primordial para el desarrollo de estas.

El inicio solo cuenta con un imagen de fondo, el nombre del videojuego y los botones para iniciar el juego, tabla de posiciones, ayuda y ajustes.

La parte de ayuda, contiene 3 opciones la primera muestra un video del modo de juego y como es su funcionamiento, la segunda de información de la investigación y por último los créditos del juego.

La tabla de posiciones o líderes, se puede acceder desde el inicio y muestran los primeros 7 puntajes más altos con sus respectivos usuarios.

El juego consiste en una interfaz sencilla que muestra toda información del estado del juego, una barra como temporizador, puntaje, número de ronda, animación que muestra si se está escuchando al usuario y el botón de disparo.

Creación de Diccionario

- Pollo*
- Oveja**
- Chango/Mono/Simio*
- Pingüino**
- Gallina**/Gallo*
- Hipopótamo***
- Cabra**
- Elefante**
- Puerco/Cerdo**
- Serpiente***
- Conejo**
- Panda**
- Perro**/Can*
- Rinoceronte***

- Jaguar**
- Zorro**
- Camello**
- Cebra*
- Avestruz**
- Tucán*
- León*
- Cocodrilo***
- Canguro**
- Oso Hormiguero***
- Caballo*
- Pato*
- Gato*

Dificultad: *Sencilla **Media ***Difícil

Matriz de trazabilidad

RF01	Análisis de la correcta pronunciación de la palabra.	
RF02	Reproducción de sonidos con la correcta pronunciación de las palabras.	
RF03	Contadores de tiempo.	
RF04	Acumuladores de puntaje.	
RF05	Banco de palabras que sirvan como terapia.	
RN01	Imágenes de los objetos.	
RN02	Menú inicial.	
RN03	Iniciar sesión con una cuenta personalizada del usuario.	
RN04	Animaciones de los personajes.	
RN05	Visualización del puntaje y el tiempo que le resta para pronunciar la	

	palabra.	
RN06	Uso de colores brillantes para atraer la atención de los niños.	
RN07	Personajes con vestimenta del momento.	
RN08	Tabla de puntajes de todos los usuarios.	

Herramientas a utilizar

Lenguaje de programación

Para dar solución a los objetivos planteados y cumplir con todos los requerimientos, se llevará a cabo la programación para dispositivos móviles Android con ayuda de varios lenguajes de programación como java, sql, css y xml.

Reconocimiento de voz

El reconocimiento de voz, se llevará a cabo con el api de “google speech” la cual captura la voz del usuario, se digitaliza y envía al servidor central de Google que devuelve el texto reconocido en un arreglo con las posibles palabras u oraciones que se pronunció.

Sintetizador de Voz

De igual manera Google cuenta con el api para síntesis de voz compatible con distintas plataformas el cual se denomina speech.tts, donde se analiza una cadena de caracteres y genera un clip de audio.

Base de Datos

Android es compatible con las bases de datos SQLite “android.database.sqlite”, lo que nos permitirá almacenar de manera ordenada los usuarios y los puntajes de los jugadores, de manera que se pueda analizar al usuario de acuerdo a las actividades que ha realizado.

Diseño Gráfico

El diseño es una parte importante, ya que debe llamar la atención a niños y ser presentable, para ello se utiliza el programa de edición fotográfica Gimp para darle efecto y diseño a los fondos. Además del software vectorial Inkscape con el cual se pueden crear los iconos, botones y personajes de la historia.

Definición de un estándar de programación

El lenguaje de programación a utilizar es del tipo orientada a objetos muy similar a java, pero utiliza otras librerías y una máquina virtual diferente a la proporcionada por Oracle. Pero la manera en que se lleva a cabo la codificación es la misma por lo tanto se utilizaran los mismo estándares para el desarrollo de las clases que utiliza este lenguaje. Además para el desarrollo de interfaces gráficas existe una herramienta que mejora la manera de trabajar, muy parecida a CSS (Utilizada para dar diseño a páginas web) por lo que se intentará seguir también estándares parecidos aunque no se trate del mismo lenguaje.

1.1 Java

1.1.1 Organización de ficheros por paquetes

Para mantener organización en las clases, estas se organizan por paquetes los cuales brindan jerarquía y así definir qué clases tiene acceso a otras clases. Asegurando su mantenimiento, jerarquía y de fácil lectura si alguien externo al proyecto en un futuro desea mejorar o agregar funciones.

1.1.2 Fichero java

Estos contienen únicamente una solo clase, es decir, no clases anidadas una dentro de otra en caso de tener que pasar datos entre clases, se hará uso de variables globales o de sesión.

También estos ficheros contienen comentarios en el encabezado con una descripción de la clase, interfaz, etc. sobre cuál es su función.

Ejemplo:

```
1 //Descripción de la clase
2 public class MyClase {
3
4     variables
5
6     métodos
7 }
8
```

1.1.3 Comentarios

Estos deben ser claros y concisos, con una descripción general de una parte específica del código, aunque de igual manera los nombres de las variables y métodos deben tener un nombre claro y fácil de entender es necesario de un texto corto sobre el funcionamiento o el uso de este.

1.1.4 Sentencia de paquete

La primera línea de código no comentada de una clase debe ser el nombre del paquete al que pertenece esta.

1.1.5 Sentencias de importación

Solo se deben incluir las sentencias necesarias, y que en realidad se haga uso de estas, además de agrupar aquellas que son para una determinada función o de alguna librería externa.

1.1.6 Declaraciones de interfaces y clases

Estas deberán ser instanciadas con el mismo nombre de la clase o alguna abreviación e iniciando con minúscula. Evitar de instanciar la misma clase en diferentes clases, y aplicar los conceptos de herencia para ahorrar código.

Ejemplo:

```
1 //Descripción de la clase
2 public class MyClase {
3
4     Datos datos = new Datos();
5 }
6
```

1.1.7 Declaración de variables

Android brinda una herramienta muy útil para el diseño a través de ficheros xml, los cuales pueden ser ligados al código en java para instanciar los objetos de la interfaz gráfica y hacer uso de sus métodos. De manera que se hará uso de esta herramienta

para aquellos componentes que interactúan con el usuario, como botones, imágenes, campos de texto, etc.

Para el caso de variables, arreglos o matrices globales de una clase, estas deberán ser privadas y en caso de ser necesario manipular o utilizar esta información por otra clase, se debe encapsular dichos atributos. Además sus nombres deben ser claros y fáciles de identificar.

```
1 //Descripción de la clase
2 public class Persona {
3
4     String nombre; //nombre de la persona
5     long matricula; //matricula de 10 caracteres
6     int edad; //edad de la persona
7
8 }
9
```

1.2 Ficheros XML

Este tipo de ficheros presentan información de manera estructurada. Para almacenar y transmitir información hay que encontrar una forma adecuada de representación en el medio disponible como soporte, Android Studio con su herramienta de autocompletado brinda los nombres de los tags o notaciones admitidas.

```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <!-- Texto de los desarrolladores de la aplicacion-->
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/texto_creditos"
        android:layout_margin="10dp"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"/>

    <!-- Boton para volver atras -->
    <Button
        android:id="@+id/boton_atras_creditos"
        android:layout_width="55dp"
        android:layout_height="55dp"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_margin="10dp"
        android:background="@drawable/boton_atras"/>

</RelativeLayout>
```

1.2.1 Caracteres

XML brinda soporte para gran variedad de codificadores de caracteres, por lo que se puede hacer uso de todos los caracteres ASCII sin embargo para la codificación del XML en Android solo es admitido el UTF-8.

Por otro lado algunos caracteres se reservan para el marcado como lo son: < > & “ ’, de manera que si se desea utilizar a lo largo de un texto estos se pueden sustituir por &lt &gt &amp &apos &quot respectivamente, esto hará que aparezcan como un carácter en lugar de realizar una función.

1.2.2 Estructura

La estructura general de un documento XML está formada por tres partes:

1.2.2.1 Prólogo, opcional:

Conteniendo una secuencia de instrucciones de procesamiento y/o declaración de tipo de documento.

1.2.2.2 Cuerpo:

Un árbol único de elementos marcados, con anidamiento estricto.

1.2.2.3 Epílogo, opcional:

Conteniendo una secuencia de instrucciones de procesamiento.

1.2.3 Marcas

Son fragmentos de información delimitados por marcas, de la siguiente manera:

Marca inicial: <x ....>

Contenido: texto u otros elementos.

Marca final: </x>

1.2.4 Comentarios

Al igual que en los ficheros de java, se debe colocar comentarios.

Appendix B

Project: Desarrollo de un juego serio para soporte a una actividad de terapia de lenguaje.

1 Introducción

Esta investigación tiene como fin desarrollar un juego serio el cual sea capaz de ayudar a los niños con discapacidades auditivas y de lenguaje por medio de actividades didácticas y entretenidas, dando como resultado una actividad de terapia la cual pueda ser llevada a cabo en cualquier lugar y día de la semana.

Siendo que las terapias se llevan a cabo mínimo una hora por semana, de manera que se pierde seguimiento por casi siete días y es responsabilidad de los familiares ayudar al paciente a practicar las actividades indicadas por el terapeuta. (Secretaria de Salud, 2010)

Sin embargo no siempre es posible brindar este tipo de atención y apoyo dando como resultado un mínimo progreso. De manera que el paciente deja de practicar, facilitando que olvide o no recuerde las habilidades adquiridas durante esa hora de terapia.

De manera que un juego serio el cual el paciente pueda utilizar por sí mismo, daría paso a la práctica continua sin sustituir la terapia, si no complementándola reduciendo el tiempo de rehabilitación. Aprovechando que en la actualidad el 87% de la población en México cuenta con un dispositivo móvil inteligente es factible llegar a los pacientes por medio de una aplicación móvil. (ITU, 2015)

1.1 Propósito

Ayudar a la rehabilitación de niños con discapacidades auditivas y de lenguaje por medio del desarrollo de un juego serio entretenido y divertido capaz de realizar actividades terapéuticas que mejoren la calidad de vida de los pacientes.

1.2 Glosario

Hipoacusia: Incapacidad total o parcial para escuchar sonidos en uno o ambos oídos. (MediPlus, 2014)

Juegos Serios: Son aquellos juegos que se usan para educar, entrenar e informar. (Michael, D. & Chen, S. 2006)

Rehabilitación: Conjunto de acciones y programas dirigidos a la utilización del potencial máximo de crecimiento personal de un individuo, que le permita superar o disminuir desventajas adquiridas a causa de su enfermedad en los principales aspectos de su vida diaria. (Herrera, A. et al, 1998)

Terapia de lenguaje: Tratamiento que se lleva a cabo para resolver las dificultades para expresar ideas. (Nelson, A. 2008)

Terapia del habla: Tratamiento que se lleva a cabo para resolver problemas con la producción de sonidos. (Nelson, A. 2008)

1.3 Referencias

Secretaría de Salud. (2010). Hipoacusia Neurosensorial Bilateral e Implante Coclear. México.

ITU. (2015). Mobile-cellular subscriptions. Julio 15, 2015, de ITU Sitio web: http://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx

Rondal, J. A. (1982). El desarrollo del lenguaje. Barcelona: Médica y Técnica.

MediPlus. (2014). Hipoacusia. Julio 15, 2015, de MediPlus Sitio web: http://www.nlm.nih.gov/medlineplus/spanish/ency/article/003044.htm

Michael, D. & Chen, S. (2006). Serious Games. Games that educate, train and infoms. Canadá: Thonsom.

Herrera, A. & Gamiochipi, J.A. (1998). Rehabilitación temprana en pacientes hospitalizados en unidades psiquiátricas. Psiquis - Hospital Psiquiátrico Fray Bernardino Alvarez, 7, 128-141.

Nelson, A. (2008). La terapia del lenguaje y del habla. Julio 12, 2015, de KidsHealth.org Sitio web: http://kidshealth.org/parent/en_espanol/crecimiento/speech_therapy_esp.html#

2 Descripción general del proyecto

2.1 Objetivos particulares

- Uso del reconocimiento de voz para analizar que la pronunciación del paciente sea correcta.
- Síntesis de voz de manera que el paciente escuche la pronunciación correcta de la palabra a aprender.
- Generar un banco de palabras que sean fáciles de analizar por el software y que las pueda utilizar el paciente en su vida cotidiana.
- Crear gráficos atractivos que logren captar la atención del paciente.

2.2 Objetivo general

Llevar a cabo el diseño y desarrollo de un juego serio, con el que el paciente logre practicar y aprender como pronunciar palabras de manera correcta por medio de una actividad que no sea monótona ni aburrida.

Permitiéndole practicar a cualquier hora y momento sin necesidad de asistir a un consultorio o presencia del terapeuta. Logrando que el paciente haga terapia por más tiempo.

2.3 Características de los usuarios

Los usuarios a los que está orientado el software son pacientes con discapacidades auditivas (hipoacusia), habla o lenguaje (Disfasia, Afasia, etc.) imposibilitándoles la correcta pronunciación de palabras.

De manera que tienen problemas de fluidez en el habla, variación en la tonalidad y volumen, así como problemas al distinguir palabras o escuchar determinadas frecuencias.

3 Requisitos del juego serio

3.1 Requisitos funcionales

RF01 - Análisis de la correcta pronunciación de la palabra.

RF02 - Reproducción de sonidos con la correcta pronunciación de las palabras.

RF03 - Contadores de tiempo.

RF04 - Acumuladores de puntaje.

RF05 - Banco de palabras que sirvan como terapia.

3.2 Requisitos no funcionales

RN01 - Imágenes de hojas que contengan las palabras a pronunciar.

RN02 - Menú inicial.

RN03 - Iniciar sesión con una cuenta personalizada del usuario.

RN04 - Animaciones de algunos personajes.

RN05 - Visualización del puntaje y el tiempo que le resta para pronunciar la palabra.

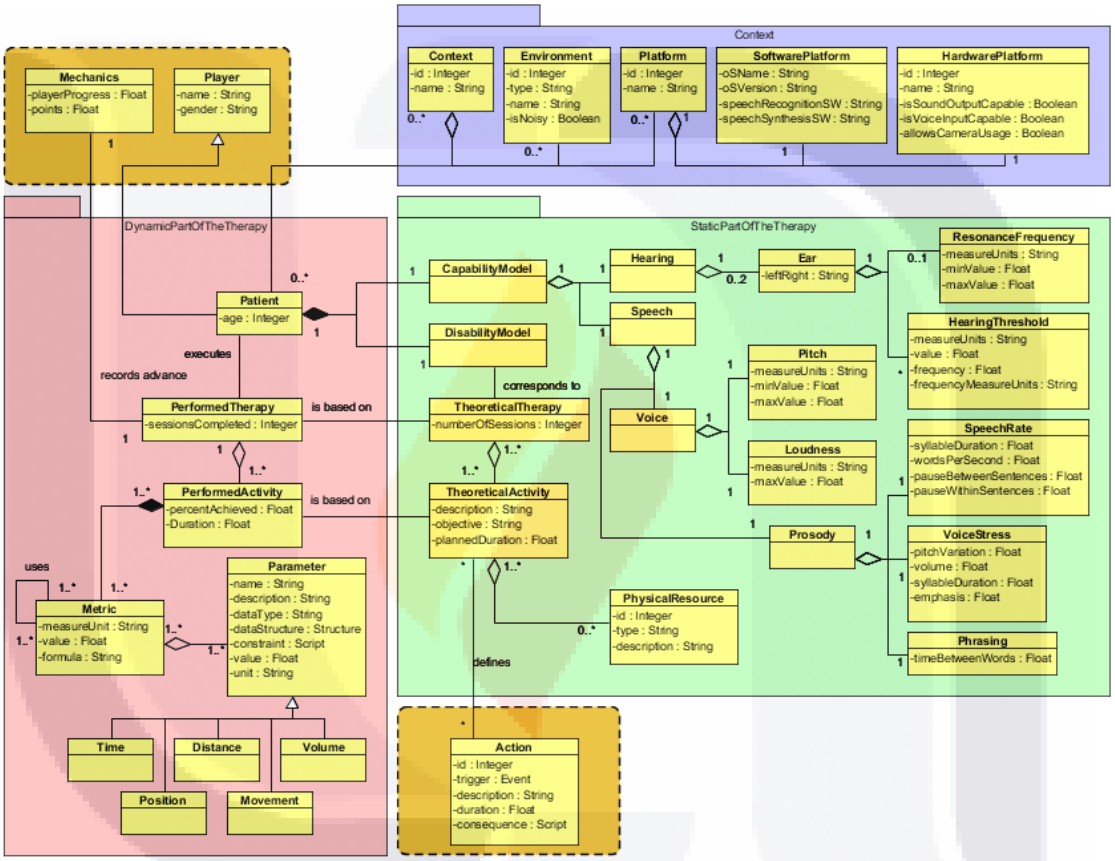
RN06 - Uso de colores brillantes para atraer la atención de los niños.

RN07 - Tabla de puntajes de todos los usuarios.

4 Apéndices

4.1 Reportes de asistencia a sesiones de terapia de lenguaje

4.2 Instanciación del metamodelo de contexto, terapia estática y terapia dinámica



4.3 Instrumento de contexto

La aplicación deberá ser utilizada en lugares con poco ruido para su correcto funcionamiento, como en el consultorio durante la terapia o en casa del paciente.

4.4 Instrumento de terapia estática

4.5 Instrumento de terapia dinámica

El paciente será una rana la cual tendrá que alimentarse de hojas y mosquitos, para ello será necesario pronunciar la palabra que hay en cada hoja para que la rana brinque de hoja en hoja, la rana puede caer al agua en caso de que no se pronuncie de manera correcta la palabra que se muestra.

4.6 Matriz de trazabilidad de requerimientos

RF01	Análisis de la correcta pronunciación de la palabra.	
RF02	Reproducción de sonidos con la correcta pronunciación de las palabras.	
RF03	Contadores de tiempo.	
RF04	Acumuladores de puntaje.	
RF05	Banco de palabras que sirvan como terapia.	
RN01	Imágenes de los objetos.	
RN02	Menú inicial.	
RN03	Iniciar sesión con una cuenta personalizada del usuario.	
RN04	Animaciones de los personajes.	
RN05	Visualización del puntaje y el tiempo que le resta para pronunciar la palabra.	
RN06	Uso de colores brillantes para atraer la atención de los niños.	
RN07	Tabla de puntajes de todos los usuarios.	

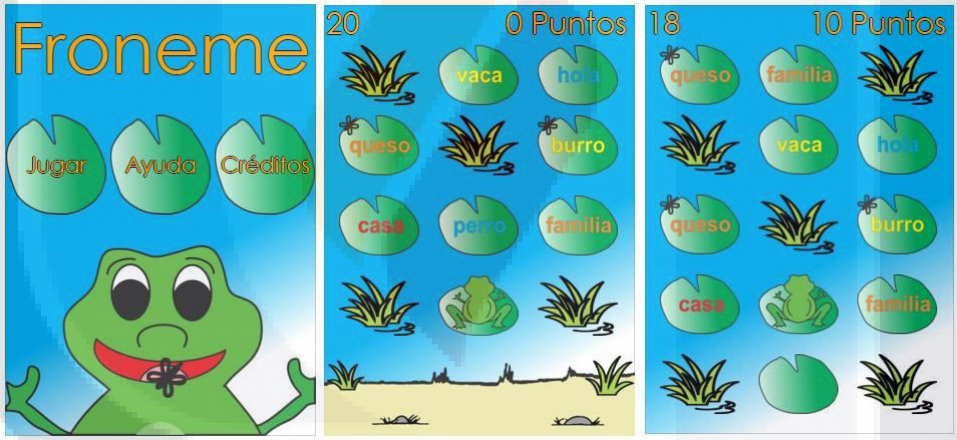
Historia

Froneme la rana está en el borde del río buscando alimento para comer. Para encontrarlo, el deberá brincar entre las hojas evitando caer en el agua. Para saltar, el jugador deberá identificar la palabra asociado a la hoja destino y pronunciarlo. Si el jugador no es capaz de pronunciarlo correctamente. Froneme caerá en el agua y por lo tanto, el juego termina.

Funcionamiento

Por cada vez que el jugador mande a Froneme a una hoja válida, 10 puntos se le agregaran al puntaje, si un mosquito está en una hoja, 10 puntos extras deberán ser agregados. El tiempo para pronunciar la palabra es de 5 segundos. Las puntuaciones y el tiempo serán guardados y reportados.

Guión Gráfico



Elementos de diseño de juego

Personajes:

Primarios: Froneme.

Secundarios: Mosquitos.

Descripción Narrativa: Froneme la rana esta en el borde del río buscando alimento para comer. Para encontrarlo, el deberá brincar entre las hojas evitando caer en el agua. Para saltar, el jugador deberá identificar la palabra asociada a la hoja destino y pronunciarlo. Si

el jugador no es capaz de pronunciarlo correctamente Froneme caerá en el agua y por lo tanto, el juego termina.

Retos: Saltar de una hoja a otra pronunciando una palabra de manera correcta.

Vistas de juego: El juego está diseñado con una vista isométrica.

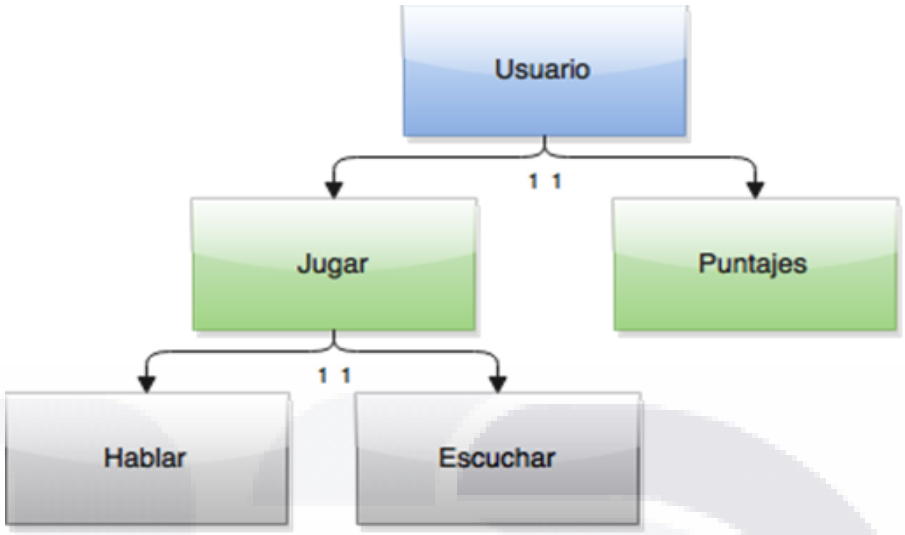
Restricciones: Escoge una hoja para saltar en ella entre una serie de opciones, saltar de una hoja a otra en un tiempo determinado.

Recompensas y castigos: Saltar de una hoja a otra dentro de un determinado intervalo de tiempo para ganar puntos o si Froneme cae en el agua el juego acaba. Si se brinca a una hoja que tenga un mosquito, se da una mayor cantidad de puntos. Por cada vez que el jugador mande a Froneme a una hoja válida, 10 puntos se le agregaran al puntaje, si un mosquito está en una hoja, 10 puntos extras deberán ser agregados. El tiempo para pronunciar el fonema es de 5 segundos. Las puntuaciones y el tiempo serán guardadas y reportadas.

Elementos interactivos: Los mosquitos representan la comida de Froneme, las hojas permiten a Froneme brincar de una hacia otra, identificadas con un fonema y por último el agua que Froneme tiene que evadir.

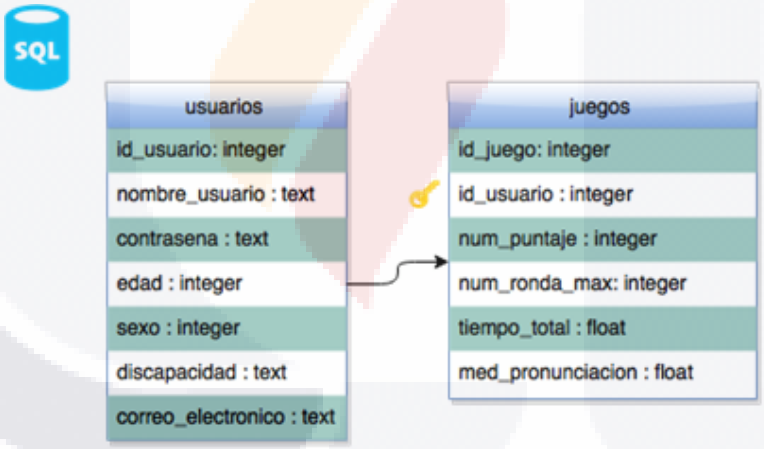
Retroalimentación: Al inicio de la ronda se activará el reconocimiento de voz y cada que acierte con la palabra.

Diagramas conceptuales



Modelado de Datos

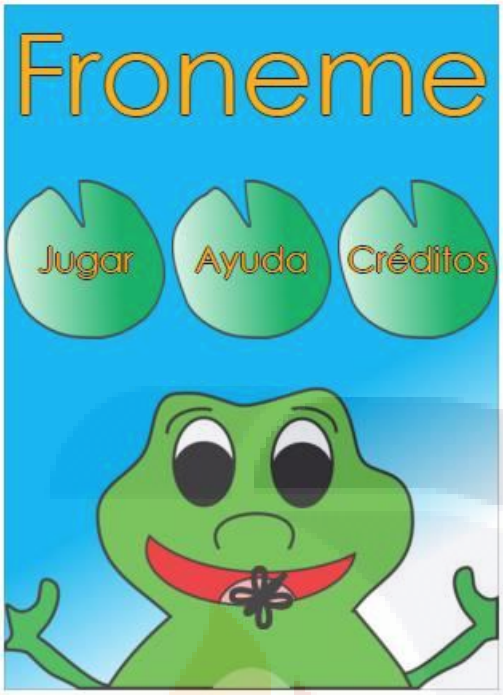
Para la información que se manejara lo más conveniente es una base de datos en la cual se registren los usuarios, los datos del juego y las palabras del diccionario.



Interfaces de Usuario

Las interfaces deberán ser intuitivas para los niños, el uso de iconos fáciles de entender es primordial para el desarrollo de estas.

El inicio solo cuenta con la imagen de fondo, el nombre del videojuego y los botones para iniciar el juego, ayuda y créditos.



El juego consiste en una interfaz que muestra toda información del estado del juego, puntaje, el tiempo que queda, las hojas con las palabras animación que muestra si se está escuchando al usuario y el botón de disparo.

Cuando Froneme cae al agua, la aplicación arroja un mensaje notificándole al usuario que ha perdido, en caso de querer comenzar de nuevo, el usuario deberá dar Aceptar a la alerta que la aplicación le muestra.

La tabla de estadísticas, se puede acceder una vez que la actividad finaliza y muestran los primeros 5 puntajes más altos con sus respectivos usuarios.

Matriz de trazabilidad

RF01	Análisis de la correcta pronunciación de la palabra.	
RF02	Reproducción de sonidos con la correcta pronunciación de las palabras.	
RF03	Contadores de tiempo.	
RF04	Acumuladores de puntaje.	
RF05	Banco de palabras que sirvan como terapia.	
RN01	Imágenes de los objetos.	
RN02	Menú inicial.	
RN03	Iniciar sesión con una cuenta personalizada del usuario.	
RN04	Animaciones de los personajes.	
RN05	Visualización del puntaje y el tiempo que le resta para pronunciar la palabra.	
RN06	Uso de colores brillantes para atraer la atención de los niños.	

RN07	Tabla de puntajes de todos los usuarios.	

Herramientas a utilizar

Lenguaje de programación

Para dar solución a los objetivos planteados y cumplir con todos los requerimientos, se llevara a cabo la programación para PC con ayuda de varios lenguajes de programación como HTML, SQL, CSS y JavaScript.

Reconocimiento de voz

El reconocimiento de voz, se llevará a cabo con el API de “Google Speech” la cual captura la voz del usuario, se digitaliza y envía al servidor central de Google que devuelve el texto reconocido en un arreglo con las posibles palabras u oraciones que se pronunció.

Sintetizador de Voz

De igual manera Google cuenta con el api para síntesis de voz compatible con distintas plataformas el cual se denomina speech.tts, donde se analiza una cadena de caracteres y genera un clip de audio.

Base de Datos

Para la base de datos se utiliza SQL, lo que nos permitirá almacenar de manera ordenada los usuarios y los puntajes de los jugadores, de manera que se pueda analizar al usuario de acuerdo a las actividades que ha realizado.

Diseño Gráfico

El diseño es una parte importante, ya que debe llamar la atención a niños y ser presentable, para ello se utiliza el programa Corel Draw para el diseño de los personajes y elementos que forman parte de la interfaz.

Definición de un estándar de programación

El desarrollo de este proyecto se llevará a cabo utilizando tecnología Web, y en específico los lenguajes de programación: HTML, SQL, PHP, CSS y JavaScript. Cada uno de ellos con diferentes finalidades y en archivos independientes. En las secciones siguientes se describen los estándares a considerar para la implementación en los lenguajes previamente mencionados.

HTML

Este lenguaje permitirá realizar la implementación de las interfaces de usuario y no deberá contener dentro de los archivos que contengan código fragmentos de otros lenguajes de programación, excepto para llamado a librerías. HTML se basa en la utilización de etiquetas que contienen palabras reservadas, las cuales son interpretadas por el navegador web, y traducidas a elementos que se despliegan en él mismo.

Los archivos que contengan código HTML del proyecto, deberán ubicarse en el directorio raíz del mismo, y tener nombres descriptivos de su contenido (como menu.html y bienvenida.html), que comiencen con letras minúsculas y que en caso de estar formados por más de una palabra se ajustarán a la notación dromedario.

Es importante mencionar que deberá existir en el proyecto un único archivo “index.html” que fungirá como pantalla principal a ser mostrada al usuario.

Con respecto a las etiquetas de HTML, éstas deben ser escritas con letras minúsculas en su totalidad y se deberá cuidar que la anidación de ellas se denote con indentación.

Finalmente, se requiere que todos los elementos que se declaren en el código HTML y cuya instanciación así lo permita, tengan asociado un identificador (atributo id en la mayoría de los casos), mismo que permitirá su manejo desde otros lenguajes de programación. Los estilos de los elementos que se definan usando este lenguaje no deberán ser definidos explícitamente in situ.

PHP

La principal función de este lenguaje en el proyecto, es proveer un mecanismo para realizar conexión con la base de datos, que permitirá almacenar datos para posteriormente poder generar estadísticas de uso. El lenguaje PHP se ejecuta del lado del

servidor, y a pesar de que en este proyecto se usará únicamente para asuntos relacionados a la base de datos, su utilidad no está restringida a esta tarea.

De la misma manera que sucede con los archivos HTML, los de PHP no deberán contener código en lenguajes distintos, con el fin de mantener siempre una estructura legible y ordenada, excepto en los casos en los que se requira agregar código SQL, llamados a procedimientos almacenados y creación de estructuras JSON. Además, éstos deberán colocarse en una carpeta llamada “PHP” dentro del directorio raíz del proyecto.

Los nombres de los ficheros PHP deberán seguir la notación dromedario, es decir, empezar siempre con minúsculas, y si el nombre está formado por más de dos palabras, cada una de las subsecuentes deberá comenzar con mayúsculas.

Deberá existir un único archivo llamado “conexión.php”, en el cuál se tenga la información y el código necesario para llevar a cabo la conexión con la base de datos. Los demás archivos que requieran hacer uso de esta característica, deberán realizar un “include” de éste.

Es importante mencionar que el indentado deberá considerarse al editar los archivos para denotar jerarquía u orden de elementos.

SQL

El lenguaje SQL será utilizado en el presente proyecto para llevar a cabo consultas y operaciones sobre la base de datos del mismo. Es deseable que las operaciones con este lenguaje se lleven a cabo a través de procedimientos almacenados, pero igualmente es admisible que estas se coloquen de manera explícita en los archivos PHP.

Las palabras clave de las sentencias SQL deberán escribirse siempre en mayúsculas.

CSS

Con la utilización de este lenguaje se tiene como objetivo definir estilos para los objetos que se presentan en la interfaz de usuario del juego serio a desarrollar, pero además jugará un papel importante en las animaciones que se crean para él mismo. La definición de hojas de estilo CSS, se lleva a cabo en archivos independientes del resto del código que se deberán colocar en una carpeta llamada “CSS” dentro del directorio raíz del proyecto. Preferentemente sólo se creará un archivo para este fin, pero en caso de ser requerido por el equipo de desarrollo, otros más podrán ser elaborados cuyos nombres deberán ser escritos en notación dromedario.

Al diseñar hojas de estilo, se pueden definir formatos en distintos niveles, para un componente en especial (por id), para una clase de componentes (por class), o para un tipo de elemento general. Para este proyecto, al definir CSS, podrán definirse estilos para los distintos niveles, partiendo siempre del principio de que el objetivo es que los estilos se difinan de la manera más general posible, dejando solo detalles a nivel de clase o id. Nuevamente, el indentado deberá ser considerado al llevar a cabo la creación de los archivos CSS.

JavaScript

JavaScript es un lenguaje que se ejecuta del lado del cliente, y es comunmente usado como parte del backend de proyectos web en los que se requiere procesamiento y validación de información. JavaScript, al ser un lenguaje orientado a eventos, permite capturar éstos con el fin de ejecutar acciones y favorecer la interacción con el usuario de manera síncrona o asíncrona.

Para este proyecto en particular, JavaScript se utilizará para administrar la interacción con el usuario, y como nexo del juego serio a desarrollar con el API de Google Speech que permite el reconocimiento de lo que el usuario dice y su procesamiento como texto.

Los archivos que contengan código en este lenguaje, deberán ser colocados en una carpeta llamada “JS” dentro del directorio raíz del proyecto. El nombre de estos archivos deberá seguir la notación dromedario y ser descriptivo del proceso que llevan a cabo.

Dentro de los archivos, los llamados a librerías deberán ser colocados en la parte superior de los mismos, posteriormente, deberá llevarse a cabo la declaración de variables globales, si las hay, anteponiendo la palabra reservada “var”, y considerando que todas, excepto las constantes deberán seguir la nomenclatura dromedario. Para el caso de las constantes, sus nombres deberán ir completamente en mayúsculas.

Las funciones que se deban declarar usando este lenguaje, serán creadas usando para su nombre la notación camello, es decir, la primera letra de cada palabra que los conforme deberá ser mayúscula, incluyendo la primera letra de su nombre.

Al igual que en los casos anteriores, el indentado deberá considerarse al estructurar el archivo y dentro de éste, no deberá existir en medida de lo posible código en otros lenguajes.

Appendix C

Context Model

CONTEXTO		
Nombre del contexto		
AMBIENTE	**PLATAFORMA**	
Tipo	Nombre de la plataforma	
Nombre del ambiente	**SOFTWARE**	**HARDWARE**
¿Es ruidoso?	Nombre del sistema operativo	Nombre del equipo
	Versión del sistema operativo	¿Permite reproducir sonido?
	Software de reconocimiento de voz	¿Permite recibir entrada de voz?
	Software de síntesis de voz	¿Permite uso de cámara?
Formato a ser llenado por el analista de requerimientos		

Static Therapy Models

CAPACIDADES	
AUDICIÓN	
OÍDO IZQ.	**OÍDO DER.**

FRECUENCIA DE RESONANCIA	UMBRAL DE AUDICIÓN	FRECUENCIA DE RESONANCIA	UMBRAL DE AUDICIÓN
Unidad de medición	Unidad de medición	Unidad de medición	Unidad de medición
Valor mínimo	Valor	Valor mínimo	Valor
Valor máximo	Frecuencia	Valor máximo	Frecuencia
	Unidad de medición de la frecuencia		Unidad de medición de la frecuencia
Formato a ser llenado por el especialista en terapia del lenguaje			

CAPACIDADES				
HABLA				
VOZ		PROSODIA		
TONO DE VOZ	**VOLUMEN**	**VELOCIDAD**	**ACENTO**	**FRASEO**
Unidad de medición	Unidad de medición	Duración de sílaba	Variación en tono	Tiempo entre palabras
Valor mínimo	Valor máximo	Palabras por segundo	Volumen	
Valor máximo		Pausa dentro de las oraciones	Duración de sílaba	
		Pausa entre oraciones	Énfasis	
Formato a ser llenado por el especialista en terapia del lenguaje				

DISCAPACIDADES			
TAREAS AFECTADAS			**DISCAPACIDAD**
TAREA AFECTADA 1	**TAREA AFECTADA 2**	**TAREA AFECTADA 3**	Tipo
Tipo	Tipo	Tipo	Nombre
Nombre	Nombre	Nombre	Detalles de la discapacidad
Objeto relacionado	Objeto relacionado	Objeto relacionado	
Detalles	Detalles	Detalles	
Nivel de falla	Nivel de falla	Nivel de falla	
Formato a ser llenado por el especialista en terapia del lenguaje			

TERAPIA TEÓRICA			
Número de sesiones			
ACTIVIDAD TEÓRICA 1	**ACTIVIDAD TEÓRICA 2**	**ACTIVIDAD TEÓRICA 3**	**ACTIVIDAD TEÓRICA 4**
Descripción	Descripción	Descripción	Descripción
Objetivo	Objetivo	Objetivo	Objetivo

Duración planeada	Duración planeada	Duración planeada	Duración planeada
Recursos físicos (Tipo y descripción)	Recursos físicos (Tipo y descripción)	Recursos físicos (Tipo y descripción)	Recursos físicos (Tipo y descripción)
Formato a ser llenado por el especialista en terapia del lenguaje			

Appendix D

SEGA-ARM: A Metamodel for the Design of Serious Games to Support Auditory Rehabilitation

David Céspedes-Hernández[1], Juan Manuel González-Calleros[3], Francisco J. Álvarez Rodríguez[1], Jaime Muñoz-Arteaga[1]

[1]Universidad Autónoma de Aguascalientes
Av. Universidad 940, Ags., Mexico
{dacespedes, fjalvar, jmunozar}@correo.uaa.mx

[2]Université catholique de Louvain
Place des Doyens 1
Louvain-la-Neuve, Belgium
jorge.perezmedina@uclouvain.be

[3]Benemérita Universidad Autónoma de Puebla
Av. 14 Sur y San Claudio, Puebla, Mexico
juan.gonzalez@cs.buap.mx

ABSTRACT

Hearing impairments are widely recognized as one of the most common disability that affects many people. There are several works addressing the design and development of systems for the rehabilitation of users with disabilities, as well as for the definition of models to express their capabilities and disabilities. Unfortunately, when we want to develop interactive systems to assist hearing impairment therapy, knowledge of analysis and design, typical of the methodologies is absent. In this paper, we introduce SEGA-ARM: a metamodel to support the design of serious games for auditory rehabilitation, considering concepts related to speech therapy sessions, user capabilities, the context of use, and concepts related to serious games design. The proposed solution is provided using a semi-formal UML notation. Two game prototypes are presented to validate the viability of the application of the metamodel for serious games development and to point out its reutilization and extension.

Categories and Subject Descriptors

H5.2 [**Information interfaces and presentation**]: User Interfaces–*Prototyping; user-centered design;* K8.0 [**Personal computing**]: Games.

General Terms

Design, Theory

Keywords

Model-driven development, speech therapy, serious games, Human-computer interaction, hearing impairment, cochlear implant, user-centered design.

1. INTRODUCTION

In several contexts such as healthcare, military and educational, games have been applied as a way to improve skills and for training professionals [1] [2]. This type of games is called serious games [3] [4]. Despite the fact that the concept of serious games itself, might be considered as an oxymoron since *"Games are inherently fun and not serious"* [5], it is important to clarify that many works had been conducted in the field addressing the development of these games, always considering fun as a main component of them [6]. Several works addressing the design and development of systems for the rehabilitation of users with disabilities [7] [8], as well as for the definition of models to express people's capabilities and disabilities [9] [10] had been carried out. However, even though hearing impairment is widely recognized as one of the most common disabilities [11], there is a lack of work focusing on providing tools to ease the design and development of applications for the domain of auditory rehabilitation. In this sense, the objective of this work is not only to provide an application that may be used to help on the performance of assisted therapy sessions, but to propose a definition of modeling elements along with their relationships and rules in order to enable the creation of semantic models, that may later result on the implementation of a family of serious games with the purpose of supporting the process of auditory rehabilitation, meaning by this, a metamodel [12] for the design of serious games to support auditory rehabilitation.

The rest of the paper is structured into sections. Section two presents a state of the art with relevant concepts to auditory rehabilitation and serious games, as well as other related works, and the description of useful techniques and tools for the development of this project. For the description of the elaborated model, section three firstly presents the packages in which it is divided. Later, in order to facilitate the comprehension of the model's classes, a case study is introduced. Finally, instancing the case study, each one of the packages' element is described. In the fourth section, a discussion on the characteristics of the full model and the relationships that exist between classes of its different packages is carried out, introducing a second case study. Last, conclusions and future work are mentioned in section number five.

2. STATE OF THE ART

2.1 Concepts on Auditory Rehabilitation

In order to better understand the aim of the present work, a set of concepts relative to auditory rehabilitation and serious games has to be defined.

Rehabilitation is the dynamic process by which physical, sensory, and mental capacities are restored or developed in people with disabling conditions [13]. When it comes to rehabilitation for auditory disabilities, it is covered mainly by speech therapy and reinforced by occupational and physical therapy. The success of a rehabilitation program depends on various factors: timing, patient diagnosis and treatment planning [14]. For the purposes of this

particular work, only speech therapy is addressed. Speech therapy, also known as speech-language pathology, is the rehabilitative treatment of physical and/or cognitive disorders resulting in difficulty to perform verbal communication. Speech therapists asses, diagnose, treat and help to prevent disorders related to speech, language and other elements of communication [15]. The ultimate goal of these specialists is to help patients develop or recover reliable communication and other skills. Most speech-language pathologists work with determined age groups, such as children or elderly and focus on the treatment of certain communication problems, such as those resulting from strokes or hearing loss [16].

Hearing impairment may be caused by different factors, such as, accidents, viral diseases, noise, and genetic causes [17]. Depending on the background of the patients with hearing impairment, among other classifications, they can be categorized into two groups: prelocutive (loss hearing toke place before the acquisition of spoken language) and postlocutive (patients whose loss of hearing has taken place after the development of the basic skills of spoken language) [18]. Once a hearing impairment has been diagnosed, the specialists involved on the patient's treatment, evaluate the best option in order to improve his/her conditions. In this sense, some of the patients may be candidates to receive a cochlear implant as the one mentioned in the case study (Subsection 3.1) of the present work.

A cochlear implant [19] is an electronic device that stimulates the auditory nerve to allow sound perception. It is composed by an external component that receives incoming sound, processes it and transfers the resultant signal across the skin. Another device receives and decodes the transmitted signal and stimulates the auditory nerve directly, bypassing the hair cells that implement the first stage of auditory neural processing in normal ears. According to the National Institute on Deafness and Other Communication Disorders (NIDCD) on the United States of America, by December 2012, approximately 324,200 people worldwide have received cochlear implants [20].

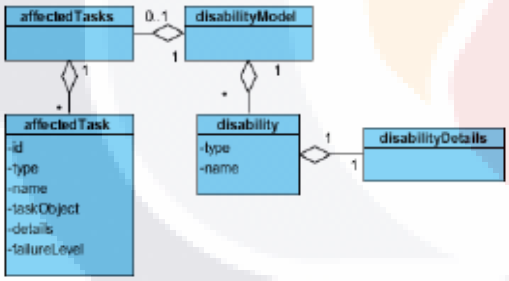


Figure 1. The disability model proposed by Kaklanis [10].

2.2 Concepts on Serious Games

To define a serious game, [21] begins by defining a game as a physical or mental challenge, that is played according to specific rules, with the objective of amusing or entertaining the participant. Then, continues to build the definition of a videogame as a mental challenge that is played through a computer according to certain rules for entertainment, fun or to achieve a goal. And finally, provides the definition of a serious game as a mental challenge, played through a computer according to specific rules, which uses entertainment mechanisms to achieve training or capacitation goals in areas such as education, health and military.

Notice that in the rest of the paper, this is taken as the definition of a serious game, just pointing out that it may also consist of physical challenges (e.g., tongue movement, phoneme pronunciation, and lips movement). The interest on the usage of serious games on rehabilitation lies on the fact that is has been demonstrated that they increase motivation towards therapy sessions, which represents a major problem due to the repetitive nature of exercises [14].

2.3 Related Work

Software designers may achieve the goal of expressing the capabilities and disabilities, along with other valuable insights of a user to developers via "*Personas*" models [9]. "*Personas*" are hypothetical archetypes of actual users and their description may consist on different aspects according to what is intended to make the development team aware of. Alternately, in [10] a metamodel was proposed by Kaklanis to allow the expression of the whole user capabilities and disabilities with the intention of virtual user modeling. The main advantage that was encountered while comparing the "*Personas*" representation to this second proposal, is that "*Personas*" provide only a natural language description of the characteristics of the user, while Kaklanis' disability and capability models' structure allows their representation on a machine-readable format.

For the present work, the disability model by Kaklanis, shown in Figure 1, is considered as it was defined, since it provides a generic description of any kind of disabilities with the affected tasks on it involved, while the capability model was in turn simplified to extract from it only those parts related to the hearing and speech skills, and enriched with characteristics that authors mention as relevant for the domain of speech therapy in children, such as loudness, speech rate and phrasing [22][23].

Concerning serious games design, an effort has been realized by several authors, in order to abstract components that lead to the definition of a guide on what to be aware of when developing a game with these characteristics [24][25]. However there is not yet a consensus on the components or elements that should be considered when performing design and implementation, Table 1 presents some of the design elements that are usually taken into account for the development of games [24-28].

Moreover, Longstreet & Cooper [29] proposed a conceptual guide for development of serious games to support higher education, their conceptual definition includes most of the design elements reported on Table 1, but a structural notation was not provided and the relationship between the involved elements is only explained in a narrative manner, making its reutilization complicated.

In this section, a general panorama of the domain concepts was provided in order to facilitate the comprehension of further sections. Also, related works are considered and described, pointing out their usefulness for accomplishing the objectives presented on section one, and also reporting their limitations.

Finally, taking into account the concepts addressed in the last two subsections, a decision was made to consider the proposition of a metamodel composed by three main packages: context, theoretical therapy, and performed therapy, and to add it the definition of a serious games design elements extension.

Table 1. Game design elements in the literature [24-28].

Game Design Element	Description
Characters	The definition of characters allows user engagement to the game, and provides the possibility to define actions. Helps improving the player experience. Considers both, player and non-player characters
Narrative description	The rational of the game described in a narrative way, promotes the immersion of the player, helps to define the needed resources, characters, game mechanics and challenges.
Challenges	The definition of challenges to be faced during the gameplay, gives the player the opportunity to try his/her skills and to compete either with a partner or with the game itself.
Type of view	Determines the level of immersion that is decided to use in the game. The most used point-of-view perspectives for gameplay are: first-person, third-person, third-person trailing, overhead, and three-fourths isometric.
Constraints	Constraints determine characteristics of the challenges (e.g., limited time or resources, competitive and sequential activities).
Rewards	Establishing a reward system (e.g., points, badges, levels, and leaderboards) allows improving the player experience and motivating him/her to achieve a goal.
Punishments	As with rewards, the establishment of punishments, such as a decrement of points, and losing a life in the game, motivates the user to achieve a determined goal.
Interactive elements	The definition of interactive elements supports the immersion of the player on a virtual world. Their definition helps to design player tasks.
Feedback	For each action performed by the player, the game provides a reaction. In this way, the user is aware of the consequences of his/her acts on the game and an appearance of continuous dialog is given to him/her.

3. SEGA-ARM: A METAMODEL FOR THE DESIGN OF SERIOUS GAMES TO SUPPORT AUDITORY REHABILITATION

For achieving the objectives mentioned on the introduction, a metamodel (SEGA-ARM) is proposed, considering for its definition concepts related to context, auditory rehabilitation, planned therapy, performed therapy, and serious games design elements. The elaborated model is divided for its best comprehension and usage into three packages, presented in Figure 3, and an extension shown in Figure 4, each one of them identified by a different color and name, and designed to contain elements that support the design of serious games for auditory rehabilitation.

In the rest of this section, SEGA-ARM is described in detail and explained through a case study of a serious game design for supporting the process of rehabilitation of a patient with a deep hypoacusia condition, recipient of a cochlear implant.

3.1 Case Study

The case study consists on a scenario in which the design of a serious game is required in order to support the therapy sessions conducted by a speech therapist for a six year old male patient diagnosed with deep bilateral hypoacusia who has recently received a cochlear implant. The therapist has already performed tests over the patient to determine his exact condition and counts on a full expedient describing his capabilities towards hearing and speech skills. The intention with the required game is to extend and complement the therapy sessions that the patient already receives with sessions conducted by himself in his home as those suggested in [30]. The language therapist wants the game to reinforce one of the therapy activities that the patient finds to be repetitive since it has to be performed in almost every single session.

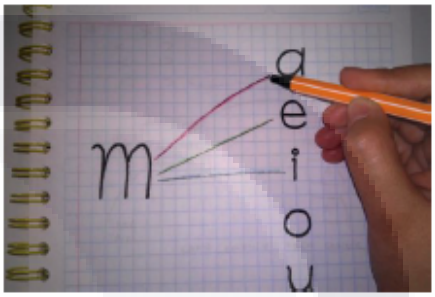


Figure 2. The exercise conducted in the therapist's office.

The exercise consists on that the therapist chooses a series of phonemes sharing a consonant (e.g., ma, me, mi, mo, and mu), write them down on a notebook as shown in Figure 2, putting the consonant on one column of the page and the vowels in another next to it, then asks the patient to pronounce each one properly while connecting the consonant with the vowels with lines, and then repeat this task several times and with different phonemes. This activity is designed to be performed in approximately ten minutes and also considers phoneme visual recognition by the patient. It is also known by the therapist that the patient counts on a 10.1" tablet with Android 4.2 OS. The information that the specialist possesses has been structured in terms of the here proposed metamodel as a way to facilitate the communication between the speech therapist and the game development team. The following section reports as an instantiation of the metamodel's classes and how data was arranged and organized.

3.2 Description of the Classes Involved on the Model

The following subsections are for used for providing a detailed description of the packages and classes that take part on SEGA-ARM. For every single package that is described, there is an instantiation in terms of our case study introduced on section 3.1.

3.2.1 The Context Package

A package designed to understand the *Context* [31] in which the user interacts with the system, and to provide the developers with a brief description of relevant characteristics to make decisions about what interactive modality to implement, and resources selection among others. The *Context* is identified by an id and a descriptive name, and conformed by an *Environment*, a *Platform* and a user (*Patient*).

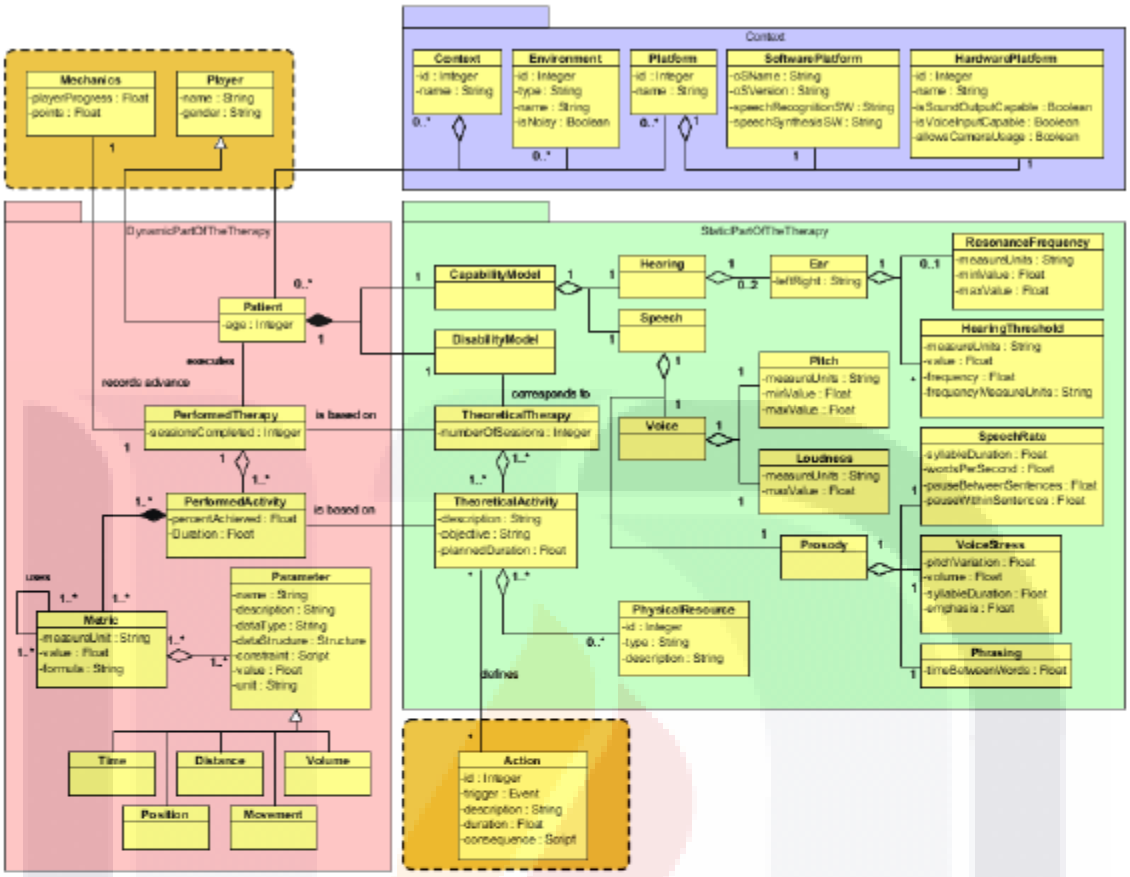


Figure 3. The Context, Theoretical part of the Therapy, and Dynamical part of the Therapy packages with their relationships to the serious game extension of the SEGA-ARM metamodel proposed.

The *Environment* describes the physical place in which the interaction occurs. For the specific domain of this work, it is necessary to point out if it is noisy or not, and to give it a name, an id and a type (i.e., exterior or interior).

The *Platform* definition is carried out to determine the characteristics of the device that is going to host the application, and is formed by a *SoftwarePlatform* and a *HardwarePlatform*. The *SoftwarePlatform* represents those characteristics relative to the software in the hosting platform i.e. operating system name and version, as well as the speech recognition/synthesis software on it installed, while the *HardwarePlatform* class defines characteristics regarding the hardware of the hosting platform and relevant to the domain, such as the possibility of using sound outputs, voice inputs or a camera.

In our example, the context in which the therapy sessions take place is defined by a therapist's office as a non-noisy, interior environment, has not platform and the user role on it is taken by the patient, meanwhile, the context of use of the application that is intended to be developed, is composed by the patient's home as non-noisy interior environment, a 10.1" tablet with Android Jelly Bean 4.2 operating system, audio output, voice input and camera usage enabled, with android speech recognition/synthesis software as platform, and the patient as user. Note that even though the characteristics of the context must be taken into account by the serious game's designer; the implemented solution may work under other similar contexts as well.

3.2.2 *The Static Part of the Therapy Package*

The purpose that is pursued with the design of this package is to provide the speech therapist with a semiformal notation to express the characteristics of a therapy plan, keeping that information structured in a way that results understandable for serious games designers. The definition of this part of the model was inspired mainly by the work done by Kaklanis for modeling a Virtual User [10], extended to allow the expression of a full auditory therapy session program, and enriched with concepts found out to be relevant for patients of auditory rehabilitation on the literature [22] [23] and through the attendance to speech therapy sessions. Abstracted from the full capability model [10] that describes physical, cognitive and behavioral user characteristics, for the specific domain of this work, the *CapabilityModel* presented as part of this paper, only considers the elements related to speech and hearing.

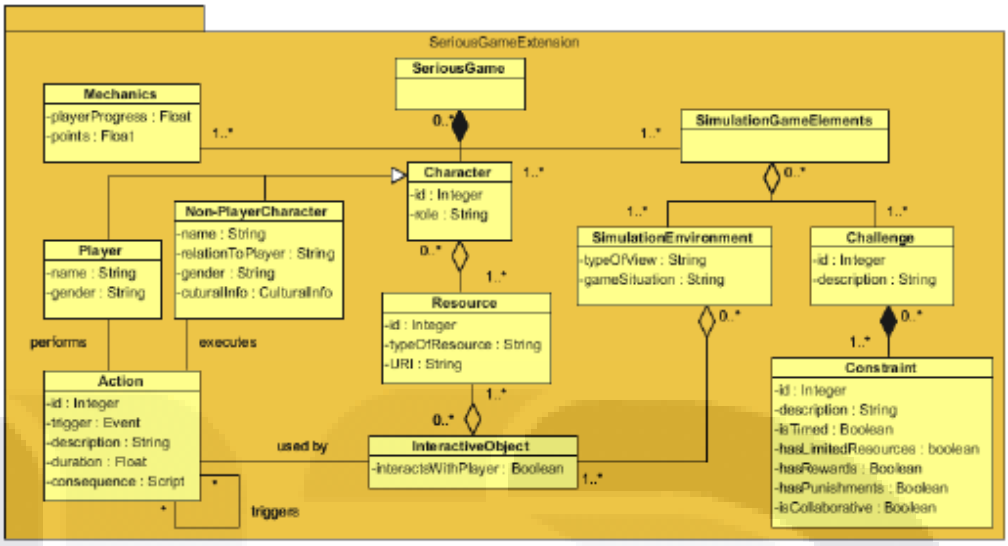


Figure 4. The serious games design elements extension provided for SEGA-ARM

The only parameter in the *Hearing* container is the *Ear* that in turn acts as one for the values related to each ear of the user: *ResonanceFrequency* and *HearingThreshold*. The *Speech* container includes *Voice* and *Prosody* elements. *Voice* elements include *Pitch* and *Loudness*, each one with its measure units and values; while *Prosody* is formed by *SpeechRate*, *VoiceStress* and *Phrasing*.

A *DisabilityModel* [10], allows the description of all the disabilities of the user as well as the affected by them tasks. For each disability, a name and a type (e.g. motor, auditory, and vocal) must be provided, and it is related to affected tasks. For a determined *DisabilityModel*, a *TheoreticalTherapy* may be proposed. A *TheoreticalTherapy* is a full rehabilitation plan and consists of a certain number of sessions and a series of theoretical activities. A *TheoreticalActivity* is in turn, a series of actions with a common objective to be performed by the patient either using or not *PhysicalResources* (e.g., a mirror, sticks or cards). The planned duration, objective and description are attributes of a *TheoreticalActivity*.

For our case study, the disability and capability models correspond to the diagnosis that was realized by the specialist in speech therapy. The disability that was encountered is deep hypoacusia, and the affected tasks by it, are hearing and speaking. For the capability model, in the hearing branch, the patient presents hearing thresholds of more than 90 dB nHL for frequencies of 1000, 2,000 and 4,000Hz in both ears; and a resonance frequency between 800 and 1,200Hz. For the speech branch, the patient's voice was measured, getting as values for the pitch 250-300Hz, and 73dB for the maximum loudness. The prosody was also analyzed, obtaining for it typical results for a patient of the age and condition as the previously described. Corresponding to the identified disability, a rehabilitation program was designed comprising approximately 300 hours, to be provided in half an hour, two times a week sessions. One of the activities to be performed by the patient consists on repeating a series of phonemes in a determined order. The therapist mentions a phoneme and expects the patient to repeat it. If a phoneme is not correctly pronounced, the therapist reinforces its pronunciation and motivates the user to try again. The objective of the task is to improve the patient's pronunciation and to get him/her used to repeat the sounds that he/she listens to. For this particular activity, only a notebook and a color pencils are used.

3.2.3 The Dynamic Part of the Therapy Package

In order to represent the current progress and performance of the *Patient* towards the therapy program, and to provide the development team with a structure to understand how to evaluate patient's activities, the dynamic part of the therapy package was created. This package contains then the *Patient* class, with an age attribute, related to a *PerformedTherapy* which reports the number of completed sessions and groups objects of type *PerformedActivity*.

A *PerformedActivity* is based on a *TheoreticalActivity* and its main goal is to keep a record of the advance presented by the patient while performing a given activity. In order to allow the evaluation of a *PerformedActivity*, a *Metric* class was also defined. Metrics allow the use of a formula in order to compute a value and to express it on a determined measure unit (e.g., centimeters, seconds and decibels). For the evaluation of a formula, a series of *Parameters* i.e., criteria to be used while evaluating, may be defined. There could be nested metrics since some formulas may need some others to be previously computed in order to take values from them. Metrics in this domain may be useful for measuring among others, tongue movement (i.e., tongue tip position, tongue vibration, and tongue position), lips movement (upper lip movement and lower lip movement), pronounced phonemes and blow action.

In the context of the case study, the performed therapy and performed activity classes keep track on how the patient is evolving and advancing through the therapy program and their involved values are constantly updated. The metric to be used for the specific theoretical activity described in section 3.2.2, consists

on evaluating how a phoneme is pronounced by the patient, recognize it and compare it to an expected phoneme passed as parameter according to a specific tolerance, using for this purpose the formula represented on Figure 5 where the pronounced by the patient phoneme (pp) is compared to the expected phoneme (ep) and then, if the *distance* between them is smaller than the tolerance value (t), the utterance from the patient is accepted as correct.

$$Recognition = distance(ep, pp) - t$$

Figure 5. Formula for recognition considering expected phoneme (ep), utterance (pp), and a tolerance value (t).

3.2.4 The Serious Games Extension

In attention to the main objective of this paper, an extension to the therapy metamodel for considering serious games design elements was elaborated. For the proposal of this extension, mainly Longstreet's approach is considered [29], enriched with concepts obtained through a deeper survey on serious games design elements [26-28]. The intention on the definition of this extension is to provide developers with a tool for passing from a narrative description of a game to a semiformal structure that may be related to the one of the core metamodel. A *SeriousGame* is composed by *Mechanics* to record the player progress and to manage the punctuations, one or many instances of the *Character* class to define players as well as non-player characters (*NPC*), and *SimulationGameElements* such challenges, rules, scenarios and interactive objects.

A *Character* object represents an entity that realizes actions. These instances can be either a *Player* or a *Non-PlayerCharacter*, and need *Resources* in order to be properly displayed. For the definition of *NPCs*, it is necessary to give them a name, decide the relationship that they maintain towards the *Player*, and when possible, provide their gender and cultural information e.g. language, nationality, degree among others [29]. In order to handle events on the game and describe their consequences, it was necessary to define the *Action* class. An *Action* is represented by an event (e.g. key press, click, tap, recognized word or movement, or scripts) that triggers it, along with a description, duration, and a script to execute as consequence. Actions are performed by *Players* or executed by *NPCs* and may whether use or not an *InteractiveObject* that are in turn, objects that belong to the *SimulationEnvironment* and are created to support the game interaction, for instance, the scenario, the floor, and colliding objects. The *SimulationEnvironment* description contemplates the definition of a type of view (e.g., first person, third person, or isometric) [26], and helps to hold the game situation i.e. level, state, and maximum amount of errors [27] in a narrative manner. Finally for this package, a series of challenges has to be designed to represent the goals that are pursued while playing the game. A *Challenge* consists of a description and is composed by at least one *Constraint*. Constrains may include one or several conditions (viz. time, limited resources, rewards, punishments, and mandatory collaboration).

For the given case study, taking into account the description provided by the speech therapist for the context, theoretical and dynamic packages, the serious game designer proposes to elaborate a game with the characteristics that are presented in Table 2 and which resultant user interface is shown in Figure 6.

It is important to point out that the definition of some of the game design elements was achieved through the establishment of relationships between the therapy packages and the serious games extension (e.g. *TheoreticalActivity* and the *Action*). Next section emphasizes this characteristic of SEGA-ARM along with others that had not been yet addressed.

Table 2. Characteristics of the designed serious game.

Game name	Froneme the frog.
Narrative description	Froneme the frog is in the edge of a river looking for food to eat. In order to go find it, he has to jump between leaves avoiding to fall on the water. To perform a jump, the player has to identify the phoneme associated to the destination leaf and pronounce it. If the player is unable to provide a correct utterance, Froneme falls down into the water and therefore the game is over.
Player	The patient represented by a frog.
Interactive objects	Mosquitoes representing Froneme's food; leaves to allow Froneme jump from one to another identified by a phoneme; water that has to be avoided by Froneme.
Player actions	Eat mosquitoes by landing on the leaves that they are on, jump from one leaf to another by pronouncing phonemes, and fall down on the water if not correct utterance is provided on the given time interval.
Simulation environment	The game is designed with an isometric type of view, and the following game situations: Froneme in the border of the river (initial state of the game), Froneme on a free leaf, Froneme on a leaf occupied by a mosquito (eating), and Froneme falling on the water.
Challenges	The challenges involved in the game are the following: jump from a leaf to another pronouncing a correct phoneme, jump from a leaf to another within a determined time interval (time limit) to earn points (reward), or let Froneme to fall into the water (punishment); choose a leaf to jump on from a series of provided options (limited resources) and if there is a mosquito on the selected one, earn a bigger amount of points (reward).
Mechanics	For each time the player takes Froneme to a valid leaf, 10 points are going to be added to the budget, If there is a mosquito on the occupied leaf, ten extra points should be added. The time available for producing the phoneme is five seconds. Time and punctuations are recorded and reported.

4. DISCUSSION

As it may be seen in Figure 3, there are relationships between the therapy packages and some of the serious games design elements extension classes. These connections allow the developers to know where to extract information that is supposed to be represented in the game, and how to present it. Punctually, there are relationships between: the *Patient* (Dynamic part of the therapy) and the *Player* (Serious games) to remark the fact that the

patient has to be considered as the player and to get some information from his/her profile; the *PerformedTherapy* (Dynamic part of the therapy) and the *Mechanics* (Serious games), to track the advance of the player through the game and give a feedback for the therapist; and the *TheoreticalActivity* (Theoretical part of the therapy) and the *Action* (Serious games) in order to understand the tasks that the patient is supposed to perform while playing the serious game. It is also important to point out that three levels were considered while modeling the therapy: (1) the patient (capabilities and disabilities), (2) the therapy program (theoretical therapy), and (3) the involved activities to meet the plan (theoretical activity). Each one of those three levels has equivalence on the dynamic part of the therapy in order to keep track of the advance and performance of the patient through the therapy.

When compared to the proposal in [29], our metamodel has the advantage of providing a semiformal notation which may be used in a methodological process as a conceptual guide for achieving the goal of expressing and therefore implementing serious games for a specific set of therapy activities. Along with its description, it also allows identifying the actor that is supposed to fulfill the different classes and attributes, and provides a definition of the relationships that exist among them. However, in this paper the use of knowledge bases and taxonomies is not considered resulting on the lack of a repository of activities to provide to the developers.

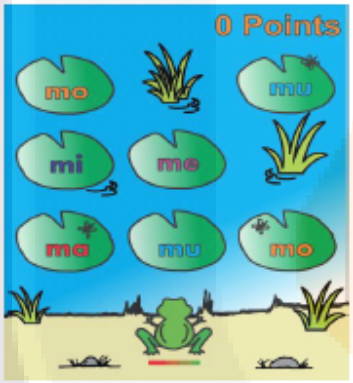


Figure 6. Screenshot of the serious game Froneme the frog.

One of the aimed advantages of our proposal is reutilization, and in order to demonstrate it, a second instantiation of serious game, called Roberto the robot, was realized in which the very same definitions of the case study described in section 3 for the context and for the static and dynamic parts of the therapy packages are taken into account, varying only the serious game extension elements and therefore creating a different serious game but for the same rehabilitation purpose. For this second game, a character representing the player called Roberto the robot is set on a scenario representing the moon surface looking for pieces to assemble a rocket to return to the Earth. In order to find them, he has to run and jump over rocks avoiding hitting them. To perform a jump, the player has to pronounce a required phoneme. If the player is unable to provide a correct utterance, Roberto hits a rock and finally after 5 impacts falls down. The interactive objects are: small rockets representing rocket pieces, and rocks that have to be avoided by the player. The player actions are: to take rocket pieces by reaching them, jump over rocks, hit rocks if no correct utterance is provided on a given interval. The game was designed with a third person type of view, and the following game situations: Roberto on the surface of the moon, Roberto jumping over a rock, and Roberto reaching a rocket piece. The challenges involved in the game are the following: jump over rocks pronouncing a correct phoneme (limited resources) within a determined time interval (time limit) to earn points (reward), or hit rocks and fall (punishment); and if a rocket piece is reached earn a bigger amount of points (reward). As mechanics of the game, for each time Roberto jumps a rock, 10 points are added to the budget, if a rocket piece is reached, ten extra points are added. The time available for producing the phoneme is five seconds. On the development of this game, shown in Figure 7, also interface elements from the Froneme the Frog game were reused. There are also works on the definition of rules for defining different user interfaces as output [32] but those objectives are out of the aim of this paper.

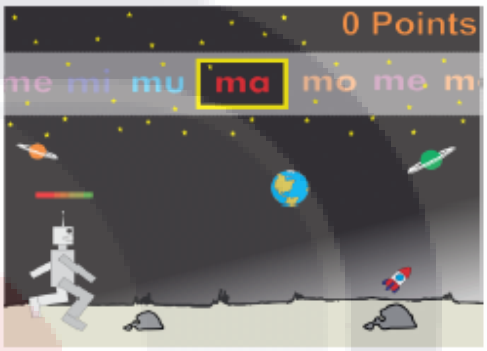


Figure 7. Screenshot of the serious game Roberto the robot.

It is important to note as another advantage of the here proposed metamodel that since the packages included on it keep a certain level of independence towards the serious games design extension, it is possible to use them, and to define new extensions in order to design and develop other types of software solutions for auditory rehabilitation (e.g., diagnosis, tracking, and advising systems). Likewise, by developing models for other domains than auditory rehabilitation such as educational and military, it may be possible to relate them to the extension here provided to enable the development of serious games for those ambits.

5. CONCLUSIONS AND FUTURE WORK

In this paper we proposed SEGA-ARM, a metamodel for the design of serious games to support auditory rehabilitation. The mentioned model consists of three packages, regarding the context of use, the theoretical part of the therapy, and the dynamic part of the therapy; and an extension for its implementation through serious games. As the description of every single package and class was provided, the actors on it involved (i.e., designer, development team, and speech therapist) were mentioned along with tasks to be performed by them, and a case study was in it instanced to allow a better understanding. Two prototypes of serious game to support a therapy activity were developed. In the near future the developed serious games are going to be evaluated by a speech therapist in order to determine if it is possible to include them at first as support tools for in-consulting-room therapy sessions, and then as a way to complement and extend the received therapy sessions with sessions conducted by the patient

himself. Also, more extensions are going to be developed in order to verify the completeness of the metamodel for different scopes.

6. ACKNOWLEDGEMENTS

We acknowledge the support that was given by CONACYT, The Université catholique de Louvain and the Autonomous University of Aguascalientes for the development of this work through a research stay program. We also acknowledge Professor Jean Vanderdonckt from Université catholique de Louvain for his valuable support on the development of this project.

7. REFERENCES

[1] Amab, S., Durnwell, I., & Debattista, K. (2013). Serious games for healthcare: Applications and implications. Medical Information Science Reference.

[2] McDowell, P., Darken, R., Sullivan, J., & Johnson, E. (2006). Delta3D: a complete open source game and simulation engine for building military training systems. The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology, 3(3), 143-154.

[3] Michael, D. R., & Chen, S. L. (2005). Serious games: Games that educate, train, and inform. Muska & Lipman/Premier-Trade.

[4] Breuer, J. S., & Bente, G. (2010). Why so serious? On the relation of serious games and learning. Eludamos. Journal for Computer Game Culture, 4(1), 7-24.

[5] Kuipers, D. A., et al. (2013). Design for Transfer. In Serious Games Development and Applications (pp. 239-246). Springer Berlin Heidelberg.

[6] Ritterfeld, U., Cody, M., & Vorderer, P. (Eds.). (2010). Serious games: Mechanisms and effects. Routledge.

[7] Kikuchi, T., et al. (2007, June). Quasi-3-D OF rehabilitation system for upper limbs: its force-feedback mechanism and software for rehabilitation. In Rehabilitation Robotics, 2007. ICORR 2007. IEEE 10th International Conference on (pp. 24-27). IEEE.

[8] Burdea, G., Popescu, V., Hentz, V., & Colbert, K. (2000). Virtual reality-based orthopedic telerehabilitation. Rehabilitation Engineering, IEEE Transactions on, 8(3), 430-432.

[9] Casas, Roberto, et al. User modelling in ambient intelligence for elderly and disabled people. Springer Berlin Heidelberg, 2008.

[10] Kaklanis, N., A Coupled User and Task Modelling Methodology for Accessible Product Design. PhD dissertation, University of Surrey. March 2013.

[11] Ventry, I. M., & Weinstein, B. E. (1982). The hearing handicap inventory for the elderly: a new tool. Ear and hearing, 3(3), 128-134.

[12] Koch, N., & Kraus, A. (2003). Towards a common metamodel for the development of web applications. In Web Engineering (pp. 497-506). Springer Berlin Heidelberg.

[13] Brandt Jr, E. N., & Pope, A. M. (Eds.). (1997). Enabling America: Assessing the Role of Rehabilitation Science and Engineering. National Academies Press.

[14] Rego, P., Moreira, P. M., & Reis, L. P. (2010, June). Serious games for rehabilitation: A survey and a classification towards a taxonomy. In Information Systems and Technologies (CISTI), 2010 (pp. 1-6). IEEE.

[15] McDonnell, A., & Akruvala, S. D. (2012). Rehabilitative Therapists in Geriatric Health Services. Managing Geriatric Health Services, 45.

[16] Bureau of Labor Statistics, U.S. Department of Labor, Occupational Outlook Handbook, 2014-15 Edition, Speech-Language Pathologists, on the Internet at http://www.bls.gov/ooh/healthcare/speech-language-pathologists.htm (visited March, 2015).

[17] World Health Organization, Fact sheet No.300 Updated March 2015, on the Internet at http://www.who.int/mediacentre/factsheets/fs300/en/ (visited March, 2015).

[18] Pereira, A. (2010). Criteria for elaborating subtitles for deaf and hard of hearing adults in Spain: Description of a case study. Listening to Subtitles. Subtitles for the Deaf and Hard of Hearing, 87-102.

[19] Svirsky, M. A., et al. (2000). Language development in profoundly deaf children with cochlear implants. Psychological science, 11(2), 153-158.

[20] National Institute on Deafness and Other Communication Disorders, U.S. Department of Health & Human Services. Quick Statistics. On the Internet http://www.nidcd.nih.gov/health/statistics/pages/quick.aspx (visited February, 2015).

[21] Zyda, M. (2005). From visual simulation to virtual reality to games. Computer, 38(9), 25-32.

[22] Chin, S. B., Bergeson, T. R., & Phan, J. (2012). Speech intelligibility and prosody production in children with cochlear implants. Journal of communication disorders, 45(5), 355-366.

[23] Lenden, J. M., & Flipsen, P. (2007). Prosody and voice characteristics of children with cochlear implants. Journal of Communication Disorders, 40(1), 66-81.

[24] Aldrich, C. (2009). The complete guide to simulations and serious games: how the most valuable content will be created in the age beyond Gutenberg to Google. John Wiley & Sons.

[25] Deterding, S., et al. (2011, September). From game design elements to gamefulness: defining gamification. In Proceedings of the 15th International Academic Mind Trek Conference: Envisioning Future Media Environments (pp. 9-15). ACM.

[26] Taylor, L. N. (2002). Video games: Perspective, point-of-view, and immersion (Doctoral dissertation, University of Florida).

[27] Salen, K., & Zimmerman, E. (2004). Rules of play: Game design fundamentals. MIT press.

[28] Sánchez, J. L. G., Zea, N. P., & Gutiérrez, F. L. (2009). From usability to playability: Introduction to player-centred video game development process. In Human Centered Design (pp. 65-74). Springer Berlin Heidelberg.

[29] Longstreet, C. S., & Cooper, K. (2012, July). A meta-model for developing simulation games in higher education and professional development training. In Computer Games (CGAMES), 2012 17th International Conference on (pp. 39-44). IEEE.

[30] Ittleman, M. A. (2012). The Teaching of Talking: Learn to Do Expert Speech Therapy at Home with Children and Adults. Morgan James Publishing.

[31] Calvary, G., et al. (2003). A unifying reference framework for multi-target user interfaces. Interacting with Computers, 15(3), 289-308.

[32] Florins, M., et al. (2006). Splitting rules for graceful degradation of user interfaces. In Proceedings of the working conference on Advanced visual interfaces (pp. 59-66). ACM.

Diseño de un Videojuego para Terapia de Lenguaje en Pacientes con Hipoacusia Bilateral Profunda con Implante Coclear

David Céspedes-Hernández[1], Francisco J. Álvarez Rodríguez[1], Jaime Muñoz-Arteaga[1], Juan Manuel González-Calleros[2], Liliana Rodríguez-Vizzuett[1]

[1]Universidad Autónoma de Aguascalientes, Centro de Ciencias Básicas, México
[2]Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación, México
{dcespedesh, fjalvar.uaa, jmauaa, lilianarv90}@gmail.com,
juan.gonzalez@cs.buap.mx

Abstract. Los videojuegos serios han sido usados en múltiples contextos como el educativo, el militar y el de la salud. Se encuentra la necesidad de elaborar videojuegos para soportar el proceso de rehabilitación del lenguaje. Para esto, se propone un proceso para desarrollo que considera actividades realizadas en una sesión de terapia y se sigue una metodología para generar modelos que sirvan para el desarrollo de juegos en dispositivos móviles que permitan extender las terapias y favorecer el proceso de rehabilitación. Se siguen guías durante la implementación para garantizar la usabilidad del sistema. El proceso de desarrollo se ilustra tomando como base una actividad propuesta por un especialista en terapia de lenguaje para pacientes diagnosticados con hipoacusia bilateral profunda con implante coclear, pero la naturaleza del desarrollo del trabajo permite extender la solución a distintos contextos y a casos particulares.

Keywords: Videojuegos serios, rehabilitación, terapia del lenguaje, desarrollo basado en modelos, interacción humano-computadora, hipoacusia.

1 Introducción

Los videojuegos serios se definen como juegos soportados tecnológicamente cuyo objetivo principal no es el de entretener al usuario. Esta clase de juegos han sido utilizados y estudiados ampliamente en múltiples contextos [1][2][3]. La rehabilitación es un proceso dinámico adaptativo que se lleva a cabo con el objetivo de cambiar las

condiciones de vida impuestas a un individuo por enfermedades o accidentes [4]. En este trabajo, se trata de manera puntual la rehabilitación en cuanto a la comunicación se refiere, es decir, terapia del lenguaje. Los desórdenes que la terapia del lenguaje atiende van desde la sustitución de sonidos hasta la discapacidad para entender o usar el lenguaje. Los especialistas en rehabilitación del lenguaje llevan a cabo prevención, diagnóstico y tratamiento de dichos padecimientos [5].

Al consultar la literatura existente, se nota falta de desarrollo de aplicaciones para favorecer la rehabilitación conforme al conocimiento de especialistas y específicas a un padecimiento particular, así como la necesidad de complementar las sesiones de terapia llevadas a cabo por el especialista con sesiones en casa conducidas por el paciente [6]. Se identifica así, el requerimiento de Interfaces de Usuario (UI) altamente interactivas y usables y se propone seguir un enfoque basado en modelos para desarrollo de un videojuego serio cuyo objetivo principal es permitir a pacientes de terapia de lenguaje extender las sesiones que reciben y poder efectuarlas en múltiples contextos. El proceso propuesto en este trabajo comprende, el análisis de las actividades realizadas en la terapia convencional por el especialista y el paciente, el planteamiento de dichas actividades en términos de modelos y su evolución para llegar a diseñar con estos UI finales así como la utilización de guías de usabilidad. En la sección dos se presentan antecedentes y proyectos similares encontrados en la literatura. La sección tres, contiene el procedimiento propuesto para realizar el análisis de las tareas que son parte de una actividad propuesta por un especialista, y con base en él se realiza el diseño de modelos que dan como resultado interfaces finales para el prototipo del juego serio que se desea implementar. Posteriormente se presenta la evaluación del prototipo elaborado en el presente trabajo por parte del especialista en terapia del lenguaje. Finalmente se presentan conclusiones y el trabajo futuro en términos de los objetivos alcanzados y de los que quedan por alcanzar.

2 Estado del Arte

En la literatura, pueden encontrarse trabajos en los que se clasifican y describen sistemas para rehabilitación física [7], mientras que en otros se analiza la inclusión de juegos serios en terapia para discapacidades cognitivas [8]. En cuanto a la terapia del lenguaje, existen trabajos en los que se evalúa el uso de videojuegos como parte de las sesiones de terapia reportando resultados positivos [9]. Sin embargo, se puede observar al llevar a cabo una revisión de las listas de aplicaciones disponibles para móviles, con

el objetivo de soportar la rehabilitación del lenguaje, que la mayoría no cuenta con el soporte de especialistas, no atienden un padecimiento de manera específica y no se reporta haber seguido para su desarrollo un método que avale que el sistema implementado satisfaga las necesidades de los usuarios. El incumplimiento de estos criterios deriva en UI poco usables, sistemas complejos y falta de aplicaciones para atender padecimientos específicos. Al identificar estos criterios faltantes en la mayoría de las aplicaciones, se propone seguir un método de desarrollo para, a partir de actividades diseñadas por un experto, elaborar un sistema altamente interactivo y que siga guías de usabilidad para favorecer el proceso de rehabilitación del lenguaje.

3 Desarrollo

Como parte del desarrollo de un videojuego para soporte a la terapia del lenguaje se requiere una guía metodológica que permita de manera ordenada y sistémica resolver el problema planteado. Existen muchas metodologías para desarrollar de videojuegos serios [10][11][12], en este caso se considera la propuesta de [13], en la que se toma como base el desarrollo basado en modelos del marco de trabajo CAMELEON [14]. Se propone entonces, un proceso que comienza con la observación y análisis de las actividades llevadas a cabo durante las sesiones de terapia con un especialista, para posteriormente elaborar Interfaces Abstractas (AUI), ajenas a la plataforma y a la modalidad de interacción, Interfaces Concretas (CUI), ajenas a la plataforma pero específicas a un tipo de modalidad e Interfaces Finales (FUI), implementaciones para una plataforma y modalidad. Finalmente, se describen guías seguidas durante la implementación del sistema con el fin de garantizar que sea usable y que las tareas identificadas puedan ser llevadas a cabo de manera eficiente.

Para realizar este trabajo, se considera una actividad propuesta por un especialista en terapia de lenguaje adscrito a la Unidad de Atención Integral a la Salud de la Universidad Autónoma de Aguascalientes, para terapia a pacientes diagnosticados con hipoacusia bilateral profunda con implante coclear, la cual consiste en mostrar al paciente instrumentos musicales y enseñarle el sonido que emiten. Posteriormente se hacen sonar de nuevo los instrumentos sin que estos sean visibles al paciente para evaluar si los puede identificar correctamente.

Fig. 1. Modelo de tareas para la actividad descrita usando notación CTTE.

Para analizar la actividad proporcionada por el especialista, se identifica a los actores involucrados en la sesión de terapia y se les asigna el rol de *sistema* o de *usuario*. Comúnmente sólo se involucran en una sesión al especialista y al paciente y por lo tanto la asignación de roles resulta trivial, dando al especialista el rol de *sistema* y al paciente el rol de *usuario*. Aplicando este análisis a la actividad descrita, se puede notar que los actores en ella son sólo el especialista y el paciente por lo que como se indicó, los roles de *sistema* y *usuario* se asignan de manera inmediata. Después, se realiza la identificación de tareas y la asignación de las mismas al rol correspondiente. Para la actividad descrita, se identifican como tareas del sistema: mostrar instrumento, reproducir sonido, solicitar respuesta y evaluar respuesta. Para el usuario, se identifica sólo la tarea de proporcionar respuesta. A partir de la información con la que se cuenta hasta este momento, se elabora el modelo de tareas mostrado en la Fig. 1 utilizando la notación CTTE [15].

La tarea en el nivel más general, recibe el nombre de la actividad y tiene naturaleza abstracta. En un segundo nivel, se derivan dos tareas: presentar información que a su vez se divide en mostrar instrumento y reproducir sonido; y evaluar avance que se subdivide a su vez en solicitar respuesta, proporcionar respuesta y evaluar respuesta. Siguiendo los pasos del marco de trabajo, a partir del modelo de tareas, se efectúa un proceso de concretización del que se obtienen la AUI que consta de un contenedor general, correspondiente a la tarea de más alto nivel del modelo de tareas, dos más correspondientes a las tareas del segundo nivel y componentes que describen la interacción del usuario con el sistema. Después, la AUI recibe un proceso más de concretización del que se tiene como resultado una CUI. Este modelo se realiza en dos partes, una correspondiente a la modalidad gráfica y la otra presentada en la Fig. 2 para la modalidad vocal utilizando la notación de [16]. Con respecto a la parte gráfica del juego, del modelo de AUI se observa la necesidad de contar con un contenedor general que se traduce en la pantalla del dispositivo, la cual se divide en dos partes, una para

presentar información y otra para permitir acceder a la evaluación. En el modelo de la parte vocal se presentan contenedores en los dos niveles más generales y componentes, en el tercer nivel, que permiten la interacción del usuario con el sistema. El componente *Sistema reproduce sonido* forma parte de la tarea *Reproducir sonido*, mientras que para la *Evaluación del avance*, se utilizan: *Sistema solicita respuesta* y *Usuario proporciona respuesta*.

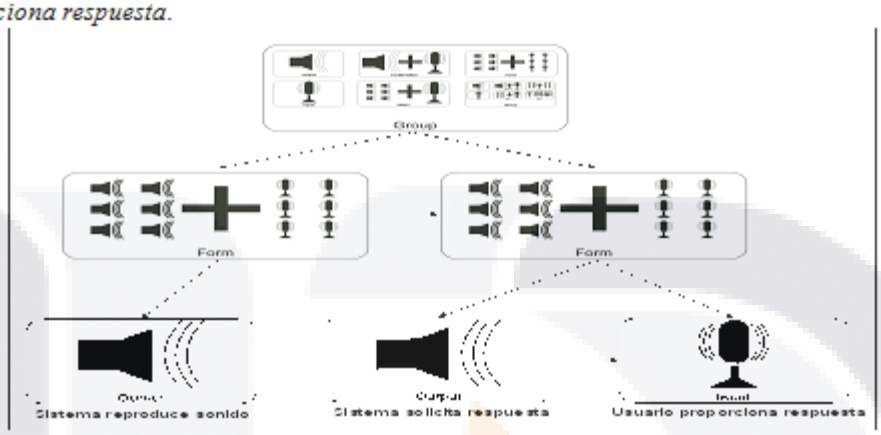


Fig. 2. Modelo CUI para la parte **vocal** del sistema utilizando la notación de [16].

A partir de este punto, se puede llevar a cabo la implementación de un videojuego serio para soporte a la terapia de lenguaje. Se considera un dispositivo móvil tipo tableta con pantalla de 10.1 pulgadas y sistema operativo Android Versión 4, pero al contar con los modelos diseñados, implementar el sistema en otro dispositivo requiere sólo un esfuerzo en cuanto a programación. El modelo FUI se muestra en la Fig. 3.

Fig. 3. Modelo FUI para el prototipo de videojuego serio desarrollado.

De las guías de usabilidad mencionadas en [13], se utilizaron para la implementación de este prototipo por su pertinencia y compatibilidad con los requisitos, aquellas referentes a la utilización de recursos de audio claro y fuerte, imágenes representativas de los elementos que forman parte del videojuego, la experiencia del paciente en el uso de dispositivos móviles que puede ir desde nula tomando en cuenta la edad de los usuarios y la variación en el volumen.

4 Evaluación del prototipo

La evaluación del prototipo elaborado puede realizarse tomando en cuenta diversos aspectos, pero para este trabajo, se decide evaluar la pertinencia de la aplicación del videojuego como parte complementaria a las sesiones de terapia utilizando la técnica de evaluación con base en la opinión de un experto [17]. Así, una vez realizada la implementación, se muestra ésta al especialista que diseño la actividad de manera que la evalúe cualitativamente. El especialista califica de manera general la aplicación como útil para complementar la labor llevada a cabo en las sesiones de terapia. Menciona además que una aportación del videojuego diseñado está en la motivación que puede causar en los pacientes de acuerdo con su experiencia personal al incorporar la utilización de dispositivos móviles. Señala también que al integrar la actividad interactiva con la evaluación de la competencia en la aplicación, llevar a cabo la tarea se simplifica y que se puede extender la terapia gracias a la portabilidad que le da el dispositivo móvil. Expresa la necesidad de aumentar la galería de objetos que forman parte del sistema y diseñar más niveles de complejidad.

5 Conclusiones y Trabajo Futuro

Se logró desarrollar un prototipo de juego serio con base en una actividad diseñada por un especialista al seguir un método de desarrollo que involucra análisis de actividades, identificación de usuarios y roles, modelo de tareas y de interfaces y finalmente, implementación de un sistema. La validación del prototipo fue realizada con base en la opinión de un experto, de la que se obtuvo como resultado una valoración positiva. Como trabajo futuro, se incorporará el uso del videojuego desarrollado a las sesiones de terapia convencional para extenderla y poder entonces evaluar la interacción de los pacientes con el sistema y comprobar la utilidad de la incorporación de videojuegos en el proceso de terapia en cuanto al tiempo de tratamiento requerido y el alcance objetivos por parte de los pacientes respecta. También, se desarrollaran más videojuegos para soportar distintas etapas de la terapia.

Agradecimientos

Se agradece el apoyo brindado por la LTF Claudia Verónica Ávila Díaz de León para el diseño de la actividad base de este trabajo y para la labor de evaluación de la aplicación implementada.

6 Referencias

1. Yildirim, S., Serious Game Design for Military Training, Proceedings of the Games: Design & Research Conference. Volda University College. Noruega, 2010.
2. Stege, L., van Lankveld, G., Spronck, P., Serious Games in Education. International Journal of Computer Science in Sport. Vol. 10. 2011.
3. Wattanasoontorn, V., Boada, I., García, R., Sbert, M., Serious Games for Health. Entertainment Computing Vol. 4. 2013.
4. Gunasekera, W., Bendall, J., Rehabilitation of Neurologically Injured Patients, Neurosurgery, Springer. 2005 pp. 407-421.
5. Cleuren, L., "Speech Technology in Speech Therapy?", State of the Art and Onset to the development of a Therapeutic Tool to Treat Reading Difficulties in the First Grade of Elementary School. SLT Internship en ESAT-PSI Speech Group. 2003.
6. Ittleman, M., The Teaching of Talking, Learn to do Expert Speech Therapy at Home. Morgan James Publishing. USA, 2013.

7. Rego, P., Moreira, M., Reis, P., Serious Games for Rehabilitation, a Survey and a Classification Towards a Taxonomy. 5th Iberian Conference on Information Systems and Technologies (CISTI). Santiago de Compostela, 2010.
8. Martins T., Carvalho, V., Moreira, F., Serious Games as a Tool to Intellectual Disabilities Therapy: Total Challenge. IEEE 1st International Conference on Serious Games and Applications for Health (SeGAH). Braga, 2011.
9. Cagatay, M., Ege, P., Tokdemir, G., Cagiltay, N., A Serious Game for Speech Disorder Children Therapy. 7th International Symposium on Health Informatics and Bioinformatics (HIBIT). Nevsehir, 2012.
10. Bowman, D., The Science of Interaction Design, en Doug Bowman, Ernst Kruijff, Joseph LaViola, Ivan Poupyrev, and Mark Mine, SIGGRAPH 2000 Course 3D User Interface Design: Fundamental Techniques, Theory, and Practice. 2000.
11. Dachselt, R., Hinz, M., Meißner, K., CONTIGRA: An XML-Based Architecture for Component-Oriented 3D Applications. In Proceedings of 7th International Conference on 3D Web Technology Web3D'2002. ACM Press, New York, 155–163. 2002.
12. Laviola, J., Bringing VR and Spatial 3D Interaction to the Masses through Video Games, IEEE Computer Graphics and Applications.2008.
13. Gonzalez-Calleros, J.M., A Model-Driven Approach for Developing Three-Dimensional User Interfaces of Information Systems in a Principle-based Way, PhD Thesis, Université catholique de Louvain Press. 2010.
14. Calvary, G., Coutaz, J., Bouillon, L., Florins, M., et al., The CAMELEON Reference Framework, Deliverable 1.1, Version V1.1, CAMELEON Project Campbell, E., Maintaining accessible Websites with Microsoft Word and XML, Proc. of XML Europe 2003, XML Workshop Ltd., Londres. 2003.
15. Paternò, F., ConcurTaskTrees: An Engineered Notation for Task Models, en The Handbook of Task Analysis for Human-Computer Interaction. 2003. pp. 483-503.
16. Céspedes-Hernández, D., González-Calleros, J., Guerrero-García, J., Rodríguez-Vizzuett, L., Model-Driven Development of Vocal User Interfaces. Human Computer Interaction, Proceedings of the 6th Latin American Conference, CLIHC 2013. Springer. 2013.
17. Castillo-Ávila, A., Desarrollo de una Aplicación en Línea para la Creación de Encuestas de Evaluación de Usabilidad de Software, Tésis de Licenciatura, Benemérita Universidad Autónoma de Puebla. México, 2013.