

**VIDEOJUEGO BASADO EN LOS SINTOMÁS Y FACTORES DE RIESGO DE LA
TUBERCULOSIS PULMONAR, UTILIZANDO UN MODELO DE INCORPORACION DE
AUDIO AL MOTOR GRÁFICO OGRE**



Trabajo de Grado

Jimmy Jesús Santacruz Cifuentes

Diego Felipe Zapata Martínez

Director: MSc. Miguel Ángel Niño Zambrano

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Departamento de Sistemas

Grupo I+D en Tecnologías de la Información

Popayán, Febrero de 2009

Nota de Aceptación

Firma Presidente del Jurado

Firma de Jurado

Firma de Jurado

Popayán 09 de Febrero de 2009

AGRADECIMIENTOS

A la Universidad del Cauca institución que nos forjó como personas brindándonos la oportunidad a través del programa de Ingeniería de Sistemas para realizar nuestros estudios de pregrado y de la cual siempre hemos recibido apoyo.

Por el apoyo que recibimos en el desarrollo del presente trabajo de investigación, queremos expresar nuestros agradecimientos a las siguientes personas:

MIGUEL ANGEL NIÑO ZAMBRANO, Magíster en informática, director del trabajo de investigación, por su apoyo constante e incondicional durante el desarrollo del proyecto.

NORMAN HERNANDO MUÑOZ, Ingeniero de sistemas, por sus aportes, ardua colaboración y valiosos consejos que nos permitieron alcanzar los objetivos de este proyecto.

CARLOS COBOS, MARTHA ELIANA MENDOZA, JAIME LOPEZ docentes del departamento de sistemas, que ante cualquier inconveniente presentado durante el transcurso del proyecto siempre estuvieron dispuestos a colaborar para proporcionar la solución adecuada.

ALVARO BACCA, Estudiante de diseño gráfico. Por su labor en el realce visual del juego.

Nuestros amigos y compañeros que siempre estuvieron brindándonos su apoyo y animándonos para salir siempre adelante.

En especial a nuestros padres, hermanos y demás familiares que nos brindaron todo su amor, comprensión y apoyo para que este proyecto pueda culminarse.

A todas las personas que finalmente de una u otra forma nos ofrecieron ayuda en el momento más oportuno.

MUCHAS GRACIAS

TABLA DE CONTENIDO

1	INTRODUCCIÓN	10
1.1	DESCRIPCIÓN DEL PROYECTO	10
1.1.1	Definición del Problema	10
1.1.2	Justificación	13
1.2	OBJETIVOS	16
1.2.1	Objetivo General	16
1.2.2	Objetivos Específicos	16
2	MARCO TEORICO	17
2.1	TUBERCULOSIS	17
2.1.1	Síntomas	18
2.1.2	Factores de Riesgo	18
2.2	VIDEOJUEGOS	19
2.3	MOTORES	23
2.3.1	Motor de Videojuego	23
2.3.2	Motor Gráfico	23
2.3.3	Clasificación de los Motores	24
2.4	OGRE	25
2.4.1	Proyectos que Utilizan OGRE	26
2.5	MOGRE	27
2.5.1	Proyectos que utilizan MOGRE	28
2.6	AUDIO	29
2.6.1	Librería de Audio BASS	29
2.6.2	Librería de OPENAL (Librería Abierta de Audio)	30
3	MODELO DE INTEGRACION DE AUDIO A OGRE	32
3.1	ARQUITECTURA DEL MODELO DE AUDIO MOGRE	33
3.2	ANÁLISIS	35
3.3	DISEÑO	37
3.4	IMPLEMENTACION	38
4	DISEÑO E IMPLEMENTACIÓN DEL VIDEOJUEGO	42
4.1	METODOLOGÍA ESCOGIDA PARA EL DESARROLLO DEL VIDEOJUEGO	42
4.2	DISEÑO DEL JUEGO	43
4.2.1	Objetivos del Juego	44
4.2.2	Descripción Genérica del Videojuego	44
4.2.3	Guión del Videojuego	50
4.3	DISEÑO DE SOFTWARE DEL VIDEOJUEGO	54
4.3.1	Arquitectura del Videojuego	55
4.3.2	Patrones de Diseño Aplicados y Diagrama de Clases de la Arquitectura	59
4.4	IMPLEMENTACIÓN	76
4.4.1	El Motor Simón Engine	77
4.4.1.1	Extensibilidad del videojuego, uso de XML y XSD	79
4.4.2	Test de Preguntas del Juego	83
4.5	PRUEBAS DE CAJA NEGRA	84
5	PRUEBA DE CAMPO	91
5.1	DESCRIPCIÓN GENERAL	91
5.1.1	Marco de Referencia	91
5.1.2	Diseño del experimento	92
5.1.2.1	Definición del Experimento	94
5.1.2.2	Ejecución de la práctica	95
5.1.2.3	Resultados	99
5.1.2.4	Análisis de los resultados	101

6	CUMPLIMIENTO DE LOS OBJETIVOS	105
7	CONCLUSIONES RECOMENDACIONES Y TRABAJO FUTURO.....	110
7.1	CONCLUSIONES.....	110
7.2	RECOMENDACIONES.....	112
7.3	TRABAJO FUTURO.....	112
8	GLOSARIO Y BIBLIOGRAFIA.....	114
8.1	GLOSARIO	114
8.2	BIBLIOGRAFIA.....	118

INDICE DE FIGURAS

Figura 1. Modelo de inclusión de Audio a MOGRE	32
Figura 2. Arquitectura del modelo de inclusión de las librerías de audio BASS y DIRECT SOUND a MOGRE	34
Figura 3. Arquitectura del modelo de inclusión de las librerías de audio BASS, DIRECT SOUND, OPENAL y otras a MOGRE	35
Figura 4. Diagramas de caso de de uso.	36
Figura 5. Diagrama de clases de diseño del modelo de audio.	37
Figura 6. Aplicación de prueba para el modelo de audio.....	39
Figura 7 Diseño de la IGU que presenta el menú principal. "Interfaz de Controles"	45
Figura 8. Primer Still de narración de la historia	46
Figura 9. Simon en su escenario de acción.....	46
Figura 10. Diagrama de Flujo del videojuego	47
Figura 11. Estados de Simón.....	49
Figura 12 Ejemplo de Convenciones usadas en el planteamiento de una escena	52
Figura 13 Planteamiento de una Escena con mapa y diálogos.....	53
Figura 14. Diagrama de Capas de la Arquitectura.....	56
Figura 15. Diagrama de Capa "Módulo de Juego"	57
Figura 16. Diagrama del Paquete de Lógica	58
Figura 17. Diagrama de Paquetes de clases. Visión de la Arquitectura	60
Figura 18. Paquete de PlopOgre	61
Figura 19. Muestra las clases del módulo Graphics	62
Figura 20. Muestra las clases del módulo Input	63
Figura 21. Muestra la clase del módulo Utils	63
Figura 22. Sección OnGameState	64
Figura 23. Sección mundo.	66
Figura 24. Sección Conversación.	69
Figura 25. Sección Condición	71
Figura 26. Sección Still	73
Figura 27. Sección Guión	74
Figura 28. Clases en Persistencia	75
Figura 29. Visión de la Solución en Visual Studio	77
Figura 30. Visión del directorio de carpetas en Visual Studio.....	78
Figura 31. Diagrama de clases test de preguntas del videojuego	83
Figura 32. Niños del grupo experimental en interacción con el videojuego.....	93
Figura 33. Niños del grupo de control en charla magistral.	93
Figura 34. Diseño de la prueba.....	95
Figura 35. Niños en igualdad de condiciones para la charla magistral.....	96
Figura 36. Presentación del cuestionario presentado al grupo experimental. Ver ANEXO D ...	98
Figura 37. Presentación del cuestionario presentado al grupo de control. Ver ANEXO D.	98
Figura 38. Medición del conocimiento.	101
Figura 39. Comparación de los datos obtenidos en la pregunta 1 por el grupo de control y el grupo experimental	102
Figura 40. Comparación de los datos obtenidos en la pregunta 2 por el grupo de control y el grupo experimental	102
Figura 41. Comparación de los datos obtenidos en la pregunta 3 por el grupo de control y el grupo experimental	103
Figura 42. Comparación de los datos obtenidos en la pregunta 4 por el grupo de control y el grupo experimental	103

Figura 43. Comparación de los datos obtenidos en la pregunta 5 por el grupo de control y el grupo experimental 104

INDICE DE TABLAS

Tabla 1. Videjuegos beneficiosos para la salud.....	21
Tabla 2. Videjuegos educativos.	22
Tabla 3. Clasificación de motores.	24
Tabla 4. Características de algunos motores que son de libre distribución.....	24
Tabla 5. Calidad de la documentación y los efectos especiales de los motores más utilizados.	24
Tabla 6. Algunos proyectos que utilizan OGRE.....	27
Tabla 7. Algunos proyectos que utilizan MOGRE.....	28
Tabla 8. Clasificación del videojuego Simon y la Tuberculosis.	45
Tabla 9. Descripción de los estados del videojuego.	48
Tabla 10. Descripción de las acciones del personaje principal.	50
Tabla 11. Descripción de las Capas de la Arquitectura.	57
Tabla 12. Descripción de los componentes de la capa lógica.	59
Tabla 13. Descripción de las carpetas del motor Simon Engine.	79
Tabla 14. Clases de equivalencia válida e inválida	88
Tabla 15. Evaluación de los Casos de Pruebas	90
Tabla 16. Tiempos estimados para la realización de la prueba teórica	97
Tabla 17. Tiempos estimados para la realización de la prueba práctica	97
Tabla 18. Perfil grupo de control.....	100
Tabla 19. Perfil grupo experimental	101

INDICE DE LISTADOS

Listado 1. Segmento de código que muestra la forma de crear y manipular un dispositivo de audio	40
Listado 2. Código a cambiar para usar Bass como librería de audio	41
Listado 3. Marcado XML que define un MUNDO	80
Listado 4. Marcado en XML que define las conversaciones en el mundo	81
Listado 5. Marcado en XML que define los flags	81
Listado 6. Marcado en XML que define la secuencia de escenas en el juego	82
Listado 7. Marcado en XML que permite configurar uno de los StillSet.	82
Listado 8. Marcado en XML que permite configurar uno de los StillSet.	84

1 INTRODUCCIÓN

1.1 DESCRIPCIÓN DEL PROYECTO

1.1.1 Definición del Problema

El computador se ha convertido en una herramienta utilizada en diversas áreas de la vida cotidiana, es por esto que en la actualidad es común ver a todo tipo de usuarios interactuando con éstas máquinas, ya sea profesionales que realizan labores de oficina, o estudiantes de colegio y universitarios en sus actividades académicas. Sin embargo no es el único uso que se les da, ya que también se utilizan como fuente de entretenimiento. Esta herramienta ha ido desplazando lentamente el uso de otros medios como la televisión, que antes era el primero en ser empleado como instrumento de distracción y en algunos casos de educación. Es así como hoy en día se aprecia que los niños prefieren sentarse por horas frente a un computador debido a la interacción con dicha máquina [1].

Esta preferencia de la juventud, se debe al control que el usuario tiene sobre el computador y por la facilidad de aprovechar los innumerables servicios que se pueden realizar en este medio en donde el joven usuario no se limita a observar -como en el caso del televisor- sino que controla el tiempo y el espacio del flujo de la información, manipulándolo, además, como un instrumento de ocio, tal es el ejemplo de los videojuegos, donde se puede observar que los niños emplean grandes cantidades de tiempo divirtiéndose frente a sus monitores [1].

Lo anterior se reafirma en los estudios realizados por profesores de la Universidad de Navarra y el Colegio Irabia de Pamplona, en el que participaron 4.000 escolares de varias ciudades españolas dando como resultado que el 47% de los niños prefieren los videojuegos frente a un 34% que prefieren ver televisión [1].

Esta mixtura de tecnología, entretenimiento y educación es de especial interés y es por esto que en los países desarrollados se están utilizando los videojuegos con el objetivo de educar a todo tipo de público interesado en aprender sobre diferentes áreas [2].

La estrategia de aprendizaje, educar mientras se juega, es una forma atractiva para enseñar a niños, jóvenes y adolescentes, pero cabe analizar si el uso de videojuegos a través de consolas y computadores, es una metodología efectiva para adquirir conocimiento. En tal sentido se puede explorar si, por ejemplo, es posible que un determinado grupo etáreo pueda aprender una temática en particular usando videojuegos, esto teniendo en cuenta que un alto porcentaje de niños entre los 10 y 14 años asegura utilizar más los computadores que otros sistemas de información tradicionales [3].

La temática de un videojuego, puede abordar una historia cualquiera, puede estar relacionada con hechos reales sobre una problemática en particular que esté afectando directa e indirectamente a la sociedad. Un tema que hoy en día es de mucha importancia en nuestro medio y en el que se ven afectados tanto los adultos como los jóvenes y sobre todo los niños, es el de la salud, es por esto que se desea abordar este tema para el desarrollo de un videojuego e incluirlo en una historia referente a una enfermedad y verificar si es posible aprender acerca de ella.

En la actualidad existen enfermedades infectocontagiosas como la malaria o paludismo, dengue, dengue hemorrágico, VIH/SIDA y la tuberculosis pulmonar, entre otras [4]. La tuberculosis pulmonar es la causa de una gran cantidad de muertes en países desarrollados y en vía de desarrollo. Según los últimos estudios realizados, en los últimos diez años viene presentándose un dramático incremento en la incidencia de la tuberculosis pulmonar al punto que desde 1993, la tuberculosis es considerada por la Organización Mundial de la Salud como un problema urgente de salud pública en todo el mundo [5]. En Colombia se calcula que cada año se diagnostican 10.000 casos nuevos de tuberculosis siendo Vaupés, Guainía, La Guajira, San Andrés y Putumayo, los departamentos que se consideran con más alto riesgo [6]. Esta estadística se podría reducir o evitar si las personas infectadas y las potencialmente contagiadas estuvieran al tanto de los síntomas y factores de riesgo de esta enfermedad y del grave peligro que corren al estar expuestos a la bacteria que la causa.

El gobierno Colombiano cuenta con varios programas de atención y prevención, los cuales no solo son ofrecidos por el Ministerio Nacional de Salud, sino también por diferentes organizaciones a nivel mundial como la Organización Mundial de la Salud –OMS, con el mismo

fin: lograr disminuir el índice de infectados por la bacteria que ocasiona la tuberculosis pulmonar. Estos programas son llevados a cabo mediante campañas educativas y anuncios publicitarios. La OMS esta apoyando la creación y formación de nuevos programas propuestos para la reducción del índice de contagios de tuberculosis, debido a que con las actuales técnicas de información utilizadas no se logra llegar a los grupos más afectados como lo es el sector infantil [7].

Ante esta necesidad existen en Colombia las herramientas o los medios de comunicación para divulgar la información y son muy diversos, como lo es, la radio para llegar a lugares con difícil acceso ya sea por su ubicación geográfica o por problemas de orden público; la televisión que abarca a diferentes clases sociales; las nuevas tecnologías como Internet que es una herramienta que ha ido desplazando diferentes medios de comunicación; y todo tipo de periódicos y revistas. Pero todos estos mecanismos no son suficientes para concientizar a las personas sobre la importancia de conocer las consecuencias de esta enfermedad, por tal razón es factible aprovechar el auge del uso del computador para llegar a nuevos sectores sociales, por ejemplo a través de programas que ofrece el gobierno como Computadores para Educar - CPE, donde se busca que diferentes escuelas, colegios e instituciones educativas de pocos recursos, especialmente en los sectores rurales, cuenten con las herramientas tecnológicas para suplir las necesidades pedagógicas en los procesos de enseñanza y aprendizaje y así utilizar el computador como un eficaz difusor de concientización del peligro de la tuberculosis pulmonar [8]. Por lo anterior surge la pregunta: ¿el uso de los videojuegos de computador, como un medio de comunicación diferente a los tradicionales, puede facilitar el aprendizaje acerca de los síntomas y factores de riesgo de la tuberculosis pulmonar en un grupo de niños entre los 10 y 14 años de edad?

Esta pregunta lleva a pensar si el uso de herramientas modernas como los videojuegos permite mejorar el aprendizaje en la población juvenil, siendo una alternativa a los medios ya mencionados y qué características deben poseer para tal efecto. Esas características son la razón por la cual algunos de ellos tienen más demanda que otros, por tal motivo existe la necesidad de adecuarlos de la manera más atractiva para capturar un número masivo de jugadores y facilitarles por medio de los videojuegos no sólo la adquisición de habilidades manuales, coordinación, orientación tiempo y espacio, sino también para la adquisición de conocimientos [9]. Es aquí donde se busca sacar el mayor provecho de las herramientas tecnológicas ofrecidas para la creación de videojuegos y por medio de ellas conseguir las características necesarias para atraer de mejor manera la atención del jugador.

Entre estas herramientas tecnológicas están los motores gráficos, motores de audio y plataformas entre otros, los cuales ayudan a los desarrolladores a crear sistemas de entretenimiento con mayor facilidad y rapidez. Entre estos motores encontramos al Object-Oriented Graphics Rendering Engine (OGRE) [10], motor gráfico que por sus características permite desarrollar videojuegos de una manera más rápida, pero tiene la desventaja de ser un motor orientado a la parte gráfica dejando a un lado otras características importantes como el audio. Por esta razón, es necesario cuestionarse: ¿Cuál es la mejor forma de incorporar audio a OGRE, para que pueda ser utilizado como un motor de un videojuego educativo?

1.1.2 Justificación

Según el Ministerio de Salud de Colombia la tuberculosis es una de las enfermedades más comunes en éste país y por este motivo han surgido diferentes formas o tratamientos efectivos para manejar esta enfermedad. El gobierno Colombiano se ha fijado como meta reducir hasta en un 85% la población total contagiada con la bacteria causante de la tuberculosis pulmonar y detectar el 70% de estos casos, ofreciendo un servicio totalmente gratis para el tratamiento de esta enfermedad [5]. Sin embargo, por falta de información, las personas no aprovechan este servicio ya que piensan que los gastos, tanto de la medicina, como los de la consulta corren por su cuenta.

Uno de los sectores que se ve afectado por esta enfermedad es el infantil, el cual tiene poca tendencia hacia los medios que utilizan la lectura como sistema de divulgación tales como, periódicos y revistas [11], pero cabe señalar que estos medios son los más utilizados para informar a las personas sobre la tuberculosis, por esto se hace indispensable buscar nuevas alternativas, como los juegos, que resulten más atractivas para proveer toda esta información a dicho sector.

Debe ser claro que hoy en día a causa de los avances tecnológicos como la computadora y los sistemas de entretenimiento que ésta ofrece, los niños y los adolescentes no juegan lo mismo que sus padres y abuelos, de tal forma que se dedican a jugar durante gran parte de su tiempo con un computador o una consola de videojuegos [1].

Por esta razón que mejor método para que los niños aprendan, haciendo lo que más les gusta, es “jugar”, aprovechando de esta manera las ventajas tecnológicas que hoy en día pueden estar al alcance de cualquier grupo o estrato social [1][12].

Jugar hace parte del aprendizaje y es mediante el juego que el niño transforma su mundo, lo manipula, lo arma y rearma y adquiere las estructuras cognitivas que fundamentan su razonamiento y favorecen el desarrollo de su intelecto. Según Jean Piaget los juegos ayudan a enriquecer la experiencia y el conocimiento de los niños, ya que ellos representan en los juegos experiencias vividas transformando todo lo que en la realidad pudo ser penoso y convirtiendo lo insoportable en algo soportable e incluso agradable [13].

Al realizar un videojuego educativo, relacionado en este caso con la tuberculosis pulmonar y dado que hoy en día los niños prefieren jugar videojuegos a ver televisión, se plantea un gran reto en términos de poder validar si efectivamente es posible aprender sobre una temática en especial, usando como medio de aprendizaje un juego para computador, pues no sólo se piensa en el desafío tecnológico que implica la creación del mismo, sino en la forma apropiada que permita que llegue al niño y le enseñe sobre la temática en cuestión [14].

La creación de un video juego que eduque de manera satisfactoria sobre la tuberculosis pulmonar puede generar un alto impacto social, en la medida que se va a lograr un mayor conocimiento sobre los síntomas y factores de riesgo de esta enfermedad, convirtiéndose en una atractiva fuente de información para diferentes grupos etáreos inicialmente de la sociedad caucana y tal vez en un futuro, dependiendo de su éxito, en diferentes departamentos de nuestro país.

Actualmente en el departamento del Cauca la temática del desarrollo de videojuegos está empezando, por lo cual este proyecto busca conseguir espacios y adeptos en la Universidad del Cauca para que sigan trabajando e investigando en esta área de interés y de manera paulatina actualicen los conocimientos para crecer e innovar en este campo. La idea no está centrada solo para estudiantes pertenecientes a programas relacionados con las tecnologías de Información, sino que también se desea llegar a múltiples programas en otras áreas y de esta manera impulsar al trabajo interdisciplinario para enriquecer el conocimiento de las partes involucradas a través del intercambio de ideas.

Un aporte importante para los desarrolladores de videojuegos es que al desarrollar un esquema que permita la incorporación de audio al motor gráfico OGRE, se está ofreciendo a la comunidad académica y en general a la comunidad usuaria de dicho motor, una herramienta

que ayudará en la labor de creación de videojuegos. Además, otras áreas relacionadas, como la simulación en tiempo real y en general aplicaciones que requieran el uso de las mencionadas características, se verán beneficiadas, pues tendrán a su disposición no sólo el código fuente, sino la experiencia ganada en el proceso, siendo ésta de gran relevancia al interior del área de interés en desarrollo de videojuegos del Grupo de I+D en Tecnologías de la Información (GTI) de la Universidad del Cauca.

1.2 OBJETIVOS

1.2.1 Objetivo General

Proponer un modelo que incorpore audio al motor OGRE y desarrollar un caso de estudio, a través de un videojuego, que permita a niños entre 10 y 14 años de edad, aprender acerca de los síntomas y factores de riesgo de la tuberculosis pulmonar.

1.2.2 Objetivos Específicos

- Crear e implementar un modelo que permita la incorporación de audio a OGRE, de forma tal que pueda ser utilizado en futuros trabajos para el desarrollo de nuevos videojuegos y aplicaciones multimedia en 3D.
- Crear un nivel ¹de un videojuego que cumpla las siguientes especificaciones:
 - Sea elaborado a partir de un guión basado en la detección de síntomas y factores de riesgo de la tuberculosis pulmonar.
 - Considere las características básicas de un juego de aventura 3D, tal como un escenario ambientado según la trama del juego, personajes tridimensionales genéricos, uso del teclado para la interacción con los personajes, entre otras.
 - Utilice la implementación del esquema que incorpora el audio al motor gráfico.
- Determinar, mediante la realización de una prueba de campo de tipo tradicional con estudiantes entre 10 y 14 años de un colegio público o privado de la ciudad de Popayán, si la utilización del videojuego y del audio facilita el aprendizaje de la temática referente a la tuberculosis pulmonar.

¹ Se determino crear un nivel del videojuego, debido a que la temática de la TB es muy extensa, por eso el primer nivel tratara solamente sobre los síntomas y factores de riesgo de dicha enfermedad; existe un segundo nivel relacionado a la TB que esta siendo implementado por otros estudiantes del programa de Ing. De Sistemas de la Universidad del Cauca. Se espera a futuro poder ir adaptando estos niveles para ir ampliando los alcances del videojuego.

2 MARCO TEORICO

En este apartado se dará a conocer los conceptos necesarios para que el lector pueda comprender el contenido de este documento, tal como los síntomas, factores de riesgo, formas de contagio, tipos de exámenes y otros temas relacionados a la tuberculosis pulmonar; de igual manera se tocarán temas relacionados con los videojuegos, con mayor énfasis en los de aventura que manejan gráficas tridimensionales, pues son hoy en día los más apetecidos por todos aquellos interesados en este tipo de entretenimiento; finalmente se especificarán algunos motores de juegos, motores gráficos y motores de audio para videojuegos en 3D, su descripción y especificaciones junto con la importancia de la incorporación de interesantes características como el audio a un motor gráfico y las distintas librerías que pueden utilizarse para adquirir estas particularidades.

2.1 TUBERCULOSIS

La tuberculosis pulmonar es una enfermedad infectocontagiosa bacteriana producida por una micobacteria del complejo *Mycobacterium*: *M. Tuberculosis*, *M. Bovis*, o *M. africanum* [5]. Fue una de las primeras causas de muerte en todo el planeta y dio origen a numerosas estrategias terapéuticas, desde las aguas medicinales, las sangrías hasta los más recientes esquemas de antibioticoterapia combinada [16].

Una persona solo puede llegar a ser infectada o contagiada de tuberculosis pulmonar a través de la inhalación de partículas de esputo expulsadas por una persona enferma, ya sea estornudando, hablando o tosiendo; cabe señalar que esta bacteria no se transmite por medio de un beso, relaciones sexuales, o por el simple hecho de compartir utensilios de comida y ropa [5]. En caso de contagio o infección, las manifestaciones clínicas se presentan generalmente entre los 6 a 12 meses. El período de incubación de la bacteria en una persona adulta oscila entre 2 a 12 semanas y no todos desarrollan la enfermedad, solo el 10% de los infectados en alguna época de su vida pasaran de estado de infección a enfermo. En el caso de los niños, solo el 5% de los infectados desarrollan la enfermedad en algún momento de sus

vidas, pero si los niños son menores de cinco años, el riesgo de desarrollo aumenta a un 25% [5].

Una persona puede ser infectada pero no tener la enfermedad. Las personas que se han infectado con Tuberculosis Pulmonar (TB) pero no han desarrollado enfermedad activa de TB, no pueden transmitir la enfermedad a otras. A estas personas se les practica el examen más común para detectarla, que es el de la prueba de BK esputo (Examinar que tan avanzada esta la enfermedad analizando la flema del paciente), prueba desarrollada hace más de 100 años y que a mediados de la sexta década del siglo pasado, ayudo a controlar de manera satisfactoria la tuberculosis en muchos países del hemisferio occidental [17].

2.1.1 Síntomas

La mayoría de la gente infectada por tuberculosis pulmonar no presenta síntomas. La bacteria yace latente en los pulmones y puede permanecer ahí sin causar la enfermedad. Durante la etapa latente, la persona infectada no puede contagiar la tuberculosis pulmonar a los demás. Una vez que la bacteria se vuelve activa, la infección puede contagiarse con facilidad.

Una persona puede sospechar que ha sido infectada con el bacilo *Mycobacterium Tuberculosis* porque ha padecido de 15 o más días de tos con expectoración, además no sólo debe sospechar de tuberculosis pulmonar debido a este síntoma, sino a cualquier infección respiratoria que no muestre mejoría alguna siendo tratada con antibióticos. Estos son los principales síntomas, pero generalmente se pueden incluir síntomas constitucionales como pérdida o ganancia de peso, pérdida o ausencia de movimiento o reacción, sudoración abundante, fiebre, toser sangre y esputo (mucosidad de los pulmones), dolor en el pecho, escalofríos, pérdida de apetito y vomito [17].

2.1.2 Factores de Riesgo

Un factor de riesgo es algo que incrementa la posibilidad de contraer una enfermedad o padecimiento.

La gente que tiene un sistema inmune debilitado o una enfermedad crónica está en mayor riesgo de desarrollar la TB activa. Entre ellos están:

- Infección por VIH.
- Desnutrición.
- El uso de drogas.
- La indigencia.
- Alcoholismo.
- Leucemia, linfomas y otros tipos de cáncer.
- Diabetes débilmente controlada.
- Enfermedad grave del riñón.
- Sistema inmune debilitado debido al uso de medicamentos, tales como drogas para prevenir el rechazo de un órgano trasplantado.

Otros factores de riesgo incluyen:

- Silicosis (una enfermedad del pulmón causada por el trabajo).
- Vivir en condiciones de hacinamiento, tales como:
 - Refugios sin techo.
 - Dormitorios públicos.
 - Barracas militares.
- Edad: Infantes, niños pequeños y ancianos.

2.2 VIDEOJUEGOS

También conocidos como juegos de video, son el resultado del trabajo de personas especializadas en este campo, donde desarrollan un programa informático o software, creado para divertir a personas que interactúan con la consola donde se encuentra instalado. Los videojuegos recrean entornos virtuales con una variedad de animaciones y personajes, donde

el jugador manipula uno o varios elementos de dicho entorno para poder lograr un objetivo [18].

En ocasiones los objetivos de los videojuegos se rigen a cumplir una serie de tareas o acciones para dar fin al mismo, sin recibir a manera de contrapartida por parte del videojuego algún tipo de beneficio para el usuario o jugador. Por el contrario, en ocasiones el uso excesivo de los videojuegos puede resultar nocivo para la salud así lo demostró un estudio realizado por los médicos del Instituto Mexicano del Seguro Social, quienes revelaron que cerca de un millón de usuarios de computadores y videojuegos, sufren de lesiones en las articulaciones de sus manos, en columna vertebral e incluso en el corazón [34].

Estos problemas que trae consigo el estar largas horas sentado frente a una consola o un computador personal haciendo uso único de las manos y los dedos no es nada nuevo, desde hace tiempo atrás estas prácticas vienen preocupando a padres de familia, psicólogos, a los mismos jugadores, e incluso a quienes pertenecen al área de la salud como por ejemplo la AMA (Agrupación de Médicos Americanos) [35] es un grupo que solicita que se realicen investigaciones de las consecuencias que trae el estar largas horas sentado frente a una pantalla.

Los resultados de investigaciones arrojan además de las lesiones ya mencionadas, problemas como el sedentarismo, enfermedades visuales y trastornos psicológicos [36], pero no todo es malo; en los últimos años se han venido presentando interesantes propuestas de videojuegos que brindan beneficios al usuario tales como la salud y la educación.

Gracias a que los estudios e investigaciones realizados muestran que la cantidad de personas de diferentes edades que se divierten con los videojuegos es bastante elevada, se ha querido sacar provecho de esta situación y poco a poco los videojuegos se han convertido en una importante herramienta con las cuales se puede disfrutar mientras se hace ejercicio [36]. Desde la Wii y periféricos especiales para movernos mientras jugamos, hasta campañas orientadas a fomentar el ejercicio físico, pasando por videojuegos que ayudan a pacientes a conocer sus enfermedades de una manera mucho más natural [36]. A continuación se mencionan algunos videojuegos que son beneficiosos para la salud:

NOMBRE	DESCRIPCIÓN
KILOWATT	Es un dispositivo de ejercicios isométricos en el cual los jugadores emplean la fuerza física para interactuar con las escenas que se ven en la pantalla de video. [37]
FREEDIVE	Videojuego que contiene un mundo submarino de fantasía cuyo propósito es distraer a los pacientes pediátricos del dolor o la ansiedad. [37]
BEN'S GAME	Es un videojuego que consiste en correr en patineta sobre un campo minado de células cancerosas. [37]
RE-MISSION	Es un videojuego para adolescentes y jóvenes con cáncer, el videojuego les ofrece a los jugadores una sensación de poder sobre esta enfermedad al disparar y destruir las células del cáncer. [38]
EYE TOY	Es un videojuego para consola Play Station 2, el cual permite luchar contra el sedentarismo mientras se realizan diversos deportes. La imagen del jugador es representada en pantalla por medio de una web cam. [39]
¿QUÉ HARÍAS TÚ?	Este videojuego consiste en representar a los personajes de un grupo de cuatro amigos, la idea del juego es prevenirlos contra el SIDA por medio de situaciones que se van presentando en el mismo y las decisiones que deben tomar los personajes. [39]

Tabla 1. Videojuegos beneficiosos para la salud.

Existen también videojuegos que resultan positivos para desarrollar el aprendizaje [40], son llamados popularmente videojuegos educativos, el uso de estos introduce al usuario en un entorno que lo motiva a superar fases y dificultades asumiendo así el control y el mando de la situación convirtiéndose en el verdadero protagonista de la trama. Un estudio de la Universidad Europea de Madrid (UEM) asegura que esta práctica desarrolla en los niños, jóvenes y adultos jugadores competencias como: la sociabilidad, la destreza visual, el trabajo en equipo y la capacidad de superación y liderazgo [40].

Es notorio que los videojuegos educativos pueden resultar bastante productivos para el proceso de aprendizaje de los niños sobre todo si son bien orientados por personas adultas para que no se cometan excesos y para estimular a los niños estos tipos de videojuegos. Pero, ¿como incentivar a los niños a jugar videojuegos educativos?, la idea es hacer el videojuego bastante atractivo al usuario y definir a que edades podría ir dirigido. El desarrollo de un videojuego educativo no es nada fácil, para lograr realizar un verdadero videojuego educativo que llame la atención de un niño de algún tema en particular se necesita de la ayuda e investigación de personas especialistas en los temas tratados.

A continuación se listan solo algunos videojuegos educativos:

NOMBRE	DESCRIPCIÓN
MISIÓN SEGURA	Proyecto destinado a la formación de conductores tanto militares como civiles en todos los aspectos necesarios para mejorar la seguridad vial y la conducción. [41]
EMERGENCIA 112	Es un juego para móviles que además de entretener, instruye a los jugadores como responder y prevenir una situación de emergencia médica, como un paro cardíaco. [41]
ELDESAFÍO DE WORKY	Aventura educativa destinada a escolares cuyo objetivo es dar a conocer las situaciones que puedan acarrear riesgos laborales y educar en su prevención. [41]
APRENDE MÚSICA CON PIPO	Es un videojuego enfocado a enseñar conceptos básicos de la música principalmente a los niños. Enfocado a niños entre 2 y 12 años, pero es apto para todas las edades por el material didáctico y educativo que posee
CONSUMO JOVEN	proyecto destinado a jóvenes para informarles de sus derechos como consumidores y orientarles en la adquisición de su primera vivienda.[41]

Tabla 2. Videojuegos educativos.

Hoy en día los juegos de mayor éxito son los de 3D, los cuales requieren para su implementación un motor apropiado (el motor o engine es el sistema que presenta y manipula el ambiente en tres dimensiones, además controla el sonido y permite la interacción con el jugador) [19]. En el mercado se pueden conseguir gran variedad de motores que cumplan con los requisitos apropiados para la elaboración de videojuegos en 3D, pero para la adquisición de muchos de ellos hay que pagar una licencia, mientras que existen otros que son de libre distribución, lo que significa que no hay que pagar por su uso.

El motor gráfico del presente proyecto fue escogido luego de un estudio de las necesidades requeridas teniendo en cuenta ante todo los recursos económicos disponibles para la elaboración del mismo, debido a que el costo de las licencias de algunos motores puede superar los 500.000 dólares, dinero o presupuesto con el cual no cuentan muchos grupos o empresas desarrolladoras de software [20]. Esta es una de las razones por la cual los desarrolladores de videojuegos que cuentan con un bajo presupuesto económico se inclinan por la adquisición de motores que no requieran el pago de una licencia.

2.3 MOTORES

2.3.1 Motor de Videojuego

Es el núcleo de una aplicación interactiva o videojuego con gráficos en tiempo real. La ventaja más notoria es que al utilizarlo, permite simplificar mucho el trabajo del desarrollador abstrayendo lo más importante de la idea o el proyecto a crear. Además estos motores permiten desarrollar gran variedad de videojuegos, como de aventura, multiplataforma y estrategia entre otros [21].

2.3.2 Motor Gráfico

Entre los motores que son de libre distribución se pueden encontrar los motores gráficos. Estos son un conjunto de librerías y funciones de alto nivel, implementadas para el manejo de gráficos en tiempo real, estas pueden ser en 2D o 3D. El objetivo del motor gráfico, es facilitar el trabajo al desarrollador permitiéndole una manipulación e implementación realizable de operaciones robustas durante el desarrollo de cualquier tipo de gráficos. Es necesario entender que un motor gráfico no es un motor de videojuegos, pero en ocasiones se le integran nuevas funcionalidades como el audio, manejo de graficas 3D, manejo de entrada, manejo de colisiones entre otras, para ser utilizado como un motor de videojuego.

Una gráfica 3D es la que esta construida en tres dimensiones: largo, ancho y profundidad. En el caso de los videojuegos cuando estos contienen escenas en 3D, estas escenas son conocidas como realidad virtual, la que ofrece una sensación de realismo al jugador permitiéndole involucrarse tanto en el videojuego hasta cierto punto que el jugador cree que hace parte de la trama del juego y se olvida un poco de los acontecimientos que están ocurriendo a su alrededor [22].

2.3.3 Clasificación de los Motores

A continuación se muestran algunos ejemplos de motores tanto gráficos como de videojuegos [23].

NOMBRE DEL MOTOR	GRÁFICO	VIDEOJUEGO
REALM FORGE GDK		X
TORQUE GAME		X
TV3D SDK		X
CRYSTAL SPACE		X
JME	X	
AXIOM	X	
OGRE	X	

Tabla 3. Clasificación de motores.

Para el caso de este proyecto, y por su naturaleza académica se requiere de un motor que sea de libre distribución, por lo cual en la Tabla 4 se muestra un cuadro comparativo con las principales características de los tres motores de libre distribución más utilizados actualmente para la creación de videojuegos.

CARACTERÍSTICAS DE LOS MOTORES	IRRLICHT	CRYSTAL SPACE	OGRE
GRÁFICOS API	OPEN GL	SI	SI
	DIRECT X	SI	NO
	SOFTWARE	SI	SI
MANEJO DE ESPACIOS	BSP	SI	NO
	OCTRESS	NO	NO
	CUADTRESS	NO	NO
LENGUAJES DE PROGRAMACIÓN	C	SI	SI
	C++	SI	SI
	C#	SI	NO
	VB.NET	SI	NO
SCRIPTING	SI	SI	NO
SONIDO	SI	SI	NO
VIDEO	SI	SI	SI

Tabla 4. Características de algunos motores que son de libre distribución.

	IRRLICHT	CRYSTAL SPACE	OGRE
DOCUMENTACIÓN	BUENO	REGULAR	EXCELENTE
EFECTOS ESPECIALES	REGULAR	REGULAR	EXCELENTE

Tabla 5. Calidad de la documentación y los efectos especiales de los motores más utilizados.

Según las características de estos motores y por los requisitos fundamentales para el desarrollo del proyecto, OGRE es el más adecuado por las siguientes razones:

Es un motor gráfico 3D muy completo y bastante rápido. Otros motores, con física incluida y demás (como Crystalspace), son lentos y pesados, y cuentan con una interfaz muy compleja.

OGRE ya lleva 4 años de desarrollo, su arquitectura es bastante buena y se cuenta con gran variedad de tutoriales de excelente calidad que son fáciles de seguir [24].

Otra de las principales razones es que OGRE es compatible con DirectX, que es una colección de API's² creadas para facilitar tareas relacionadas con la programación de juegos en la plataforma Microsoft Windows, plataforma en la que desarrolló el videojuego.

OGRE posee un excelente manejo de escena y da soporte para lenguajes como C#.

2.4 OGRE

OGRE (Object-Oriented Graphics Rendering Engine) es un motor para el renderizado gráfico orientado a objetos, esta escrito en C++ y diseñado para que los desarrolladores de software construyan aplicaciones que involucren gráficas 3D de una manera más fácil e intuitiva. La clase library (Biblioteca) abstrae todos los detalles de usar librerías de sistemas subyacentes como Direct3D y OpenGL y además proporciona una interfaz basada en objetos del mundo real y otras clases intuitivas.

Es importante tener en cuenta que OGRE no es un motor de videojuegos sino un motor gráfico, OGRE puede ser (y de hecho ha sido) usado para hacer videojuegos, pero OGRE esta diseñado para proporcionar soluciones gráficas de gran nivel; para otras características como sonido, gestión de redes, inteligencia artificial, colisiones, física etc, se necesita integrarlo con otras librerías.

Todo esto tiene un por que, una razón es por que no todos los que necesitan un motor 3D desean hacer videojuegos, usted puede utilizar OGRE para realizar videojuegos, simulaciones, aplicaciones de negocios, y muchas cosas más. En segundo lugar, incluso en la industria de los videojuegos, los requerimientos pueden variar ampliamente; por ejemplo un simulador de

² Interfaz de programación de aplicaciones (Applications Programming Interface): Es una serie de funciones que están disponibles para realizar programas para un cierto entorno.

vuelo necesitará un tipo de sistema diferente de colisiones físicas al de un juego de peleas. Si OGRE incluyera todas estas características, estaría forzando a los desarrolladores a usar un conjunto limitado de librerías con un conjunto inconstruible de requerimientos asumidos, y este no sería un buen diseño. En lugar de eso, OGRE proporciona una API de integración muy amigable y le deja escoger a los desarrolladores las otras librerías en caso que las necesite [25].

Muchos desarrolladores de juegos experimentados han expresado su aprobación a OGRE por que no hay restricciones en el momento de construir lo que se quiere.

2.4.1 Proyectos que Utilizan OGRE

A continuación se presenta una pequeña selección y descripción de algunos de los proyectos incluyendo videojuegos, simuladores y herramientas realizados con el motor gráfico OGRE los cuales son los más reconocidos y muestran la gran variedad de cosas que se pueden hacer al utilizar OGRE como una mejor opción.

Los proyectos son:

NOMBRE DEL PROYECTO	DESCRIPCIÓN
SUB SAFE	Es un juego que ha sido usado por HFI DTC/Aerosystems Internacional para crear un avanzado simulador 3D de un submarino de la armada. Este proyecto fue desarrollado por la compañía "Incredible Box". SUB SAFE implementa una gran variedad de técnicas de juego para enseñar conciencia espacial, ubicación de objetos y proporciona información del funcionamiento del equipo.
TIME OF WAR	Es un juego en tercera persona de un disparador en un futuro cercano, escrito por Oniric Games en Argentina.
NEO AXIS ENGINE	Es un motor de juego y un kit de herramientas diseñado para la creación de títulos de juegos de todos los géneros como también proyectos de realidad virtual y visualización 3D.
INTERACTIVE MUSEUM 18th CENTURY NANTES	Axyzimages lanzó al museo de historia "The Castle of the Dukes of Brittany" de Nantes, una aplicación interactiva utilizando OGRE donde el usuario puede visitar algunos distritos de Nantes durante el siglo 18. Algunos documentos archivados fueron usados para reproducir la ciudad cerca del año 1757. Se ha creado una vista aérea basada en un grabado del siglo 18 y se han reproducido 11 distritos.
LIVE INTERIOR 3D	Es una herramienta para el diseño de interiores en 3D, incluyen objetos importados del almacén de google 3D y considerando el tiempo real, incluye también ajustes de iluminación dinámica.
FIRSTAID SIM	Es un simulador de primeros auxilios hecho por Guppyworks/Kongo.
HCA THE UGLY PRINCE DUCKLING	Es un juego clásico de aventura para niños entre los 7 y los 12 años de edad, producido por Guppyworks en Dinamarca. Este juego se caracteriza por que el fondo gráfico esta prerenderizado pero los personajes e interfaz son todos en

	tiempo real, con la iluminación prerenderizada dentro de una malla plana para iluminación dinámica de los personajes. Las sombras en tiempo real de los personajes son un grupo de goteras proyectadas hacia un plano el cual da un efecto realmente agradable en áreas abiertas.
PACIFIC STORM	Es un juego acerca de la Guerra en el Pacífico entre Japón y Estados Unidos. Tiene los elementos de una estrategia global, juego táctico y un simulador aéreo. Jugando para cualquiera de los dos, ya sea para Estados Unidos o para el imperio japonés, usted camina a través de una secuencia dramática de un gran complejo naval y de operaciones militares aéreas. El juego cubre los años desde 1940 hasta 1948.
THERE IS ONLY WAR	Es un juego de arcada de un típico colegio ubicado en un mundo lleno de color al estilo caricatura y dirigido a ofrecer un montón de experiencias de juego divertidas y positivas para los jugadores. Soporta Internet y multijugadores en redes LAN.
WALABERS TRAMPOLINE	Es un juego de acción y simulador de trampolines para computadores con Sistema Operativo de Windows. windows. El juego utiliza física y animación para crear un juego de trampolines en 3D el cual luce muy real. Utilizando un único esquema de control usando solo 7 botones, los jugadores pueden realizar casi todo movimiento posible en el trampolín, incluyendo volteretas múltiples y torceduras en diferentes posiciones.

Tabla 6. Algunos proyectos que utilizan OGRE.

Los anteriores como ya se mencionó son sólo algunos de los diversos proyectos realizados en OGRE, que muestran gran variabilidad de características que puede ofrecer este motor gráfico [26].

2.5 MOGRE

MOGRE (Managed OGRE) es una versión de OGRE implementada mediante código administrado³ en el Framework de .NET, orientada para desarrollos en C++.NET 2.0. El código que permite la unión entre C++ y .NET esta desarrollado usando C++/CLI, el cual ofrece gran flexibilidad y control de bajo nivel que muestra fácilmente como se debe y puede trabajar en esta plataforma.

Debido a que las clases de OGRE se integran en el framework de .NET transparentemente, se puede acceder de manera directa en el transcurso de desarrollo de una aplicación a todas las funcionalidades ofrecidas por las librerías de este motor.

³ En el mundo .Net el código administrado es el que se ejecuta sobre el .NET Framework, o sea aquel compilado en MSIL (MSIL significa Microsoft Immediate Language)

Cabe resaltar que OGRE (y por lo tanto MOGRE) es un motor que carece de audio, característica importante para que los videojuegos sean más atractivos y ayude a la persona que este interactuando con este a comprender la temática del videojuego con mejor claridad. Es por estas razones que se hace necesario incorporar audio a este motor [27].

2.5.1 Proyectos que utilizan MOGRE

Al ser MOGRE una versión muy reciente en el mercado de los motores gráficos, los proyectos encontrados son pocos, pero van en aumento, ya que cada vez son más las personas que elijen este motor por su fácil adaptabilidad a las nuevas tecnologías como lo es .NET y otras de libre distribución.

Los proyectos son:

NOMBRE DEL PROYECTO	DESCRIPCIÓN
YAT	Es la versión tridimensional del tradicional tetris, lo hace interesante es el excelente renderizado y la calidad de las graficas, además el poco espacio que requiere esta aplicación para poder ser instalado.
GRAFFITI FALLS	Es un juego en tercera persona, donde el objetivo es ir por las calles de una ciudad pintando y decorando los edificios de esta. Entre más edificaciones sean modificadas, más alto será el puntaje, este videojuego es dedicado a personas que buscan mejorar los interiores de un apartamento o casa viendo previamente los resultados para ahorrar tiempo y esfuerzos antes de hacer los cambios en las edificaciones reales.
WINGS OF FURY	Es un juego de acción con algunos aspectos de simulación, donde el jugador es un piloto de un avión americano F6 Hellcat a bordo del USS Wasp. El entorno del juego es en el pacifico durante la segunda guerra mundial. El objetivo del juego es derrotar al enemigo japonés mediante la destrucción de búnkeres, torretas y cuarteles.

Tabla 7. Algunos proyectos que utilizan MOGRE.

Estos son los proyectos más recientes que utilizan este motor y que disponen el código, al igual que la aplicación sin costo alguno a todas las personas que deseen adquirirlo, además hay que recordar que los proyectos que usan OGRE, también pueden ser compilados por medio de un WRAPPER⁴ en el ambiente MOGRE [28].

⁴ Es un paquete o envoltura el cual cambia la interface a un paquete ya existente sin incrementar sustancialmente su funcionalidad

2.6 AUDIO

Además de las gráficas en 3D existen otras características muy importantes que ayudan a comprender mejor el problema o tema del juego, un ejemplo es el sonido o audio que en muchas ocasiones pasa desapercibido. Un juego de video que carezca de sonido, no tendría futuro alguno ya que el sonido que acompaña al movimiento de las imágenes hace que el juego sea mucho más interesante y realista. Esto es lo que se ha ido buscando cada día más en el avance de los videojuegos “la semejanza con la realidad”, pero no solamente el sonido ayuda a dar realismo, sino que ayuda también a aumentar su nivel de comprensión y adaptación en el aprendizaje [29].

Por todas las ventajas y aportes del audio y gráficas 3D hacia el conocimiento y aprendizaje, se busca integrarlos en cualquier medio de comunicación, con el fin de lograr conciente o inconcientemente, que las personas los escuchen y observen, uno de estos medios son los videojuegos que son utilizados por gran número de usuarios en especial los niños y jóvenes [30].

El audio en los videojuegos es una característica importante, ya que gracias a éste, el videojuego ofrece una sensación de realismo junto con otras ventajas como la adaptación en el aprendizaje y mejora en el nivel de comprensión del tema del juego [29]. Pero para adicionar esta característica en un videojuego, es necesario contar con una librería especializada en la manipulación de audio, que incluya los componentes relacionados para poder hacer uso de los efectos sonoros en la aplicación [31]. Al igual que los motores antes mencionados (motores de videojuegos y motores gráficos), también existe una gran variedad de librerías de audio a disposición de los desarrolladores y que poseen diferentes especificaciones. Dos de ellas son BASS y OpenAL.

2.6.1 Librería de Audio BASS

Es una librería de audio que se puede utilizar en los sistemas operativos de Windows y Mac OSX. Su propósito es proporcionar a los desarrolladores las características más poderosas y eficientes en cuanto a [32]:

- Reproducir diferentes formatos de audio como MP3, MP2, MP1, OGG, WAV, AIFF.

- Ofrecer un completo soporte para todos los efectos, filtros, demos de estéreo, etc.
- Usar múltiples tarjetas de sonido simultáneamente.
- Poseer un sistema de grabación flexible, con múltiples dispositivos de selección de entrada y soporte.
- Aplicar el efecto que desee el usuario y en el orden requerido.

En la plataforma Windows, BASS requiere para la salida DirectX3 o una versión mejorada y toma ventaja de tarjetas aceleradoras como DirectSound y DirectSound3D cuando se encuentran disponibles.

2.6.2 Librería de OPENAL (Librería Abierta de Audio)

Es una librería multiplataforma para el manejo de audio 3D. Gracias a esta librería es posible, de manera muy fácil, reproducir sonidos en un espacio de manera muy flexible. OpenAL también tiene la capacidad de acceder directamente a funciones avanzadas de la tarjeta de sonido y de controlar la llamada aceleración de audio. A continuación una breve descripción de OpenAL [33].

- El uso de OpenAL no requiere pago de licencia alguna.
- Utiliza un mecanismo basado en extensiones.
- El motor de reproducción se encarga de todos los cálculos necesarios como la atenuación, doppler y otros efectos.
- Utilizado en el desarrollo de varios videojuegos reconocidos como Doom3, Quake 4 y Prey entre otros.
- El formato del sonido es PCM, WAV.
- Uso múltiple de tarjetas de sonido simultáneamente.

Otro de los puntos fuertes a destacar de OpenAL es su enfoque de trabajo con 'sonido 3D', gracias a lo cual, proporciona al programador la posibilidad de reproducir sonidos en una aplicación donde se simulará un efecto sonoro real de un espacio en 3D. Básicamente, para la reproducción de sonido con OpenAL es necesario especificar en un espacio tridimensional, un

punto de emisión y un punto en el que se encuentra el receptor que lo va a escuchar, después de lo cual el sonido elegido será procesado y colocado en un buffer de salida [33]

Como se ha mostrado anteriormente, lo que se espera es utilizar a OGRE como el motor del videojuego a implementar, pero éste, por ser un motor gráfico, no posee características importantes como lo es el audio, característica que por sus cualidades y ventajas (como el aumento de la capacidad receptiva y comprensión del tema [29]), hace que sea necesario incorporarla a este motor. Lo anterior se puede lograr, adicionándole librerías compatibles de audio como OpenAL, librería que modifica al motor de tal forma que lo convierte en una herramienta óptima para el desarrollo de videojuegos con imágenes y sonido 3D.

3 MODELO DE INTEGRACION DE AUDIO A OGRE

Al ser OGRE un motor diseñado especialmente para la manipulación de gráficas, este carece de otras características necesarias para poder ser empleado como un motor de videojuego. Esta es la razón por la cual se hace necesario integrarle audio a dicho motor y transformarlo en una poderosa herramienta, la cual se podrá aprovechar al máximo por las personas interesadas en el desarrollo de videojuegos, personas que no tienen los suficientes recursos ya sean tecnológicos o económicos para adquirir un motor de igual o menor rendimiento debido a que la mayoría de estos tienen un elevado costo en el mercado.

En la Figura 1 Se observa el modelo⁵ creado para incorporar la característica de audio al motor grafico OGRE, este modelo no sólo permite adicionar un formato de audio al motor sino que esta definido de tal manera que se pueda integrar gran cantidad de librerías, obteniendo así una amplia diversidad de funcionalidades relacionadas al manejo de sonidos.



Figura 1. Modelo de inclusión de Audio a MOGRE

⁵ Un Modelo es una representación gráfica o esquemática de una realidad, sirve para organizar y comunicar de forma clara los elementos que involucran un todo.

En la parte central de la figura se observa al motor OGRE el cual contiene ciertos módulos especializados, estos módulos le permiten al motor realizar diferentes acciones como lo es el renderizado de imágenes, manejo de cámaras, manejo de luces, etc.

La siguiente capa es MOGRE, la cual al contener a OGRE ofrece las mismas funcionalidades de éste, pero la diferencia es que le permite al desarrollador implementar aplicaciones basadas en el lenguaje de programación C#, lenguaje seleccionado para implementar el videojuego según se estableció previamente.

Luego se procede a introducir las librerías requeridas y necesarias para reproducir audio. En este proyecto son las librerías Bass y DirectSound, conformando así una tercer capa que permita ampliar las capacidades funcionales del motor gráfico. Estas capas a las cuales se ha hecho referencia, están incluidas en el modelo de audio, que es el punto de partida para que el desarrollador pueda hacer uso del motor gráfico para implementar un videojuego con efectos de audio.

Un paso muy importante antes de empezar a incluir el audio, es analizar e investigar como funcionan y que tipos de formatos de sonido reproducen las librerías de audio que se van a incluir en el motor. En el caso de este modelo, se investigó sobre las librerías existentes como lo es BASS, DirectSound, OpenAL, etc., creando un punto de entrada común para las diferentes tareas relacionadas con el audio en un videojuego, de manera que solamente sea necesario elegir la librería con la cual se desea trabajar.

Para poder desarrollar este modelo se eligió un tipo de patrón de diseño apropiado para la inclusión de audio al motor, además por facilidad, organización, rendimiento y otras razones, se dividen todos los componentes del modelo en módulos, los cuales se explicarán en el siguiente aparte.

3.1 ARQUITECTURA DEL MODELO DE AUDIO MOGRE

El modelo de arquitectónico para inclusión de audio en MOGRE se compone de cuatro módulos:

- Audio

Este módulo implementa el patrón fachada, permitiendo que la interfaz IAudioDevice represente al sistema y se encarga de controlar todas las funciones de reproducción y manejo de efectos como echo, chorus, etc.

- Sonido

Este módulo implementa el patrón factoría para dotar al sistema con la capacidad de extenderlo usando cualquier tipo de librería para manejo de audio, esta es una de las principales características de la arquitectura planteada ya que no limita el uso de un sistema de audio específico.

- Audio basado en Bass

Este módulo hace uso de la librería de BASS para dotar a MOGRE de sonido con efectos de audio y opciones básicas de reproducción.

- Audio basado en Direct Sound

Este módulo hace uso de la librería de DirectSound para dotar a MOGRE de sonido con efectos de audio y opciones básicas de reproducción.

Este modelo, sus cuatro módulos y la relación que hay entre estos, se puede apreciar claramente en la Figura 2. Arquitectura del modelo de inclusión de las librerías de audio BASS y DIRECT SOUND a MOGRE; donde inicialmente esta compuesto por dos librerías de audio, las cuales serán utilizadas en la implementación del video juego.

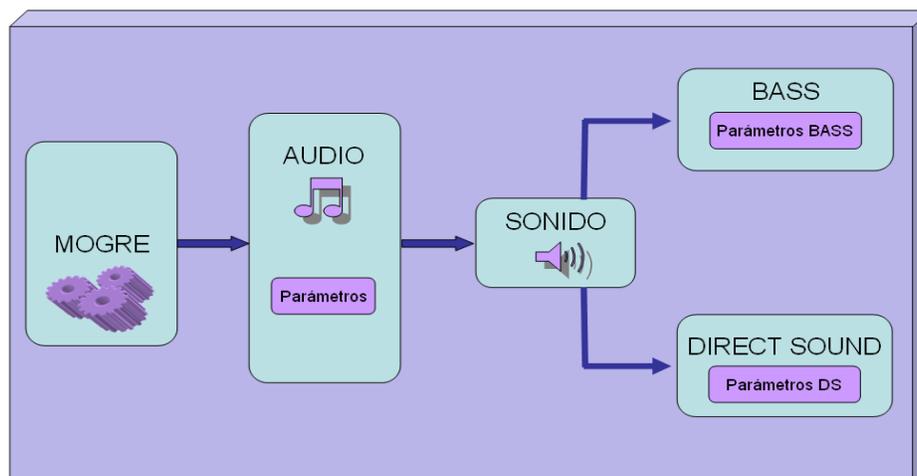


Figura 2. Arquitectura del modelo de inclusión de las librerías de audio BASS y DIRECT SOUND a MOGRE

La arquitectura de este modelo está diseñada para poder incluir más librerías de audio como se puede ver en la Figura 3. Arquitectura del modelo de inclusión de las librerías de audio BASS, DIRECT SOUND, OPENAL y otras a MOGRE. Éste modelo conserva la estructura de cuatro capas del modelo original, permitiendo así de manera fácil y sencilla al programador modificar las características de audio según las especificaciones que este desee.

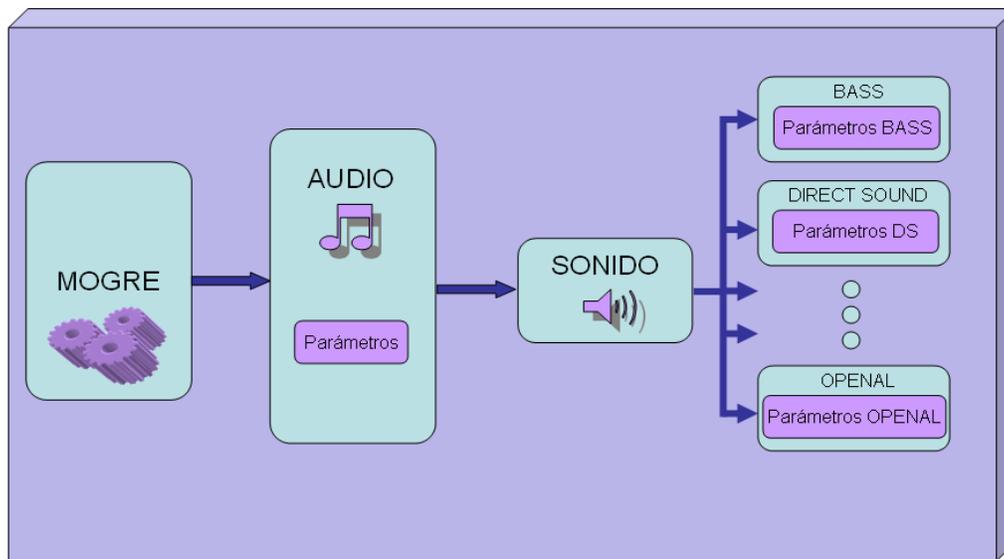


Figura 3. Arquitectura del modelo de inclusión de las librerías de audio BASS, DIRECT SOUND, OPENAL y otras a MOGRE

3.2 ANÁLISIS

Debido a que el usuario final del modelo serán los desarrolladores que estén interesados en éste, se pensó en crear un esquema basado en los requerimientos de un usuario que reproduzca música en un dispositivo como un equipo de sonido, o un Ipod, o un computador, etc. De manera que debe ofrecer funciones que estén disponibles en cualquier sistema o software como los que la mayoría de personas manipulan a diario. Esto permitirá que el desarrollador realice tareas conocidas para él, cuando esté incorporando audio a un videojuego.

Es así, como se crea el diagrama de casos de uso que se muestra en la Figura 4. Diagramas de caso de de uso. Donde se representa la forma en la cual un usuario, es decir el oyente,

opera el sistema de reproducción de audio, además se explica cómo los elementos interactúan entre sí para poder realizar todo tipo de operaciones o acciones que pueden presentarse cuando el usuario manipula el sistema.

Diagrama de casos de uso del modelo

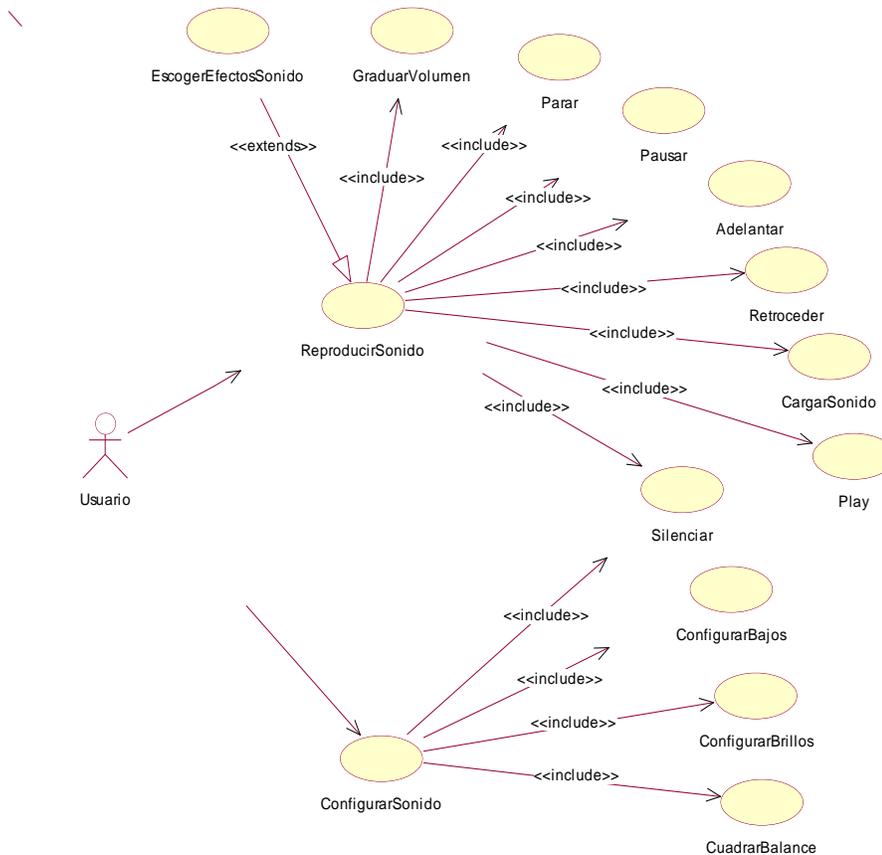


Figura 4. Diagramas de caso de de uso.

En este modelo se puede identificar las dos opciones principales que tiene un usuario en el momento de manipular un sonido y que son: Reproducir sonido y configurar sonido, operaciones con las cuales la mayoría de las personas ya se encuentran familiarizados, pues se utilizan en la vida diaria ya sea cuando se enciende la radio, el equipo de sonido, computador o hasta los dispositivos más pequeños y modernos como el Ipod, reproductores MP4 o inclusive los teléfonos celulares.

En el modelo se muestra algo que ya está casi mecanizado por las personas, cuando un usuario desea reproducir un sonido, lo primero que debe hacer es cargarlo, para ello busca el sonido de su preferencia, luego podrá escucharlo presionando ejecutar o reproducir. De ahí en adelante el usuario puede manipular el sonido en ejecución según su deseo, por ejemplo, podrá adelantarlo o retrocederlo ya sea para llegar pronto al final del sonido o para escuchar varias veces un mismo segmento del sonido en reproducción, podrá pausarlo, subir o bajar el volumen a gusto o en caso que necesite enmudecer el sonido tendrá la opción de hacerlo.

Además se presenta un caso de uso “Escoger Efectos de Sonido” el cual permite al usuario aplicar un efecto de sonido cuando lo requiera, la idea es permitirle al usuario involucrarse un poco más con el sonido, de manera que pueda usarlo para crear ambientes apropiados y dar la sensación que lo que está sucediendo en las acciones se aproxima a la realidad.

3.3 DISEÑO

Con base en los requerimientos iniciales se planteó el diagrama de clases de diseño que se muestra en la Figura 5. Diagrama de clases de diseño del modelo de audio., en el cual se integran las características básicas de lo que se espera de una librería de audio, pues no solamente se permite la reproducción del sonido, sino que se permite la reproducción de efectos de sonido.

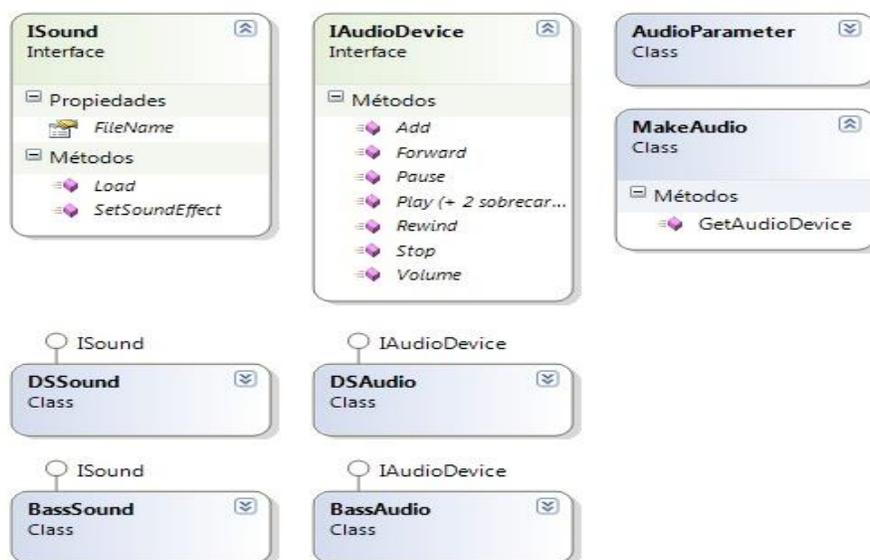


Figura 5. Diagrama de clases de diseño del modelo de audio.

El modelo propuesto se basa en la definición de una interface `IAudioDevice` que debe ser realizada por las clases que implementen la funcionalidad propia de cada librería, que para el caso son `DirectSound (DSAudio)` y `Bass (BassAudio)`. La interface define unos métodos que corresponden a las acciones necesarias para adicionar, reproducir y manipular sonidos. Cada clase que realice esta interface debe implementar dichos métodos de acuerdo a sus características e incluso su forma propia de reproducir sonidos.

De igual manera existe una interface `ISound` que corresponde a las particularidades de un sonido, por tanto, se busca que cada clase que realice esta interface (como `DSSound` o `BassSound`) adecue sus necesidades de carga de un sonido para su posterior reproducción.

Un problema que se presentaba con el modelo es que debido a las características tan disímiles entre las diferentes librerías, el tratar de unificar las interfaces fue bastante complicado, por lo que fue necesario dejar dichas interfaces un poco generales, para que después se pudieran usar.

Para poder cargar la librería apropiada, se utiliza la clase `MakeAudio` que tiene definido un método que retorna la interface de tipo `IAudioDevice`, para lo cual se le debe pasar que librería se va a usar y los parámetros necesarios para la inicialización (a través de `AudioParameter`).

El modelo se ha diseñado para soportar modificaciones sin que este pierda su consistencia, con la idea que si en un futuro se desean adicionar más librerías de audio como por ejemplo `OpenAL`, se pueda hacer fácilmente sin necesidad de cambiar lo que ya se encuentra definido y modelado.

3.4 IMPLEMENTACION

En lo que se exploró fue en elegir la plataforma más adecuada de desarrollo para implementar el modelo, dicha plataforma debe soportar el lenguaje de programación en el cual esta implementado `OGRE`. Con el fin de estar a la vanguardia en cuanto a las nuevas tecnologías, aprovechando la gran cantidad de facilidades que ofrece para el desarrollo de aplicaciones se opta por escoger `VisualStudio.NET`.

Al notar que `OGRE` esta diseñado e implementado en `C++`, el cual es un lenguaje de programación que hace unos años fué uno de los más empleados para el desarrollo de software, pero que hoy en día ya es algo obsoleto, se elige trabajar con un lenguaje más

actualizado y que esté a la vanguardia con las tecnologías recientes, por eso se opta por trabajar con C#, debido a que la comunidad de programadores en C# ha ido creciendo rápidamente y de esta manera se puede aprovechar los diversos conocimientos en varios entornos de programación utilizando un lenguaje interoperable basado en estándares.

Aquí surge una de las razones principales del por que utilizar MOGRE en vez de OGRE. MOGRE es básicamente una versión mejorada pero su principal diferencia es que el lenguaje de programación en el que esta implementado, es el mismo que se escogió para implementar el modelo y de esta forma se facilitara la inclusión y ejecución del motor en el entorno de desarrollo.

Después de implementar el esquema para el manejo de audio, se procedió a probarlo creando una aplicación que permitiera definir el dispositivo de audio a usar (o la librería a usar) permitiendo manipular sonidos guardados en archivos. Un pantallazo de la aplicación de prueba se muestra en la Figura 6.

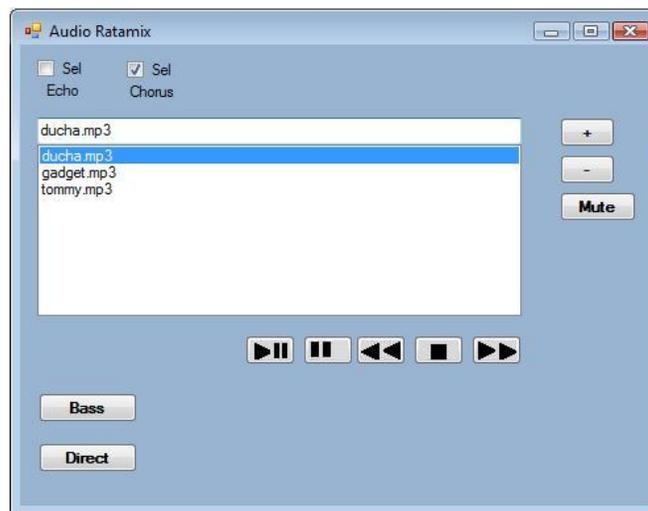


Figura 6. Aplicación de prueba para el modelo de audio

Internamente se crea un objeto del tipo de la interfaz IAudioDevice, para poder tener acceso a todas las funcionalidades que ofrece la librería, le permite al usuario manipular el audio con las funciones básicas de cualquier dispositivo de reproducción musical, tal y como se muestra en el Listado 1.

```
// Interface para el dispositivo de audio
private IAudioDevice audioDevice;
public Form1( ) {
    InitializeComponent( );
    // Parametros de inicialización del dispositivo
    AudioParameter parameters = new AudioParameter( );
    parameters.Owner = this;
    // Crea el dispositivo de audio de tipo DirectSound
    audioDevice = MakeAudio.GetAudioDevice(
AudioEngineType.DirectSound, parameters );
    // Adiciona un sonido
    audioDevice.Add( "ducha.wav" );
}
private void btnPlay_Click( object sender, EventArgs e ) {
    // Reproduce el sonido actual
    audioDevice.Play( );
}
```

Listado 1. Segmento de código que muestra la forma de crear y manipular un dispositivo de audio

La facilidad que ofrece el esquema implementado es que se puede cambiar de dispositivo de audio de una forma muy simple, tan sólo se debe especificar el tipo de motor de audio y adicionar los sonidos correspondientes, por ejemplo si se deseara trabajar con Bass en lugar de DirectSound, se debería cambiar únicamente la línea que se muestra en el Listado 2. y proporcionar los parámetros apropiados.

```
...
// Crea el dispositivo de audio de tipo DirectSound
```

```
audioDevice = MakeAudio.GetAudioDevice( AudioEngineType.Bass,  
parameters );
```

Listado 2. Código a cambiar para usar Bass como librería de audio

Esta es una gran ventaja, puesto que los programadores de videojuegos pueden adecuar sus desarrollos de acuerdo a sus necesidades, sin que esto implique que tengan que conocer todos los pormenores de cada librería. Esta característica, se constituye en lo que diferencia al modelo de otros como OGREAL que incorpora OpenAL al motor gráfico OGRE, pero no da la posibilidad de usar otras librerías.

4 DISEÑO E IMPLEMENTACIÓN DEL VIDEOJUEGO

El Videojuego planteado para la educación de la tuberculosis tiene por nombre "Simón y la tuberculosis", este se puede clasificar dentro del género de los juegos de aventura, su acción se lleva a cabo mediante conversaciones y superación de pruebas sin influencia alguna de la violencia por tratarse de un juego con propósitos educativos.

El videojuego ha sido realizado de forma interdisciplinaria con la asesoría de ALVARO BACCA estudiante de Diseño Gráfico, con el fin de mejorar su calidad estética así como el argumento y el arte alrededor de éste.

Este capítulo pretende abordar el proceso realizado para diseñar y desarrollar el primer nivel del videojuego, que muestre las características esenciales del mismo. Se abordará en su momento los detalles del diseño de Juego, la Arquitectura y Diseño de Software, algunos elementos de implementación y la descripción del soporte XML. Cabe mencionar que el videojuego ha sido desarrollado de manera altamente extensible para que en un trabajo futuro pueda ser implementado.

4.1 METODOLOGÍA ESCOGIDA PARA EL DESARROLLO DEL VIDEOJUEGO

La planeación de un juego no se aleja demasiado de los parámetros establecidos para la creación de un sistema tradicional, y por el contrario, conserva muchas de las características que se establecen en estos últimos. Sin embargo, si es conveniente aclarar que al igual que en los sistemas de información tradicional, cada quien tiene su propio estilo para planear y en la elaboración de juegos ocurre algo similar y cada diseñador de juegos define sus propias fases para el logro de la meta final, el juego.

Sólo por nombrar algunos casos, Todd Barron, define las fases de: invención, requerimientos, documentación técnica, desarrollo, pruebas, producción y distribución [42]; Francois Laramee, define seis pasos: tratamiento del diseño, diseño preliminar, diseño final, especificación del producto, la Biblia gráfica y el guión cinematográfico interactivo [43]; mientras que Chris

Crawford define solo tres fases: definir las estructuras de entrada y salida, del juego y del programa [44].

Se puede apreciar, que la planeación de un juego depende en gran parte de quien lo esté elaborando, pero a pesar de esto, existen características comunes como la documentación, la definición de características del juego, la estructura de la interfaz de entrada y salida y las relaciones existentes entre los componentes que harán parte del juego.

La metodología usada para la creación del juego motivo de esta investigación fue una adaptación del Proceso de Desarrollo Unificado, teniendo en cuenta que para la fase de análisis se utilizó el guión del juego el cual sirvió posteriormente para las etapas de diseño e implementación.

En conclusión, si se desea desarrollar un juego, antes de empezar a escribir toneladas de código, hay que sentarse a pensar en cómo se desea ver el juego, pero a partir de lo que se defina inicialmente, la historia y el propósito del juego. Con esto claro, se puede escribir las características del juego, lo que debe tener (requerimientos), las interacciones entre los componentes, la interfaz de entrada y salida y en fin todo aquello que se debería considerar al crear un sistema de información convencional.

4.2 DISEÑO DEL JUEGO

La primera fase antes del desarrollo del videojuego debe contar con el diseño del juego como tal, en esta etapa de la construcción del juego fue necesaria la total intervención por parte del estudiante de Diseño Gráfico quien se encargó, en gran medida, de establecer las reglas de juego, los estados correspondientes de cada etapa del videojuego y los detalles del diseño de personajes y por supuesto el guión de videojuego. Este trabajo interdisciplinario también fue soportado por el equipo de desarrollo, pues fue crucial contemplar aspectos limitantes tecnológicos para implementar el videojuego.

Dado que el Diseñador Gráfico debe presentar estos productos también como parte de su trabajo de grado, el diseño del juego se compone de algunos elementos adicionales, que se verán en su momento, que no están implementados en el desarrollo de este primer nivel.

4.2.1 Objetivos del Juego

El videojuego "Simón y la tuberculosis" es un producto con enfoque educativo y de entretenimiento, que pretende mostrar algunos aspectos de contenido crítico acerca de la tuberculosis, así como contar una historia argumental de fantasía para que transmita esta información de forma más agradable y entretenida.

Por tanto, "Simón y la Tuberculosis" es un videojuego que establece los siguientes objetivos.

- Proporcionar un videojuego de contenido interesante que permita transmitir información acerca de la tuberculosis.
- Utilizar una forma alternativa diferente a las prácticas comunes tales como películas y charlas para difundir e informar sobre los síntomas y factores de riesgo de la tuberculosis.
- Demostrar en un proyecto de mediano tamaño las capacidades de MOGRE como Framework ⁶y sus capacidades de desenvolvimiento Gráfico.
- Demostrar de forma práctica el uso y cualidades de la librería de Sonido basada en BASS y DirectSound explicada en el capítulo anterior.

4.2.2 Descripción Genérica del Videojuego

El videojuego, como fue concebido, es un videojuego con las siguientes características:

- Se puede jugar por niños y adolescentes.
- Es un Videojuego 3D, para ello el uso de MOGRE3D.
- Usa el audio como parte del juego a través de su incorporación al motor gráfico utilizado.

⁶ Un framework, en el desarrollo de software es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado

- Presenta al jugador un mundo de interacción donde tiene que cumplir pruebas y determinadas acciones para conseguir los objetivos y así proseguir con la narrativa de la historia (descrita más adelante).

De esta forma, el juego a grandes rasgos cuenta con la siguiente clasificación de videojuego:

CARACTERÍSTICA	DESCRIPCIÓN
CATEGORÍA	Juego de aventura narrativa.
RATING ESRB [45]	E por Everyone.
NÚMERO DE JUGADORES	1.
PLATAFORMA	Windows XP 32 bits.
DISPOSITIVOS DE ENTRADA	Mouse y teclado.
CLASIFICACIÓN DE VIDEOJUEGO POR TIPO DE CÁMARA	Videojuego de tercera persona.
TIPO DE CÁMARA	Sigue al personaje principal.
ACCIÓN DE JUEGO	Acción no violenta, resolución de pruebas a base de conversaciones y acertijos.

Tabla 8. Clasificación del videojuego Simon y la Tuberculosis.

El juego de video posee tres interfaces genéricas gráficas de usuario. La parte de interacción por controles, la cual presenta al usuario las opciones del Menú principal, los llamados "Stills" que son imágenes con descripción en texto a las que se les ha incorporado audio, que sirven para contar la historia argumental que une los mundos y las pruebas en el mundo de "Simón y la Tuberculosis"; y el desarrollo del mundo como tal.



Figura 7 Diseño de la IGU que presenta el menú principal. "Interfaz de Controles"

La figura 8 muestra el primero de los stills utilizados que narra el comienzo de la historia, en este se aprecia la imagen de Simon junto con la narración en texto. Se hace necesario aclarar que este texto va acompañado con su respectiva narración en audio.

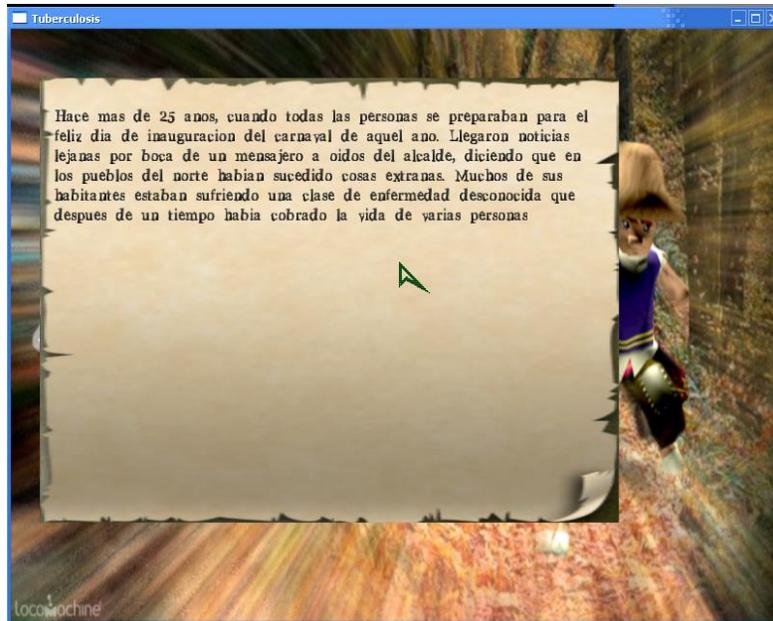


Figura 8. Primer Still de narración de la historia

En la figura 9 se muestra el personaje principal en acción dentro de su escenario de juego, en ella se alcanza a contemplar algunos modelos importantes que hacen parte del mundo del videojuego como son: el terreno, el cielo, el pozo, los árboles, la casa y los guardias.



Figura 9. Simon en su escenario de acción.

El Diseño del flujo del juego se explica mediante el siguiente Diagrama de Flujo.

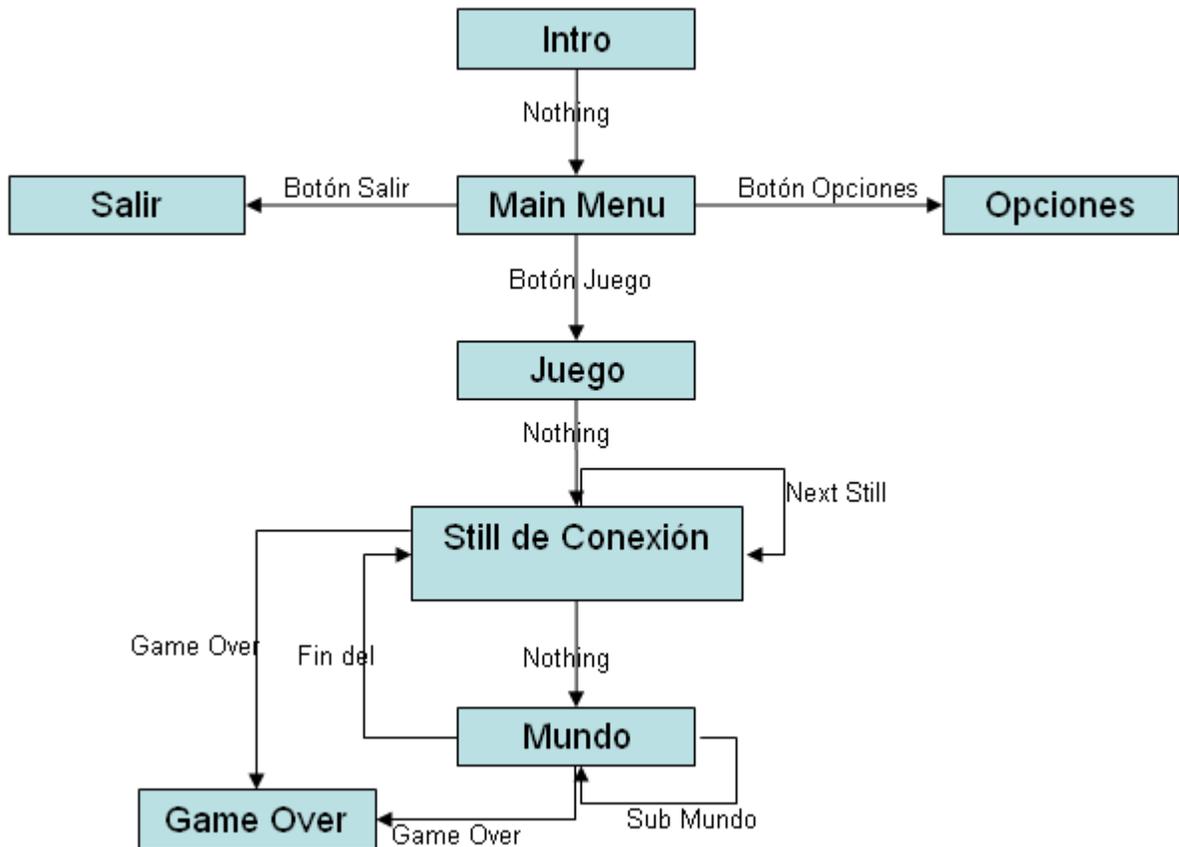


Figura 10. Diagrama de Flujo del videojuego

Como se observa en la Figura 10, el videojuego está compuesto por una secuencia de estados dentro del videojuego. A continuación se explican estos estados de forma general.

ESTADOS DEL VIDEOJUEGO	DESCRIPCIÓN
INTRO	En esta parte del juego sólo se muestra una imagen a modo de presentación del juego.
MAIN MENU	Es la interfaz de uso de controles. Aquí el usuario puede escoger entre Jugar, Salir o visitar las Opciones del juego.
SALIR	Termina el juego.
OPCIONES	Muestra la pantalla del videojuego donde se pueden configurar aspectos técnicos del juego, en particular, el volumen del sonido.
JUGAR	Esto inicia la narración de la historia y el videojuego en sí.
STILL DE CONEXIÓN ARGUMENTAL	Muestra en pantalla un Still presentando una parte de la historia. Después de que terminan todos los Stills que conectan la historia, empieza el desarrollo de la historia por parte del jugador, es decir el juego en sí o el desarrollo del mundo. Es posible naturalmente, que algunos Stills sean la terminación de la historia y por tanto den por terminado el videojuego.
MUNDO	Muestra el Mundo de interacción donde ocurren los eventos de la narración de la historia. Esta Parte del juego es el núcleo central del videojuego y es donde el jugador se concentra en desenvolver la historia y completar las pruebas. En este punto el jugador maneja a "Simón", quien es el personaje principal de la historia. Cuando el mundo termina, es posible que conecte con otro mundo para cumplir con la secuencia de mundos que determine el guión, o bien puede mostrar de nuevo Stills de Conexión Argumental. Por otro lado, también la culminación del mundo puede significar la culminación del juego total.
GAME OVER	Se ha terminado el juego.

Tabla 9. Descripción de los estados del videojuego.

Una vez definido este diagrama, se ha construido un Guión de Juego que está constituido por las pantallas, escenarios y situaciones que competen al ambiente del videojuego como tal; La narrativa de fantasía que se ha escrito como argumento para el videojuego y un conjunto de reglas de juego que comprenden las capacidades del jugador para interactuar con el juego que se traducen en las habilidades de Simón en los mundos. (Anexo A)

Acciones del Jugador.

Simón, el personaje principal del juego, desempeña el papel de Personaje Principal dentro de "Simón y la Tuberculosis". Dentro de la propuesta principal de reglas de juego que se han establecido, Simón tiene algunas habilidades o acciones que puede realizar durante la

ejecución de los mundos. Estas habilidades básicas han sido implementadas en el videojuego y son las herramientas que el jugador usa para terminarlo.

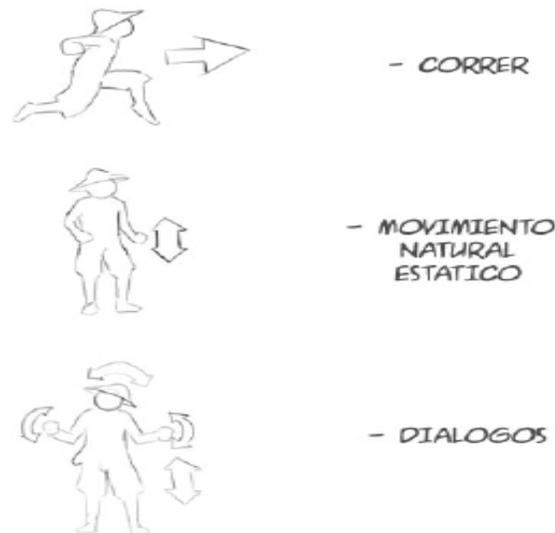


Figura 11. Estados de Simón.

La Figura 11 muestra las acciones posibles del jugador, descritas a continuación:

ACCIÓN	DESCRIPCIÓN
CORRER	Es la forma en la que Simón se desplaza por el mundo, este movimiento es controlado por el jugador mediante las teclas ARRIBA, ABAJO, IZQUIERDA Y DERECHA en el teclado. Al presionar uno de estos botones simón "Camina" en una determinada dirección dependiendo de su posición actual y de la posición de la cámara que también desempeña un papel importante en el movimiento ya que como se ha dicho el juego implementa el tipo de videojuego "Juego de Tercera Persona" y la cámara debe seguir al jugador. Así pues, al presionar FLECHA ARRIBA Simón debe alejarse de la cámara mirando en sentido contrario a la cámara. La cámara debe seguirlo describiendo el mismo trayecto. Si se presiona FLECHA ABAJO Simón debe mirar hacia la cámara y tratar de acercarse a ella. La cámara también se aleja de simón. Al presionar FLECHA DERECHA, Simón debe describir un círculo en el sentido de las manecillas del reloj con centro en la posición de la cámara, la cual se queda estática. Al presionar FLECHA IZQUIERDA, Simón debe describir un círculo en el sentido contrario al de las manecillas del reloj con centro en la posición de la cámara, la cual se queda estática. De esta forma, Simón posee un sinnúmero de posibilidades para moverse alrededor del mundo.
MOVIMIENTO NATURAL ESTÁTICO	Más conocido en el desarrollo de videojuegos como el estado "Idle" por su traducción en inglés, el "Movimiento natural" se realiza cuando el jugador no ordena nada a Simón. En este caso la animación debe ser tal que muestre a Simón en un movimiento simple y repetitivo. Estas acciones deberán ser suficientes para que el jugador cumpla con los objetivos del mundo y se narre la historia de manera concordante. También son

	las acciones que se pueden realizar para cumplir con las pruebas que se puedan imponer en los mundos, dependiendo de lo que consigne el Guión de juego. Estas pruebas están descritas en la propuesta de reglas y también una serie de Puzzles, los cuales no fueron implementados para este nivel del videojuego.
HABLAR	Al presionar la tecla H y si existe algún personaje No jugador (PNJ, en adelante) cerca que esté dispuesto a entablar una conversación. Simón realizará esta acción y lo propio realizará el Personaje que entre en charla con el Personaje Jugador (o PJ, en este caso Simón).

Tabla 10. Descripción de las acciones del personaje principal.

Paralelo a la propuesta de reglas, el Diseñador gráfico y escritor del juego ha realizado unos documentos que especifican el argumento narrativo del videojuego. Este argumento ha sido desarrollado también mediante sugerencias que se le realizaron para que se acomodara a los limitantes de OGRE, el tiempo y el grupo de desarrollo. La interpretación dentro del juego de este guión se describe a continuación.

4.2.3 Guión del Videojuego

Como se ha mencionado, el Guión del videojuego es la narración de la fantasía que plantea la historia de Simón y de la cual se debe abstraer conocimiento acerca de los Síntomas y Factores de Riesgo de la Tuberculosis Pulmonar.

A diferencia de los distintos tipos de guiones que se puedan usar, por ejemplo el guión de una película, este guión está pensado para integrar un videojuego de acción mediante conversaciones, por tanto, establece las situaciones alternativas que puedan surgir como fruto de las acciones inesperadas del jugador aparte de la secuencia propia de eventos que se espera para el flujo normal de los mismos que dan sentido a la historia. Este tipo de situaciones inesperadas son en últimas, las conversaciones que los PNJ's realizan cuando se les habla pero que no se esperan dentro de la continuidad de la historia. Además plantea de forma estructurada los objetos que forman parte del mapa de juego, los personajes y sus conversaciones y los efectos que estas tienen en el flujo de la narración, así como los objetos que se van entregando a Simón que se integran a su inventario para que se pueda continuar con la aventura. Todos estos elementos conforman el mundo de Simón.

En el documento "Guión de juego" se plantea la historia y la narración gráfica inherente a ella. En este documento se observan los parámetros gráficos que se quieren obtener con el videojuego y los elementos narrativos más importantes (ver Anexo B).

Para observar los parámetros gráficos requeridos, el diseñador ha incluido gráficas Pre-Renderizadas con un Software de Diseño 3D de cómo se plantea la interacción de Simón con el mundo, proponiendo de esta forma, la distancia a Simón de la Cámara, la escala y posición de los objetos en el mundo y otros elementos gráficos como los colores y las sombras. Cabe mencionar que no es un compromiso inexorable el cumplimiento a cabalidad de este diseño, Sin embargo, el nivel del juego está realizado para cumplir con gran parte de estos requerimientos y que se asemeje en gran medida al diseño planteado para cumplir con el enfoque artístico de este trabajo. Es el caso del mini-mapa de juego, que aunque ayuda a la visibilidad del estado de juego, no se convierte en un elemento fuerte dentro del mismo y es opcional.

Por otro lado, para apreciar los parámetros de narración del videojuego se entablan dos tipos de narración, la narración puramente gráfica, entendiendo esto como narración sin intervención del jugador o narración "Extra-Juego" y la Narración "In-Game" que se desenvuelve con Interacción del jugador y está representada mediante los mundos o escenas.

La Narración "Extra-Juego", narra los eventos fuera de los mundos, en general, estos eventos son aquellos que deben ocurrir y que sirven para contextualizar al jugador dentro de la historia para que tenga razones de cumplir los objetivos, o bien esta narración puede usarse para mostrar elementos de contenido crítico acerca de la tuberculosis. Sea cual fuere el caso, este tipo de narración es implementada mediante Stills los cuales están conformados por:

- Una imagen de fondo Estática (de ahí el nombre de Still) que muestre de forma genérica lo que se narra.
- Un texto que narre lo que se quiere mostrar.
- Audio de fondo que es una voz que narra el texto mencionado y que sirve para hacer uso del motor de audio incorporado a OGRE. En general, esta voz será la de un narrador o en este caso narradora, pues es ORAH, un espíritu benigno que acompaña y aconseja a Simón en sus aventuras, este PNJ es descrito en el documentote (ver ANEXO B)

Todos los Fragmentos de una misma línea incidental en la narración "Extra-juego" conforman un StillSet, la secuencia de este StillSet va pasando cuando el jugador presiona la tecla ESPACIO hasta que termina.

Por otra parte, la conformación de los mundos para la narración "In-Game", Están explicados mediante tablas explicativas del guión de juego y mapas que explican la posición de los elementos y los factores críticos para la correcta consecución de los objetivos en la historia. Esta estructura permite diseñar el mundo que será mostrado mediante el motor de juego (explicado más adelante). En esta descripción de los mundos se muestran los PNJ's los objetos de Mundo como casas, árboles o cercas que se encuentran en el lugar para esta escena y que influencia tienen en el desenvolvimiento de la historia. Además presenta las restricciones de control del jugador las convenciones usadas y otros detalles a tener en cuenta en el desarrollo del juego.

3. Contextualización del jugador con el juego:

- En este momento aparece en pantalla la interfaz de juego.
- Primero que todo aclaro que en este punto inicia el uso de cuadros para una guía más precisa y práctica.
- En el caso de los mapas: las zonas amarillentas son las zonas donde se desarrolla la acción. Y en el caso de los enemigos y amigos los distinguimos por los siguientes íconos. Los elementos como las piedras y las cercas son obstructores de los recorridos.



Figura 12 Ejemplo de Convenciones usadas en el planteamiento de una escena

La Figura 13, muestra el planteamiento de la estructura de la primera escena y que es el punto de partida de los mundos en el videojuego. Se hace énfasis en la disposición de los elementos y PNJ's en el mapa que forman parte importante para la historia. La zona amarilla es una marca de la zona por donde se puede mover Simón en esta escena para sectorizar la parte donde ocurre este fragmento de la historia.

	iniciar el diálogo. * El jugador no puede pasar al punto 3.1.C hasta que ha finalizado la conversación con su mamá. 3.1.B. Simón se encuentra con el ciudadano 1 (Cno.1) y con el soldado imperial 1(Sdi.1) después de haber hablado con su mamá. 3.1.C. Simón debe hablar con el Cno.1 primero para que el Sdi.1 lo deje avanzar. Hasta que no lo haga en ese orden saldrá un diálogo genérico del Sdi.1.	
RESTRICCIONES Y POSICIONES DEL MAPA:		
PERSONAJES DEL MODULO:	- Esperanza(sólo los diálogos) - Simón	- Cno.1 - Sdi.1
DIALOGOS FIJOS:	<ul style="list-style-type: none"> • Caso 3.1.A: 1- Esperanza: ¡Simón ven! • Cuando Simón se acerca: 2- Simón: ¿Sí mamá, qué necesitas? 3- Esperanza: Ven, necesito decirte algo en voz baja... 4- Esperanza: Simón, ¿recuerdas lo que te dije hace una semana el día de tu cumpleaños? 5- Simón: ... Si mamá, me dijiste que ya era lo suficientemente grandote para ayudarte un poco más con “nuestros amigos”. 6- Esperanza: ¡Shh! Simón, se mas cauteloso cuando te refieras a “ellos”... 7- Esperanza: Precisamente a eso me refería Simón, Hay alguien que nos necesita: Don Nicolás... 8- Esperanza: ¡El de la casa alta al lado más hacia el este del pueblo!, por favor se cauteloso y tráelo! 9- Simón: ¡Dalo por hecho mamá! 	
DIALOGOS LIBRES:	<p><u>Cno.1</u></p> <p>Sí Simón se acerca a Cno.1 antes de hablar con su madre (3.1.A):</p> <p>1- Cno.1: ¡Coff!, ¡Coff!.. Oye niño, ve a ver a tu madre ¿no escuchas que te está llamando? estos niños de ahora...</p> <p>Sí Simón se acerca a Cno.1 después de hablar con su madre (3.1.B):</p>	<p><u>Sdi.1</u></p> <p>Sí Simón se acerca a Sdi.1 antes de hablar con su madre (3.1.A):</p> <p>1- Sdi.1: Grrrr... ¡ATRÁS!, está prohibido el paso por aquí.</p> <p>Sí Simón se acerca a Sdi.1 después de hablar con su madre (3.1.B):</p> <p>1- Sdi.1: ¿No has entendido pequeño engendro que no</p>

Figura 13 Planteamiento de una Escena con mapa y diálogos.

Aunque para tener una visión total del guión de juego se debe referenciar el ANEXO B, se presenta aquí un resumen para facilitar la visión de la abstracción que se realizó para diseñar el Software.

Para la primera escena, el jugador es contextualizado por una voz narrativa en el mundo de "San José de las piedras" esta voz es la de Orah, un espíritu del bien que se presenta en forma de una cachorra de perro. "San José de la piedras" es un pequeño pueblo donde hace algún tiempo reinaba la paz. Sin embargo el terror se viene difundiendo por una enfermedad y un malvado emperador de nombre Kash. Simón es un niño de este pueblo que vive con su mamá y que de forma clandestina, ayuda a las personas enfermas en el pueblo. Esta actividad es realizada a escondidas de los guardias imperiales.

Como primera misión Simón debe rescatar a uno de sus vecinos, para llegar a él debe persuadir a uno de los guardias. Al terminar esta misión son sorprendidos por los guardias y es ahí donde Orah aparece para aconsejar a Simón en el camino de salvar a su pueblo. En ese Camino debe cumplir una prueba que consiste en terminar un laberinto evadiendo a los guardias. Después de estas acciones Simón debe empezar a conocer acerca de los síntomas y factores de riesgo de la tuberculosis pulmonar que es lo tratado en este primer nivel del videojuego. A partir de aquí Simón emprende un viaje en el tiempo para aprender de la enfermedad y salvar a su región pero son situaciones que se salen de la primera escena que es el escenario escogido para el primer nivel del videojuego.

Ya conocido el guión y la propuesta de reglas del videojuego se pasa a realizar el Análisis y Diseño de Software Orientado a Objetos, implementando una arquitectura sobre .Net y MOGRE usando la plataforma de audio que ya se construyó.

4.3 DISEÑO DE SOFTWARE DEL VIDEOJUEGO

Ya planteados los requerimientos de funcionalidad del videojuego, se procede a la etapa de Diseño del Software que los soportará. Este Software comprende dos partes, el motor de videojuego y los parámetros de videojuego que se utilizarán para que "Simón y la tuberculosis" luzca tal cual se diseñó.

A continuación se presenta el Diseño de Software que se realizó para posteriormente implementar el programa que corre el Videojuego "Simón y la Tuberculosis". Se realizará un Sondeo por los artefactos y diagramas construidos y su correspondiente explicación e

influencia en el producto final. Estos artefactos son parte fundamental además de ser el componente comunicativo y documental de la arquitectura e ingeniería del juego.

4.3.1 Arquitectura del Videojuego

La Arquitectura diseñada para este videojuego está soportada por la plataforma de Windows XP. Dado que está diseñado sobre .Net se ha usado la plataforma de Gráficos 3D Orientada a Objetos (OGRE) con el wrapper de código Administrado MOGRE que a su vez está soportado por el FrameWork de .Net.

El Diseño considera el módulo de audio ya implementado para realizar el experimento de su funcionalidad en el videojuego.

Se ha diseñado un módulo auxiliar denominado PlopOgre el cual contiene clases y Servicios para administrar la interfaz Gráfica de Usuario, así como el manejo de la librería de Audio y el manejo de la entrada desde teclado y Mouse usando MOIS, que es una plataforma de código administrado para el manejo de la entrada. PlopOgre cuenta con algunas Utilidades adicionales que serán explicadas más adelante.

PlopOGRE por otra parte, es la herramienta que se encarga de la implementación gráfica y funcional de los controles de usuario que se muestran en el menú principal, como botones y ventanas y el manejo de los eventos que se desencadenan al presionarlos o usarlos con el Mouse. Este módulo aparte de encargarse de la IGU ⁷del menú cuenta con funcionalidades para administrar los forms del juego como mecanismo para el cambio de interfaz entre el juego como tal y los otros estados (opciones de configuración y Main Menu).

⁷ IGU O GUI es la sigla para Interfaz Gráfica de Usuario o Graphical User Interface. Es el artefacto tecnológico de un sistema interactivo que posibilita, a través del uso y la representación del lenguaje visual, una interacción amigable con un sistema informático

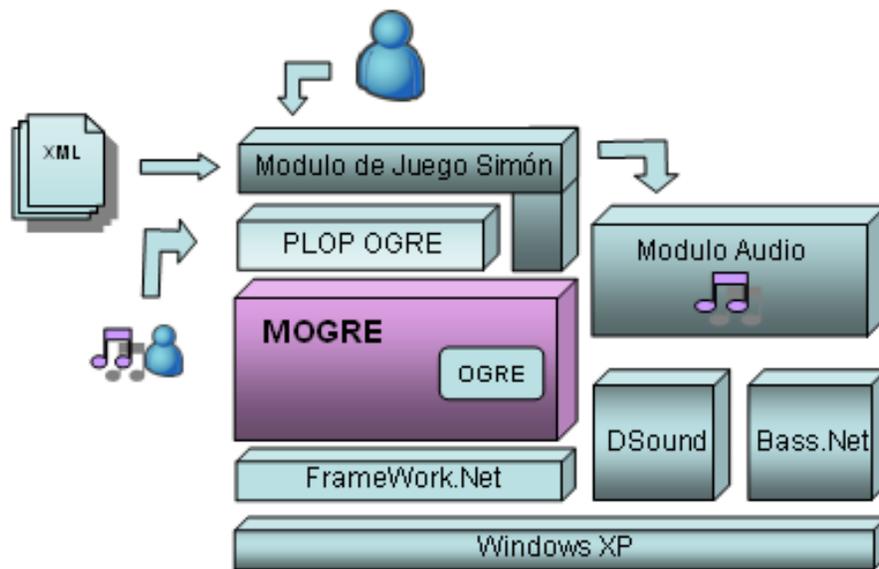


Figura 14. Diagrama de Capas de la Arquitectura.

La Figura 14 muestra el diagrama de capas Genérico de la Arquitectura del Videojuego. Se puede Observar que los elementos interrelacionados se soportan sobre .Net, MOGRE y como sistema operativo Windows XP. Se hace referencia al módulo de Audio que es usado por parte de OGRE y del módulo de juego que dentro del proceso de construcción se llamó simplemente "Tuberculosis".

Se aprecia que la interacción del jugador es directamente con el módulo de Juego y que es este mismo quien se encarga de realizar la muestra de la información mediante la IGU y los elementos gráficos del entorno del juego como texturas o modelos entre otros.

El diagrama también refleja el manejo de archivos para la ejecución del videojuego. Este manejo de persistencia está orientado a permitir poner los parámetros propios del juego, como elementos gráficos (construidos por parte del diseñador gráfico) como modelos, imágenes, texturas o elementos de audio como los sonidos de fondo, la ambientación o las voces de los PNJ's. Estos parámetros deben ser independientes del motor y este último debe encargarse de interpretarlos y dibujarlos de manera adecuada.

Se observa en el diagrama el manejo de archivos en XML, estos archivos permiten la extensibilidad y flexibilidad del videojuego. Más adelante se observará que son estos archivos

los que permiten diseñar los mundos y otros elementos del juego de forma que se ejecuten en tiempo de ejecución sin tener que cambiar el código fuente.

A continuación se describen estas capas con más detalle y posteriormente se describirá más a profundidad las capas más importantes.

NOMBRE DE LA CAPA	DESCRIPCIÓN
CAPA FRAMEWORK .NET Y WINDOWS	Sólo se indican aquí para demostrar el soporte de plataforma del videojuego.
DSOUND Y BASS.NET	Como se indicó, el módulo de audio está soportado por estas dos librerías.
MÓDULO AUDIO	El módulo de audio que se integra a OGRE y parte fundamental de este proyecto.
MOGRE(OGRE)	El Wrapper que se encarga de convertir a código administrado las primitivas de OGRE que es el motor de renderizado 3D.
PLOPOGRE	Es la capa de utilidades de IGU que contiene clases y servicios varios para el manejo del videojuego.
MÓDULO DE JUEGO SIMON (TUBERCULOSIS)	En esta capa se encuentran clases y servicios que implementan el juego y que pretenden cumplir a cabalidad el diseño del mismo.

Tabla 11. Descripción de las Capas de la Arquitectura.

La Figura 15 muestra una descripción más profunda de esta capa. Esta capa administra algunos elementos de la parte gráfica y la lógica del videojuego así como el acceso persistente a los archivos que sean parámetros para el juego.



Figura 15. Diagrama de Capa "Módulo de Juego"

Dentro de la capa de Juego se tienen algunas clases que sirven de utilidad como una clase Gestor Sonido como fachada para los servicios del módulo de Audio; la clase Modelo que

encapsula los servicios de los modelos 3D en MOGRE. El componente de configuración se encarga de cargar desde XML los aspectos de opciones de Configuración iniciales del videojuego.

En esta capa se describen las Forms que son una solución que se encontró al problema del manejo de los estados dentro del juego. Las Forms Emulan el comportamiento de las Forms de Windows Forms en cuanto a que también se muestran como ventanas, poseen controles y tiene la capacidad de aparecer y desaparecer (Show & Hide).

Esta Capa posee dos subcapas, Lógica y Persistencia. Lógica es la capa que se encarga del manejo de la lógica del juego, esta capa contiene las entidades propias para la implementación de los Mundos, los estados In-Game, PNJ's, Simon Orah Conversaciones entre otros que serán descritos más adelante. La finalidad del paquete de Lógica es tratar de encapsular el comportamiento del videojuego como entidades que se representan en el juego. Y la capa de Persistencia se encarga del acceso a datos de parámetro para incluir en el juego el diseño de los mundos de los PNJ's y en general los elementos dentro del mundo de Simón.

La Figura 16 muestra los componentes esenciales de la capa de Lógica, Además se observa que estos elementos reciben parámetro desde archivos XML. Se puede observar una descripción más detallada en la Tabla 12.

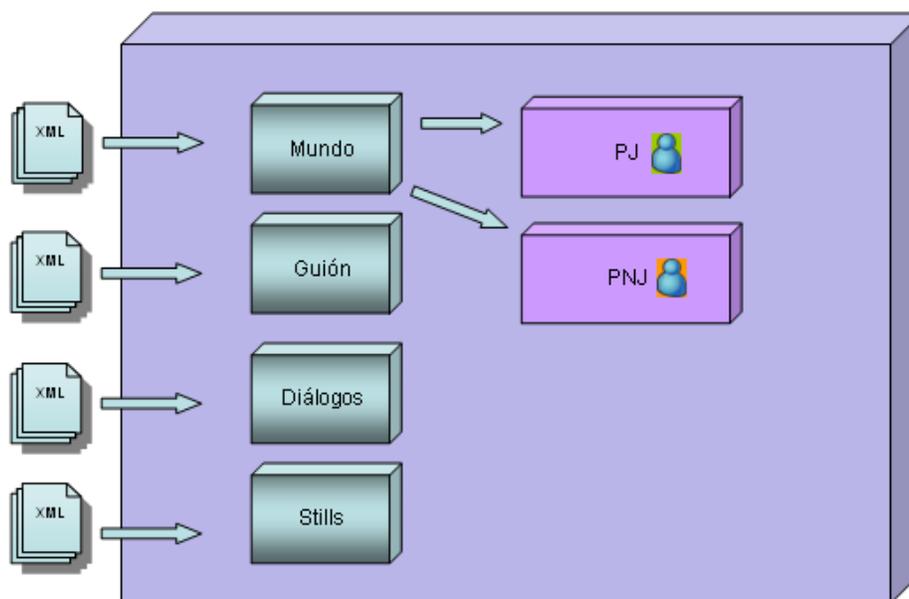


Figura 16. Diagrama del Paquete de Lógica

COMPONENTES CAPA LÓGICA	DESCRIPCIÓN
MUNDO	El mundo es la escena que se muestra, se compone de Objetos de escena, PNJ y PJ's que en este caso es sólo Simón, además se puede usar en la escena a Orah quien sigue constantemente a Simón.
GUIÓN	El guión describe de forma secuencial los Stills narrativos que se deben poner en pantalla y los mundos que se van a jugar.
DÍALOGOS	Esta parte carga y administra las conversaciones dentro de un mundo.
STILLS	Administra los Stills del juego.
PJ Y PNJ	Entidades correspondientes al comportamiento y estado de los PNJ's y Simón en el juego.

Tabla 12. Descripción de los componentes de la capa lógica.

Con la arquitectura definida, los siguiente subcapítulos describen el diseño e implementación de esta arquitectura, traducida a Diagramas de Clases y código fuente.

4.3.2 Patrones de Diseño Aplicados y Diagrama de Clases de la Arquitectura

Ya que se ha definido la arquitectura del videojuego se procede a mostrar la parte de Diseño de Software construida. Para el Diseño de "Simón y la tuberculosis" se tuvo en cuenta varios aspectos, por un lado las entidades del juego necesarias para implementarlo y el estudio de la plataforma MOGRE para utilizar sus elementos de Interfaz Gráfica en 3D, por supuesto también se tuvo en cuenta los servicios que ofrece el módulo Audio que ya había sido implementado en esta fase del desarrollo.

Este subcapítulo pretende mostrar el Diagrama de Clases que se obtuvo después de realizar varias iteraciones a través de la construcción del videojuego, siempre manteniendo la idea de Analizar, Diseñar, Implementar y probar las diferentes partes que poco a poco se iban construyendo hasta tener la última versión del nivel del motor de videojuego.

Este Diseño, como se ha mencionado, está orientado a crear un motor de Videojuego que use parámetros para funcionar y que implemente la arquitectura ya descrita. Esto con el fin de que al final el producto sea un software altamente extensible y muy fácilmente configurable mediante XML, de esta forma si en un caso dado se logra construir la primera escena, en las subsecuentes escenas no se tenga que codificar nada en absoluto sino que se especifiquen en los archivos dispuestos a ello. Esta misma filosofía aplica en varios aspectos como los stills del videojuego, los elementos gráficos como los modelos y las texturas, las conversaciones en el juego, los estados del juego y el guión mismo.

Dado que se quiere alcanzar un alto grado en la calidad del software, el diseño cuenta con la implementación o uso de algunos patrones de Diseño de Software que se usaron para permitir bajo acoplamiento y alta cohesión los cuales se mencionaran en su momento.

Se mostrará el diseño realizado y cómo, éste concreta a la arquitectura. La herramienta de modelado utilizada para los diagramas de clases fue Power Designer 12.1. Fragmentos de estos diagramas se utilizarán en la presente monografía para facilidad en la lectura.

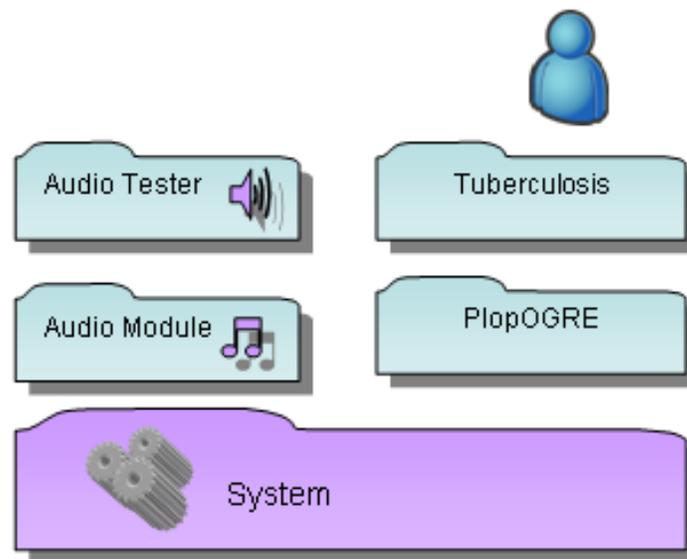


Figura 17. Diagrama de Paquetes de clases. Visión de la Arquitectura

La Figura 17 muestra el diagrama de paquetes de arquitectura, básicamente estos paquetes contienen clases que componen el diseño que cumple con los objetivos que se plantean en la arquitectura. Estos paquetes serán mapeados a namespaces al momento de implementación.

En este Capítulo nos centraremos en PlopOgre y Tuberculosis pues son estos los paquetes que presentan el diseño del videojuego como tal. Los otros paquetes son:

System: Representa las clases del namespace System de .net. Este paquete y su disposición simbólica en la parte inferior del diagrama representan que el motor utiliza las características de la plataforma .net.

AudioModule: Es el módulo de audio mencionado en el capítulo anterior.

AudioTester: Este paquete tiene las clases para implementar una aplicación que comprueba los servicios de AudioModule (intrascendente para este capítulo).

PlopOgre: Este módulo es uno de los más importantes, debido a que es el encargado de manejar las operaciones básicas de producción de audio, entrada de información desde teclado y los dispositivos de posicionamiento.

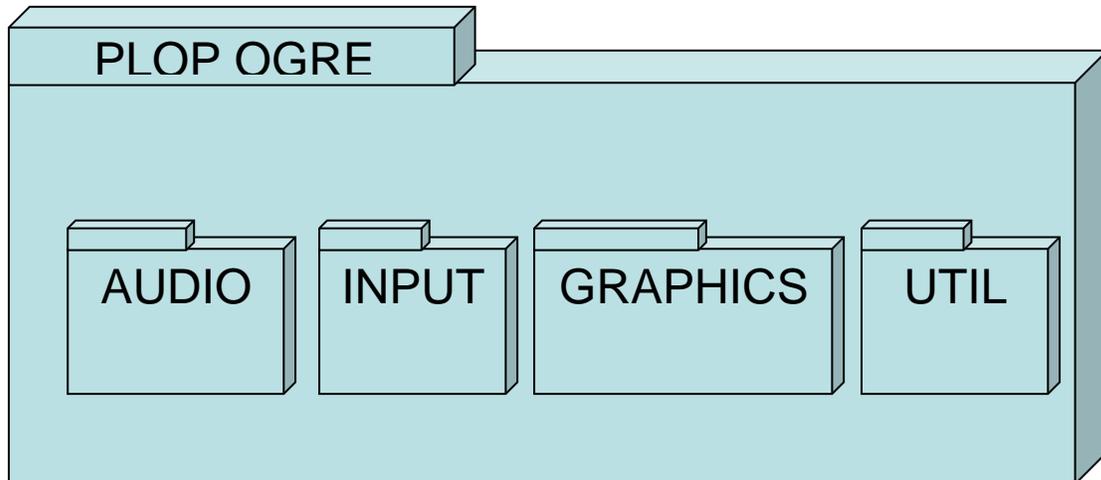


Figura 18. Paquete de PlopOgre

Módulo de Juego Tuberculosis

Este módulo muestra la vista compuesta del paquete de Tuberculosis, se hace énfasis en la vista compuesta ya que posee dos subpaquetes como se había mencionado en los submódulos de la arquitectura. Estos dos paquetes se reflejan en la Figura 15 y contienen su propio diagrama de clases. Estos paquetes son Lógica Y Persistencia. No obstante, el paquete tuberculosis también posee sus propias clases.

A continuación se describirán los fragmentos de los diagramas de clases más relevantes, describiendo su funcionalidad y el papel que cada clase cumple en la aparición de un patrón de Diseño.

Paquete PlopOgre

Este paquete está compuesto de cuatro módulos que se acoplan con OGRE para ofrecerle al desarrollador funciones de reproducción de sonido, entrada de información, manejo de gráficas y animaciones para opciones de menú.

- **Audio:** Dentro de este módulo se encuentran el modelo de audio desarrollado para ofrecer al motor las funciones básicas de reproducción de sonido con efectos de audio explicado en el capítulo anterior.
- **Graphics:** Este módulo hace uso de la librería cGUI y ofrece un conjunto de objetos que permiten el desarrollo de interfaces de usuario con opciones de menú.

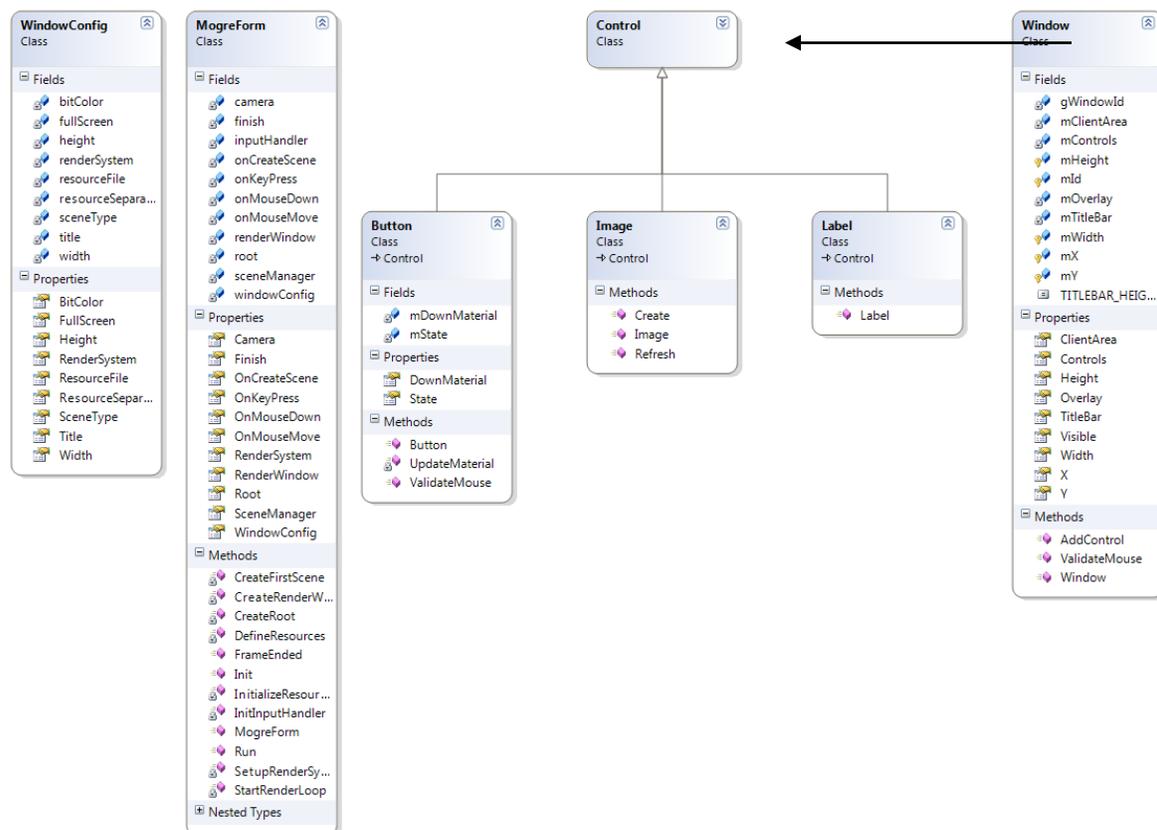


Figura 19. Muestra las clases del módulo Graphics

- **Input:** Este módulo implementa un conjunto de manejadores para la captura de eventos tanto de teclado como de dispositivos.

apuntadores.

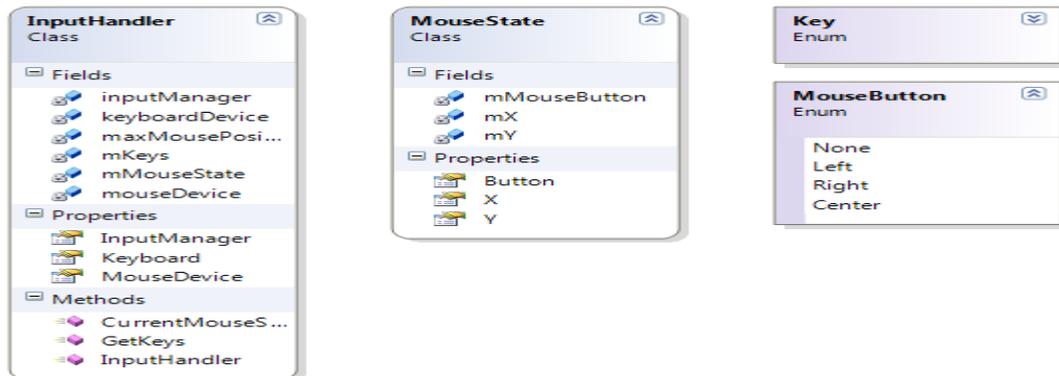


Figura 20. Muestra las clases del módulo Input

- **Utils:** Este módulo ofrece apoyo a las tareas comunes y es usado para almacenar registro de eventos y errores para audio.

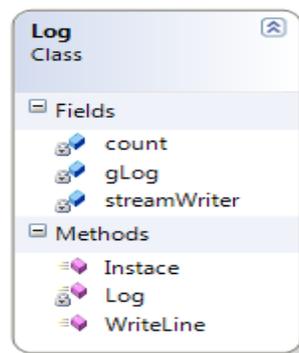


Figura 21. Muestra la clase del módulo Utils

Sección OnGameState

En esta sección se muestran las clase que tienen que ver con el manejo de los estados dentro del juego propiamente dicho, es decir los estados que se pueden tener mientras se está en la narrativa del juego por fuera de los elementos del Menú. La Figura 22, muestra la sección donde esta es una adaptación del patrón GameState [46] en su implementación para OGRE.

IOnGameState: Esta interface define los métodos que debe cumplir una clase que implemente el comportamiento de un estado a saber:

- **Entrar:** Determina lo que se debe hacer cuando se entra en un estado.
- **Salir:** Determina lo que se debe hacer cuando se sale de un estado.
- **Update:** Determina el comportamiento del estado a cada actualización de cada frame.
- **Procesar:** Dada la tecla, este método implementa el comportamiento del estado dado el evento de haberse presionado esa tecla.

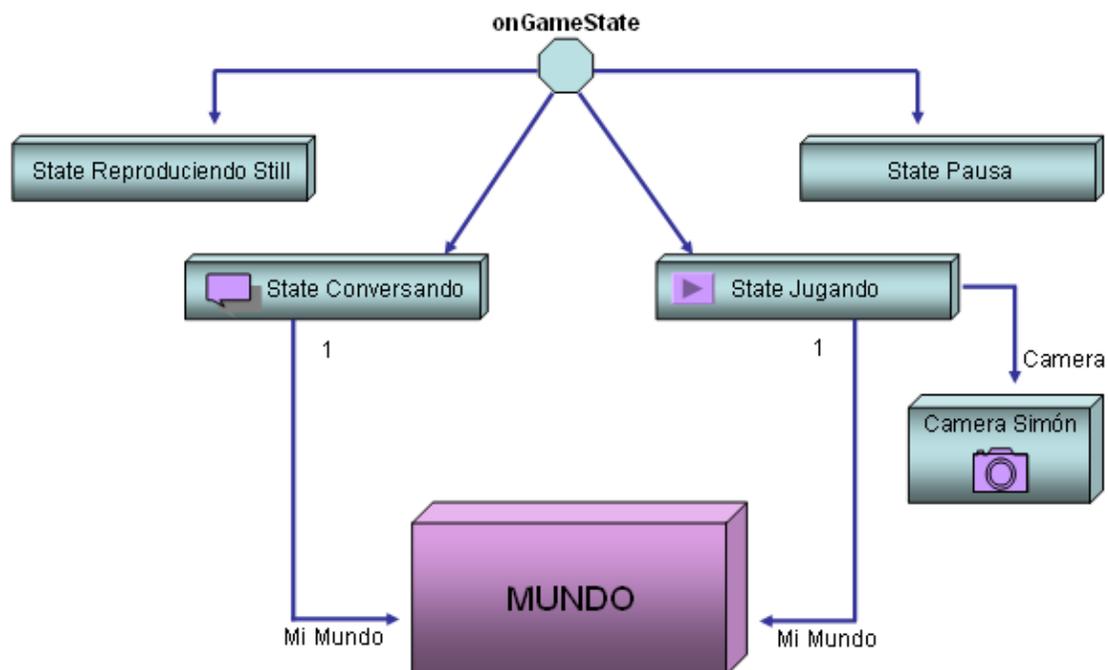


Figura 22. Sección OnGameState

Se han implementado cuatro estados en el juego, estos son:

Jugando: Clase StateJugando, Implementa el comportamiento del juego cuando Simón está de forma normal en el mundo. Con forma normal, se trata de referir al estado en el cual Simón anda por el mundo pero sin interactuar con los PNJ's. Esta clase es la que se encarga de interpretar las acciones del juego y el comportamiento y habilidades de Simón como se especificó en la sección 4.2.2. De esta forma, esta clase se encarga de actualizar a Simón y hacerlo caminar en las 4 diferentes direcciones según el jugador prefiera. Este estado actúa como un estado maestro que controla el acceso a los otros estados y actualiza el mundo de tal

forma que este procese las otras reglas de juego que no tiene que ver con la entrada del jugador, como el movimiento de Orah, este comportamiento se verá en la clase Mundo.

En este estado si se presiona la tecla H y hay un PNJ para hablar se entra en el estado Conversando. Si se presiona la tecla ENTER se entra en el estado pausa. Si se presiona ESCAPE se puede volver al menú inicial.

Conversando: Clase StateConversando, Este estado es aquel que interviene cuando Simón ha decidido hablar con un PNJ que este dispuesto y cerca para hablar. En este estado Simón entra en el estado "Hablar", así como el PNJ. Aquí se muestra un cuadro de conversación donde aparece la cara del jugador y el texto de diálogo y se reproduce el sonido que va asociado al texto. Esto se verá más a fondo en la sección de Conversaciones. En este estado el método procesar avanza los diálogos presionando la tecla ESPACIO o FLECHA DERECHA.

Reproduciendo Still: StateReproduciendoStill, Este estado interpreta el estado del StoryBoard en la Figura 10, Stills de Conexión Argumental, este estado muestra en el fondo una imagen de la narración, el texto narrado, el audio narrado y un audio de ambientación. En este estado, el método procesar avanza cada Still del StillSet presionando la tecla ESPACIO o la tecla FLECHA DERECHA.

Pausa: Clase StatePausa, en este estado el juego se detiene y muestra una pantalla que dice "Pause", el mundo no se actualiza y el único proceso que se realiza es que si se presiona ENTER se vuelve a Jugando.

Sección Mundo

El manejo del mundo, como se ha dicho en reiteradas ocasiones, es muy importante para la vida del videojuego, el mundo está representado por esta sección y a continuación se describen las clases que lo conforman. La Figura 23, muestra el fragmento del diagrama de clases.

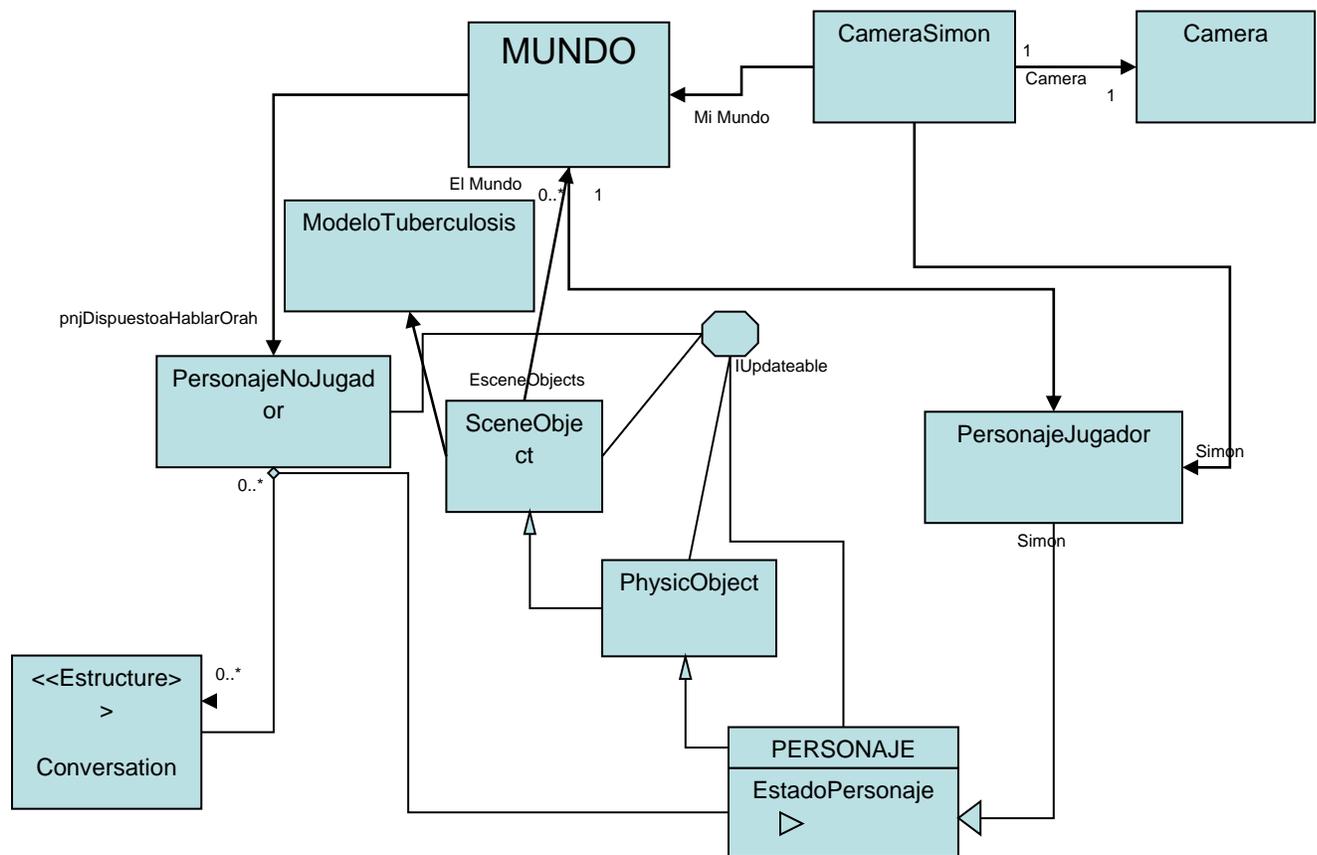


Figura 23. Sección mundo.

Mundo: Esta Clase es el núcleo de esta sección. El mundo funciona como la implementación del patrón Controlador, dado que se encarga de administrar lo referente al mundo donde vive Simón. Este mundo se encarga de administrar los objetos de Escena para lo cual cuenta con una colección de objetos llamada SceneObjects. El objetivo de esta colección de objetos es llamar a los métodos de actualización de cada objeto independientemente de su tipo ya que todos los sceneObjects implementan a IUpdateable de esta forma se está haciendo una implementación del patrón Polimorfismo. La clase mundo se encarga de administrar el terreno y los parámetros gráficos del mundo en sí como el cielo que se va a pintar, el terreno que se va a usar, el tamaño del terreno, las alturas del terreno y el audio de fondo. Esta configuración del mundo se realiza en el método Initialize. Un mundo es instanciado por la clase frmJuego de PlopOGRE. Y parametrizado mediante XML.

La clase Mundo también se encarga de otros aspectos concernientes al mundo de Simón como la detección de colisiones que básicamente es lo que evita que Simón penetre otros objetos, es

decir, que Simón se tropiece con los otros objetos y para esto se usa la detección de colisiones de MOGRE. En este punto se encontró un problema dado que al principio se creía necesario utilizar la librería física Newton lo cual fue bastante difícil tanto por el complejo manejo de la API como por que Newton no es una plataforma altamente documentada, además los resultados que se obtenían al aplicar las leyes físicas en el mundo de Simón no eran los esperados en cuanto a que se obtenían efectos demasiado estilizados para la simpleza relativa del juego, entre estos efectos dados encontramos la gravedad y las colisiones demasiado realistas para lo que se necesitaba.

En este sentido, se decidió no implementar físicas y dado que sólo se necesitaban colisiones muy básicas, el manejo de estas se realizó mediante las BoundingBox de los modelos de los objetos y los Personajes. Una BoundingBox se puede ver como una caja invisible que rodea a los modelos de los objetos en todo su tamaño. OGRE ya viene con soporte para detectar colisiones entre BoundingBoxes.

A partir de esto surgió otro problema y es que las BoundingBoxes comprenden no sólo el modelo en cuanto al mesh que se está pintando en un momento dado, sino también al volumen que comprenden todas las animaciones. De esta forma, si hacemos un modelo que tenga una animación que se aleje mucho del origen, la bounding box comprenderá bastante volumen, más del que se muestra en el esqueleto del modelo y de esta forma parecería que Simón colisionará aún con las partes vacías de los otros objetos. Para solucionar esto, se convino con el Diseñador Gráfico en no realizar animaciones muy grandes para que el manejo de colisiones sea más natural.

Por otra parte Mundo tiene una referencia a un PNJ llamado pnjDispuestoaHablar y detecta eso si hay algún personaje cerca y que tenga alguna conversación disponible que cumpla una condición para desencadenarse, Mundo guarda una referencia de este personaje para ser usada en el estado StateConversando.

Además Mundo posee otras dos referencias a Personajes, es natural que se tenga una referencia a un PJ llamada simón para que se conozca el personaje principal y otra a un PNJ que es Orah.

Mundo se encarga de una parte importante y es cargar los datos de él desde un archivo XML cuyo nombre se otorga. Este método también se encarga de Delegar la carga de conversaciones al gestor de Conversaciones y la carga de las flags al gestor de Condiciones.

Mundo es un buen administrador que cumple con sus funciones pero delega a otros, otras responsabilidades.

A continuación se describen las clases que cumplen con el papel de lo que llena el mundo y lo compone.

IUpdateable: Debido a que en OGRE se debe actualizar los objetos a cada frame que se ejecute y para cumplir con la fase de actualización del patrón de Game Loop[47], IUpdateable es una interfaz que define los métodos para actualizarse Update y para Inicializarse Initialize de un objeto dentro del mundo.

SceneObject: SceneObject es una clase base para los objetos dentro de un mundo, además es parte importante de la implementación de Polimorfismo que se hace aquí. Básicamente SceneObject representa a un objeto dentro del mundo que no es un PJ o PNJ, tiene un nombre y un modelo 3D que lo representa, este objeto debe tener una Posición en el mundo, la escala que define su tamaño y la orientación que define hacia donde está viendo el objeto. La referencia a la clase Modelo es para manipular el modelo asociado al SceneObject y al modelo propio de OGRE.

PhisycsObject: En realidad, esta es una clase que se puso cuando aún se pensaba usar Newton para el manejo de Físicas, sin embargo se conservó la clase debido a que puede ser de ayuda si en futuras versiones se quiere implementar funcionalidad de objetos que cumplan con características físicas como la gravedad.

Personaje: Es una clase Base para los métodos y atributos de un Personaje en el juego, es la clase base para los PNJ y los PJ. Los personajes tienen el nombre de la cara que es el nombre de un bitmap para que se pinte en el estado Conversación cuando Simón está hablando o el PNJ está hablando. Cuenta con Rapidez que es la rapidez con la que camina el personaje en el mundo. Dirección es la dirección en que se mueve. Y Velocidad es la velocidad como producto de la rapidez y la dirección. Un personaje puede Hablar, Caminar y Descansar. Un personaje debe ir pegado al piso lo que quiere decir que un personaje al actualizarse debe estar a la altura del terreno en la posición en la que está.

PersonajeNoJugador: Un PNJ es un personaje que no es Simón, la característica que define a los PNJ es que no son controlados por el jugador y a diferencia de Simón tienen un posible destino al que deben ir directamente si la bandera CaminarADestino se activa. Otra

característica de los PNJ es que tienen una referencia a una colección de ConversacionRef que son conversaciones que pueden ejecutarse si se cumple una condición. Si una de estas conversaciones se cumple y simón se acerca lo suficiente al PNJ un globo aparece en la cabeza del PNJ indicando que está dispuesto a hablar.

PersonajeJugador: Esta clase es la clase del objeto que representa a Simón. En ella hay un método que verifica si Simón no se ha salido del mundo sirve para controlar que el jugador sólo juegue en los límites del mundo.

CameraSimon: Esta es la cámara que sigue a Simón y se puede hacer atrás a la orden del jugador, dicha cámara al igual que los Personaje va pegada al suelo y da la sensación de que el jugador que esta en visión de tercera persona va también caminando detrás de Simón y da muestra del efecto de los altibajos del terreno. Esta clase implementa el comportamiento de la cámara como se mostró en la sección 4.2.2.

La referencia al Objeto Camera que se ve en el diagrama es para manejar la cámara propia de OGRE.

Sección Conversación.

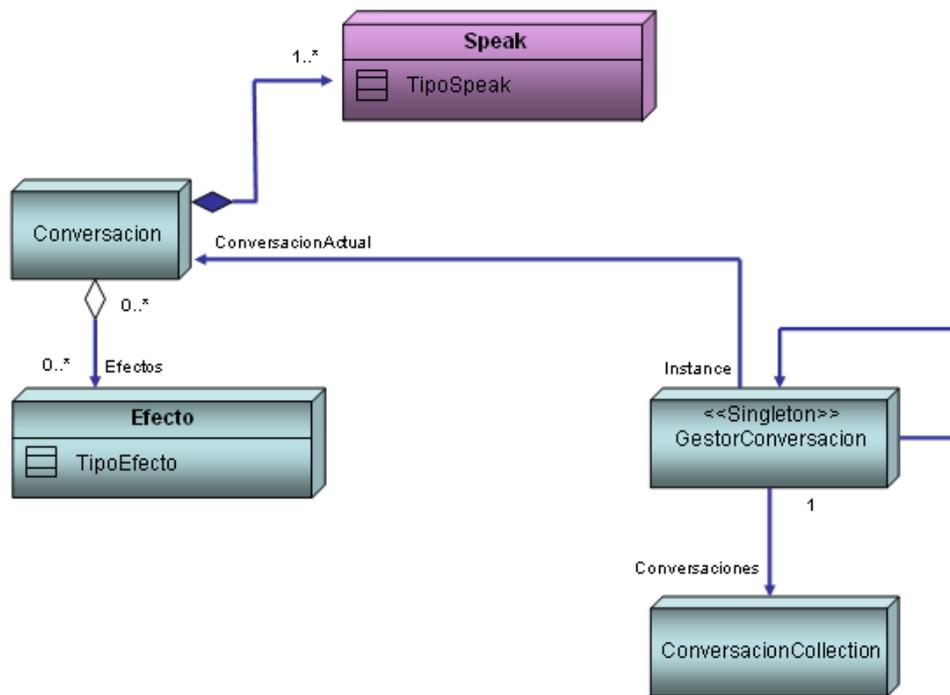


Figura 24. Sección Conversación.

La Figura 24 muestra el fragmento del diagrama de clases que se explica aquí. Las conversaciones son importantes dentro del videojuego en cuanto explican y narran la historia. Una Conversación aparece cuando el jugador desea hablar con un PNJ como se explicó anteriormente, en el diseño del mundo se deben haber especificado las conversaciones.

Este modelo utiliza como base la propuesta documental de documentos en XML[48]. Las clases descritas a continuación implementan las estructuras de datos para leer y utilizar los datos en XML.

Conversación: Una Conversación es un diálogo que se tiene entre Simón y un PNJ, las conversaciones están compuestas por Speaks, un Speak es una parte del diálogo que puede ser interpretada o bien por el PNJ o por Simón esto se especifica en el tipo del Speak.

Una conversación después de ejecutarse puede desencadenar una serie de Efectos, un efecto desactiva o activa una Flag que a su vez hace que el estado del Juego cambie y las condiciones a cumplir sean diferentes. La sección de Condiciones explica mejor las Flags y las Condiciones.

Gestor Conversaciones: Esta es una implementación del patrón de diseño singleton (por eso la referencia a sí mismo como instancia única Singletón), esta clase se encarga del manejo de las conversaciones, configurar el marco de conversación, reproducir el audio de conversación y efectuar los efectos de la conversación que cumpla la condición dada de un determinado PNJ.

ConversacionCollection: Es una implementación de la interface List y hereda de CollectionBase para determinar una colección de Conversaciones, útil para el manejo de estas.

Sección Condiciones

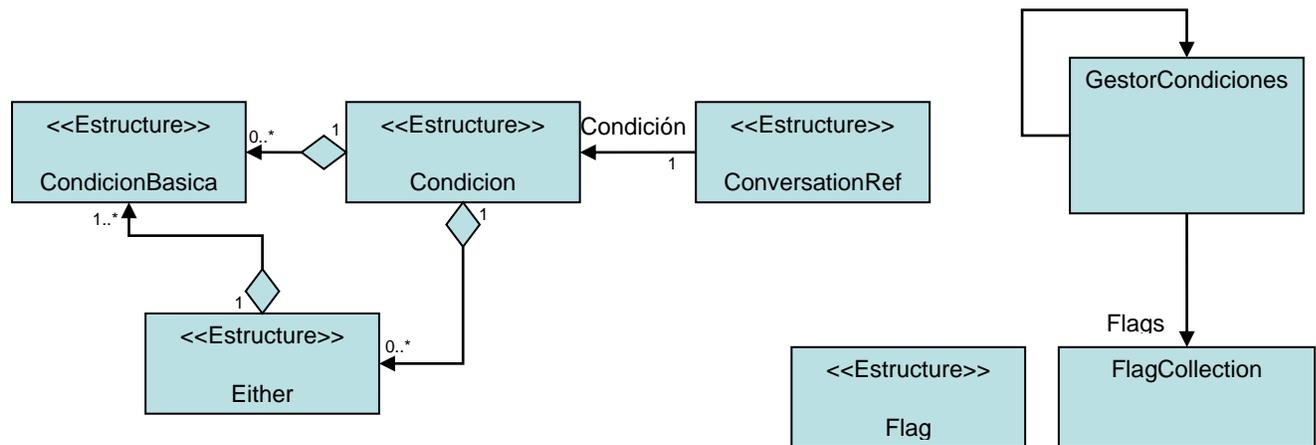


Figura 25. Sección Condición

La Figura 25, muestra el fragmento del diagrama de clases. En esta sección se implementa la idea de condiciones que se define en la propuesta documental de e-engine [48]. Las condiciones sirven para cambiar el estado de la historia del juego de esta forma las acciones de los PNJ y sus conversaciones cambian en función del estado del juego.

Condición: Esta estructura de datos representa una condición que puede cumplirse o no dependiendo del estado de las flags. Una Condición Se compone de una lista de CondicionBasica. Cada condición básica es una condición atómica que solo pregunta si una determinada flag está activa o no, El nombre de la flag se especifica en el atributo nombreflag de CondicionBasica y el tipo de condición en el atributo booleano activa, si activa es true la condición se cumple si la flag es activa, de lo contrario se cumple si la flag es inactiva.

La Condición también se compone de un Either, que es un contenedor para otra Colección de CondicionBasica.

La clase Condición tiene un método que se llama SeCumple, este método retorna verdad si la condición se cumple, de lo contrario retorna falso. Una Condición se cumple si las todas las Condiciones básicas dentro de la colección de condiciones Básicas y se cumple al menos una de las condiciones básicas dentro de la colección del Either relacionado.

Flag: Esta estructura de datos representa una bandera en el videojuego, las banderas son similares a variables booleanas y estas definen el estado de la historia del juego. Las Flags se definen en la propuesta documental de e-engine [48]. Una Flag tiene un nombre y un atributo

booleano llamado "activa" que define el estado de esta flag. Si este atributo es verdad la flag se considera activa, de lo contrario se considera inactiva. Las flags de un mundo se definen previamente en el diseño de este.

FlagCollection: Implementa List y CollectionBase como utilidad para el manejo de Colecciones numerosas de Flag.

GestorCondiciones: El gestor de condiciones es una clase Singleton que contiene métodos para:

- Activar una flag, dado el nombre.
- Desactivar una flag, dado el nombre.
- Obtener el estado de una flag.
- Determinar si una Condición se cumple en base a sus Condiciones Básicas y el estado de las flags en ese momento.
- Cambiar el estado de una flag.
- Cargar las flags de un mundo dado el nombre del archivo XML.

ConversacionRef: Una referencia a una conversación contiene el nombre de una conversación determinada y una referencia a una Condición, cuando Simón hable con un PNJ, este último conversará con la primera conversación en la colección, cuya condición se cumpla.

Sección Still

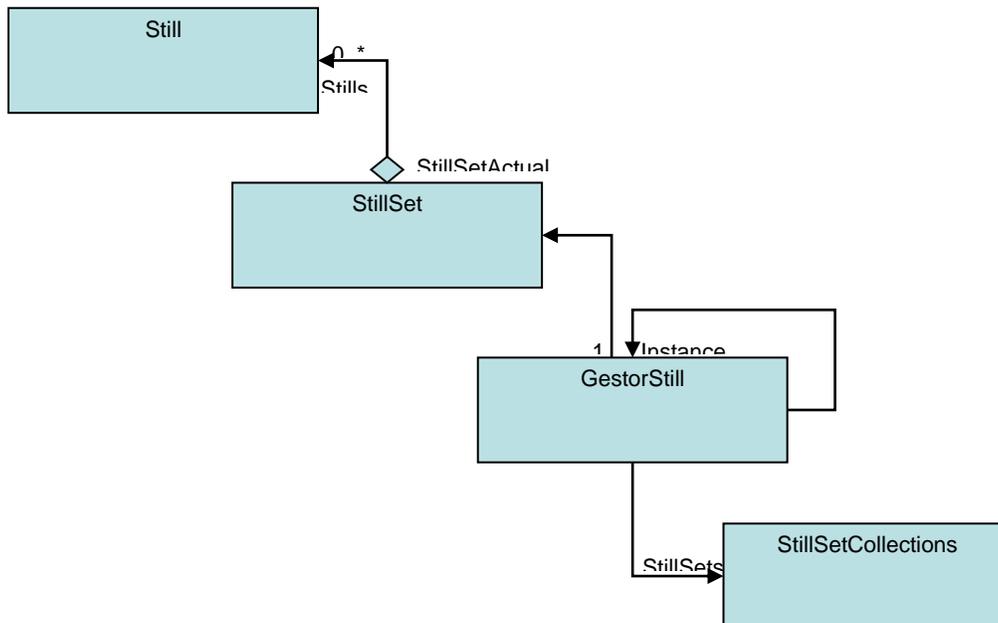


Figura 26. Sección Still

La Figura 26 muestra el fragmento del diagrama de clases que manipula los stills en el juego. Ya se ha mencionado que los Stills son elementos gráficos que cuentan una parte de la historia. Estas clases implementan estructuras de datos y comportamientos para ello.

Still: Es la estructura de datos que representa un still, contiene propiedades para el nombre del audio de fondo, el nombre de la imagen a cargar y el texto que se va a mostrar. También contiene la posición en la pantalla en la que se imprimirá el texto, con el ánimo de hacerlo variable. Contiene un nombre que lo identifica entre los Stills del mismo StillSet.

StillSet: Un StillSet más que un conjunto de Stills representa una secuencia de estos como parte del mismo hilo argumental. Un StillSet, aparte de la colección de Stills, contiene el nombre del archivo de sonido de fondo que se reproducirá durante la muestra del StillSet. También tiene un nombre único que lo identifica entre los StillSets del juego.

StillSetCollection: Implementa List y CollectionBase, se usa como utilidad para manejar colecciones de StillSets.

GestorStills: Es una clase Singleton que administra los Stills, tiene una referencia a un StillSets que son los StillSets que se han cargado a partir de los datos XML en archivos en un XML; GestorStills carga los archivos XML de un directorio dado. También se encarga de mostrar los

Stills en pantalla, esto es, escribir el texto, reproducir el audio de ambientación del StillSet y el audio de la narración. También controla la secuencia de StillSets y dada la orden avanza al Still que siga en el StillSet que se está mostrando.

Sección Guión

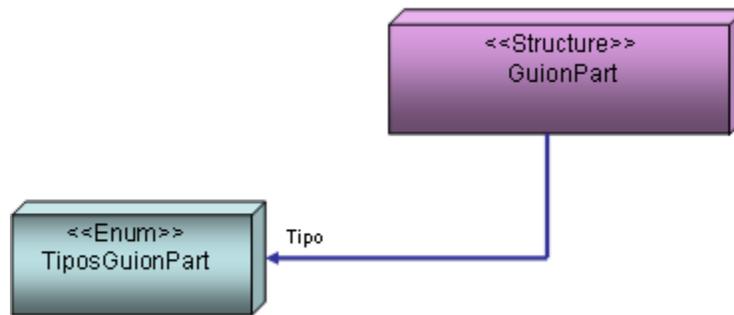


Figura 27. Sección Guión

La Figura 27 muestra el fragmento de una pequeña sección de clases que se encarga de incluir parámetros de Guión al juego, es decir, definir mediante XML la secuencia de mundos y narrativa del juego.

GuiónPart: Esta estructura de datos representa una parte de un guión tiene un tipo y unos argumentos que se interpretan dependiendo del tipo.

- Tipo Reproducir StillSet: Si se da este tipo el Juego reproduce en Pantalla el StillSet con el nombre dado en el campo argumentos
- Tipo Play Mundo: Si se da este tipo el juego carga y plantea el mundo del archivo dado en los argumentos.

El guión es una secuencia de GuiónPART y es manipulado por la clase FrmJuego de PlopOGRé, esta clase reproduce de forma adecuada todas las partes de un guión que carga de un archivo XML.

Capa de Persistencia

La capa de persistencia, como se mostró en la arquitectura es básicamente un paquete con clases para el acceso a datos desde archivos XML. Esta funcionalidad se usa exclusivamente

para acceder a los datos de los mundos, banderas, conversaciones, personaje entre otras formas de dar parámetros al juego y ofrecer extensibilidad.

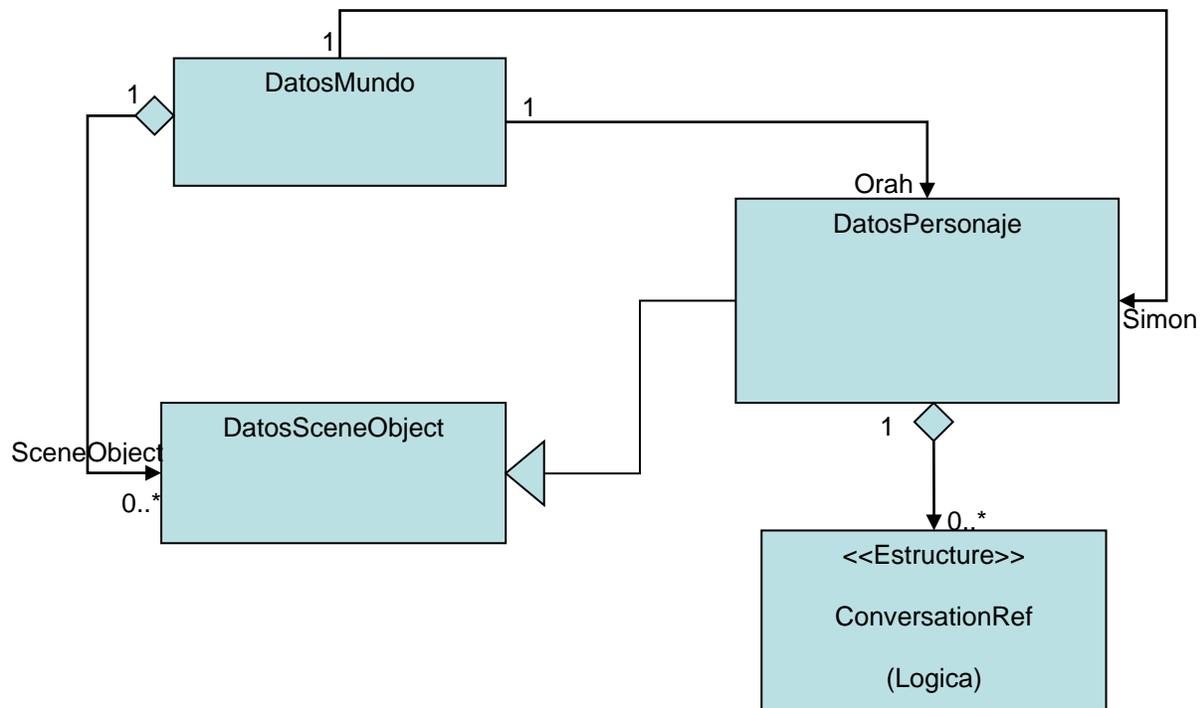


Figura 28. Clases en Persistencia

En el paquete persistencia sólo implementa clases que acceden a datos para aquellos elementos que no son serializables en sí, como el Mundo, los SceneObject y los personajes, otras clases que se cargan de archivos XML como conversación y Condición o Guión ya son serializables por lo que no se requiere su representación en esta capa.

DatosMundo: Esta clase contiene los datos que son necesarios para crear un mundo, los atributos son serializables para obtenerlos a partir de un archivo XML. Un Objeto Datos Mundo se compone, de forma similar a la clase Mundo, de una colección de SceneObjects. Pero a diferencia de la clase Mundo, DatosMundo, debe contener una colección de DatosPersonaje, para los personajes del mundo, no implementa el patrón Polimorfismo debido a que en el archivo XML se deben especificar claramente los tipo de Objetos sean objetos normales de escena o Personajes.

DatosMundo tiene dos referencias especiales a DatosPersonaje una para los datos de Simón y otra para los datos de Oras, estas referencias son tratadas de forma especial al crear los objetos propios de la clase Mundo.

DatosSceneObject: Contiene datos necesarios para crear un SceneObject.

DatosPersonaje: Contiene Datos para crear un PNJ. Un objeto DatosPersonaje también se compone de ConversacionRef, que de por sí es serializable, esta colección sirve para mapear las conversaciones que puede ejecutar un PNJ, si esta colección se especifica para el objeto de Simón no se tiene en cuenta.

4.4 IMPLEMENTACIÓN

El Diseño mostrado es el resultado de todas las iteraciones que se hicieron en la construcción del software, en este proceso se alternaba entre el diseño y la implementación y por supuesto las pruebas correspondientes y la verificación por parte del grupo de ingeniería y el Diseñador Gráfico, así como de los asesores técnicos del proyecto. Debido a que es importante el Diseño como la implementación este capítulo muestra algunos particulares de ésta y la estructura que usa el producto final para funcionar, esto es que requiere el Motor Simón Engine para funcionar en cuanto a sus archivos XML.

Para empezar el diseño fue diagramado en Power Designer 12.1 el cual tiene un gran soporte para visual Studio pues posee Generación automática de código para C# 2.0. Entre las herramientas con que se contaba se tenía Microsoft Visual Studio .Net 2005. Así que la implementación se realizó usando MOGRE sobre Visual Studio. La Solución determinada corre sobre una máquina DELL Dimensión 8300 de propiedad de la Universidad del Cauca con las siguientes Características:

- Procesador Pentium 4 3.00 Ghz.
- Memoria RAM de 1.00 GB.
- Tarjeta Gráfica Nvidía FX 5200 de 128MB.

Luego del Diseño en Power Designer y la generación de código se procede a implementar los métodos de acuerdo a la especificación en el diagrama de Clases.

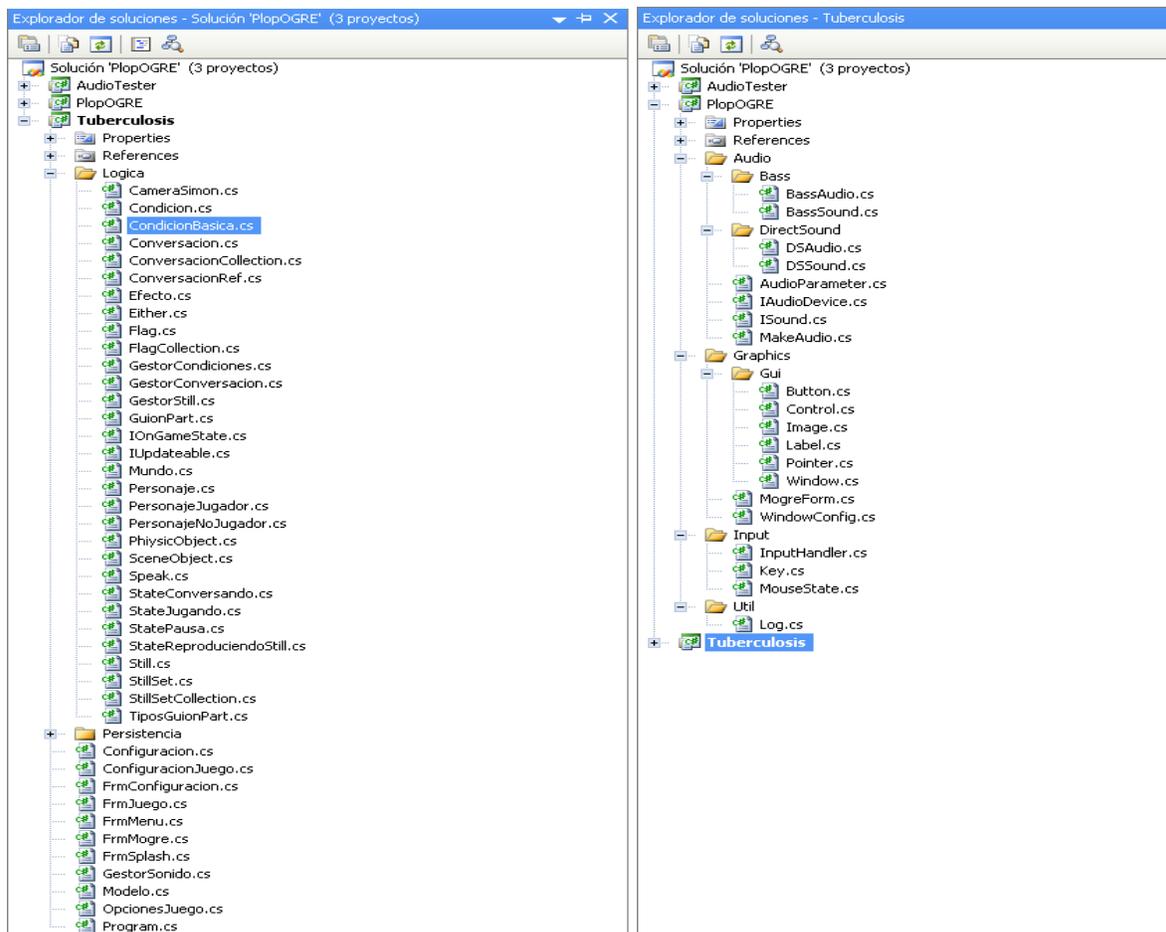


Figura 29. Visión de la Solución en Visual Studio

En la Figura 29 se muestra el explorador de soluciones con los archivos .cs de la Solución implementada para el videojuego. Esta solución se llama PlopOGRE y cuenta con tres proyectos para el AudioTester, las utilidades de PLOPOGRE que es una biblioteca de clases y el proyecto Tuberculosis cuyo ensamblado final es un ejecutable (por ello el archivo program.cs como punto de comienzo) que es el binario principal del motor de videojuego Simón Engine.

4.4.1 El Motor Simón Engine

El núcleo del videojuego como producto final es el software que se ha denominado Simon Engine, esto es por que el Ejecutable que produce la solución en visual Studio no funciona sólo

como videojuego sino como un software que mecaniza los parámetros que se otorguen mediante archivos XML para que se reproduzca el videojuego como tal.

El motor Simon Engine cuenta con una especial Estructura de Directorios que sirve para mantener el orden sistémico del motor, el videojuego asume la existencia de estos directorios y utiliza los archivos XML dentro de ellos para establecer los parámetros del juego.



Figura 30. Visión del directorio de carpetas en Visual Studio

Esta Figura muestra el árbol de directorios que maneja El motor Simon Engine. Se describirá brevemente los más importantes.

CARPETA	DESCRIPCIÓN
FUENTES	Contiene fuentes ttf que el videojuego utiliza, estas fuentes son interpretadas por OGRE.
GUIONES	Este directorio contiene guiones en archivos XML.
MATERIALES	Contiene archivos de imagen jpg, bmp, tga o png para el manejo de las texturas o imágenes en el videojuego. Estas imágenes son para ser usadas por OGRE.
MODELOS	Contiene archivos .mesh y .skeleton que son los archivos con el formato adecuado para que OGRE ponga modelos 3D en pantalla. Estos archivos son creados en una herramienta de modelado 3D y posteriormente exportado a estos formatos. Los archivos .skeleton son necesarios para aquellos modelos que tengan animación.
MUNDOS	Contiene archivos XML que definen los mundos del juego, también contiene archivos para conversaciones y Flags.
OPCIONES	Contiene un archivo XML de configuraciones básicas del juego.
RECURSOS	Este directorio es el que tiene la estructura básica de directorios "media" de OGRE dentro de este directorio y sus hijos están distintos tipos de archivos como fuentes, imágenes, modelos etc. El directorio Recursos/Material/Scripts contiene archivos

	.material necesarios para los modelos 3D que usan texturas y para otras imágenes planas dentro del juego. El directorio Recursos/Overlays contiene archivos .overlay que estructuran los overlays del juego. Un Overlay es una capa 2D que se pinta en la pantalla y que tiene paneles y contenedores que muestran información a modo de texto o imágenes. En este videojuego los Overlays se usan para implementar los Stills.
SONIDOS	Contiene archivos .mp3 y .wav que son los sonidos del juego que reproduce la librería incorporada.
STILLSETS	Contiene archivos XML que definen los StillSets del juego.

Tabla 13. Descripción de las carpetas del motor Simon Engine.

El ejecutable del Simon Engine en su mismo directorio contiene otros ensamblados y archivos de los cuales depende. Estos son: plugins.cfg, recursos.cfg, bass.dll, bass.net.dll, bass_wadsp.dll, MOGRE.dll, MOGRENewt.dll, MOIS.dll, PlopOgre.dll, bguires.zip.

Los archivos XML que se usan para dar parámetros al videojuego deben cumplir con determinados esquemas para las estructuras de datos. A continuación se explica el manejo del XML para el juego y los XSD que los validan.

4.4.1.1 Extensibilidad del videojuego, uso de XML y XSD

Ya que la aplicación en sí sólo es una parte del producto final, necesita de parámetros de videojuegos que son proporcionados al motor mediante archivos XML. Este subcapítulo muestra algunos ejemplos de los archivos usados para el nivel del videojuego y la estructura básica de los XML Schema Definition usados para que los archivos XML sean válidos y de esta forma correctamente interpretados por el Simon Engine. El marcado completo de los XSD al que se refiere este capítulo se puede observar en el ANEXO C.

4.4.1.1.1 Mundo, Conversaciones y Flags

El esquema del mundo es el documento más grande debido a la compleja estructura de datos que administra el mundo y el diseño que se realiza de estos que es bastante grande, además el diseño del mundo también debe contener la definición de las conversaciones y flags de ese determinado escenario. La figura esquemática del XSD mostrada en el ANEXO C muestra el diseño de la estructura del XSD en Power designer 12.1, esto no se explicará a fondo debido a que la estructura es exactamente la misma y las relaciones son muy similares al diseño de clases mostrado más atrás.

Un factor importante para la organización de los archivos es que Simon Engine entiende que los parámetros de XML de los mundos se encuentran, como ya se explicó, en el directorio "Mundos". Y que el nombre del archivo debe tener la extensión .xml, además si el nombre del mundo es X los archivos deberán llamarse:

X.xml = Archivo del mundo

X_Conversaciones.xml = Archivo de las conversaciones del mundo X

X_Flags.xml = Archivo de las flags del mundo X

A continuación un fragmento de un ejemplo usado para el primer nivel del videojuego.

```
<DatosMundo xsi:noNamespaceSchemaLocation="./Mundo.xsd" AudioAmbientacion
="Sonidos/Final Confrontation.mp3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <simon>
    <NombreObjeto>Simon</NombreObjeto>
    <ModeloFile>simoncuerpo.mesh</ModeloFile>
    <Rapidez>20.0</Rapidez>
    <Estado>Idle</Estado>
    <Cara>simon-icono.png</Cara>
    <Posicion>
      <x>1855.868</x>
      <y>0</y>
      <z>13586.59</z>
    </Posicion>
    <Escala>
      <x>1.0</x>
      <y>1.0</y>
      <z>1.0</z>
    </Escala>
    <Direccion>
      <x>1.0</x>
      <y>0.0</y>
      <z>0.0</z>
    </Direccion>
  </simon>
  .....

```

Listado 3. Marcado XML que define un MUNDO

El siguiente es el ejemplo de conversaciones

```
<?xml version="1.0" encoding="utf-8"?>
<Conversaciones xsi:noNamespaceSchemaLocation="./Conversaciones.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Conversacion Id ="Vete">
    <Speak>
      <Speak Tipo ="character" ArchivoSonido="Sonidos/vete.wav">
        <Texto>Vete de aqui, me estorbas!!</Texto>
      </Speak>
    </Speak>
    <Efectos>
      <Efecto Tipo="Activate" Argumentos="MiFlag"/>
    </Efectos>
  </Conversacion>
</Conversaciones>
```

Listado 4. Marcado en XML que define las conversaciones en el mundo

Y El ejemplo de Flags:

```
<?xml version="1.0" encoding="utf-8"?>
<Flags xsi:noNamespaceSchemaLocation="./Flags.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Flag Nombre="MiFlag" Activa="true" />
  <Flag Nombre="Inicio" Activa="true" />
</Flags>
```

Listado 5. Marcado en XML que define los flags

4.4.1.1.2 Guión

El esquema del guión define un Guión como un conjunto de GuionPart, que es un elemento con estructura igual a la de la clase con su mismo nombre. Se asume que la secuencia a seguir de los GuionPart es de arriba abajo, es decir a modo de estructura FIFO donde el primer elemento es el primer TAG en aparecer en el marcado.

El ANEXO C muestra el esquema con que valida los archivos en el directorio Guiones.

El archivo de guión principal, dado que Simon Engine solo usa uno, debe estar en el directorio "Guiones"

A continuación un ejemplo:

```
<Guion xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="./guion.xsd">
  <GuionPart Tipo ="playStillSet" Argumentos ="StillSetInicio"/>
  <GuionPart Tipo ="playMundo" Argumentos
="Mundos/SanJoseDeLasPiedras1.xml"/>
</Guion>
```

Listado 6. Marcado en XML que define la secuencia de escenas en el juego

4.4.1.1.3 StillSets

El archivo de StillSets debe contener el nombre del StillSets y el conjunto de Stills que se reproducirán en orden FIFO similar al Guión. A diferencia de los otros parámetros Simon Engine Carga todos los StillSets del directorio StillSets, a partir de todos los archivos que existan en ese directorio con extensión .xml. De esta forma, se debe cuidar de no poner otro tipo de archivos .xml en este directorio que no cumplan con el esquema StillSets.xsd (Ver ANEXO C).

A continuación el ejemplo de marcado de StillSets:

```
<?xml version="1.0" encoding="utf-8"?>
<StillSet Nombre="StillSetInicio" AudioFondo ="Sonidos/Samples/Jefe_rapido.ogg"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="./StillSet.xsd">
  <Stills>
    <Still Nombre="Still1" Imagen ="IntroStill1.jpg" ArchivoSonido
="Sonidos/StillSet1.wav">
      <Texto>La historia empieza en el viejo pueblode San Jose de Las
Piedras...</Texto>
      <PosTexto>
        <x>
          10
        </x>
        <y>
          10
        </y>
      </PosTexto>
    </Still>
  </Stills>
</StillSet>
```

Listado 7. Marcado en XML que permite configurar uno de los StillSet.

Y Continúa, los ejemplos completos están en el directorio StillSets ver ANEXO C del videojuego.

4.4.2 Test de Preguntas del Juego

Para evaluar a los estudiantes sobre la temática del videojuego, se ve la necesidad de adicionar a éste una funcionalidad que le permita desplegar una serie de preguntas con múltiples respuestas, donde el usuario seleccione la que crea correcta y al terminar de diligenciar el cuestionario, el sistema evalúe la cantidad de aciertos obtenidos durante la prueba.

En la Figura 32. Niños del grupo experimental en interacción con el videojuego. Se indican las clases y las relaciones que hay entre ellas para controlar la funcionalidad del test de preguntas.

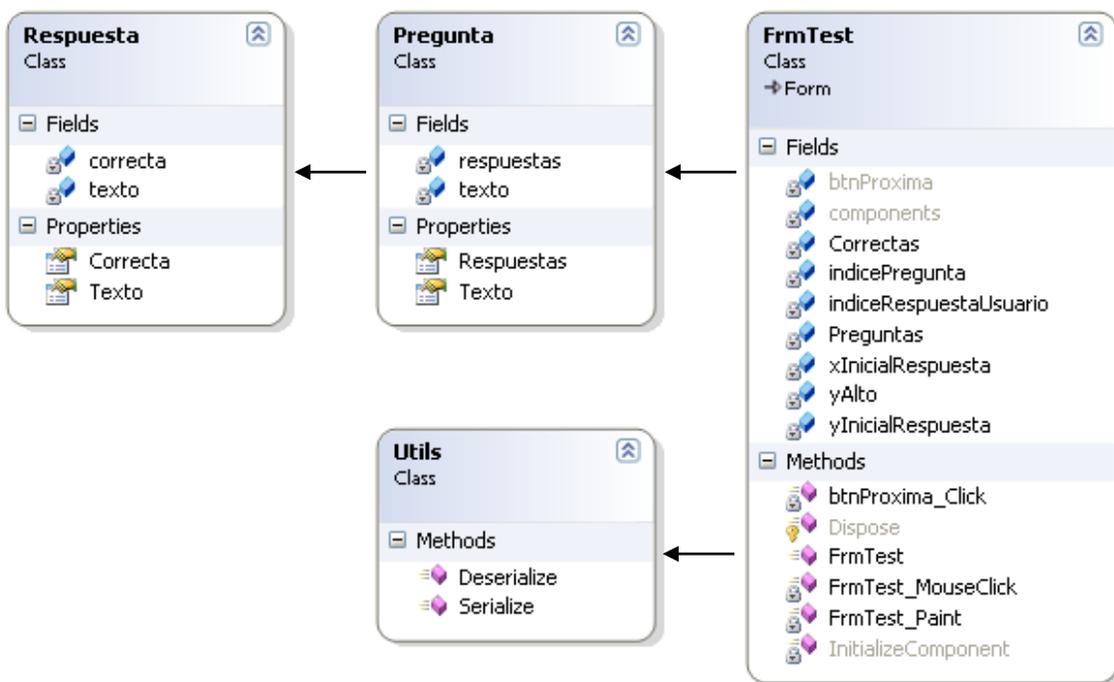


Figura 31. Diagrama de clases test de preguntas del videojuego

El formulario FrmTest es el encargado de desplegar al usuario una pregunta en particular con las posibles respuestas. Estas respuestas se cargan de la clase pregunta, y cuando el usuario selecciona una respuesta será evaluada en la clase respuesta. Luego de terminar el cuestionario, el formulario compara la información adquirida con la que se encuentra disponible en la clase Utils, clase que deserializa y serializa un archivo XML que contiene las respuestas correctas, donde posteriormente se indica al usuario los resultados obtenidos de la prueba.

```
- <Pregunta>
  <Texto>¿ Cual de las siguientes opciones no es un síntoma de la Tuberculosis Pulmonar ?</Texto>
- <Respuestas>
  - <Respuesta>
    <Texto>Tos</Texto>
    <Correcta>false</Correcta>
  </Respuesta>
  - <Respuesta>
    <Texto>Dolor de cabeza</Texto>
    <Correcta>false</Correcta>
  </Respuesta>
  - <Respuesta>
    <Texto>Dolor en el pecho</Texto>
    <Correcta>false</Correcta>
  </Respuesta>
  - <Respuesta>
    <Texto>Sudoración</Texto>
    <Correcta>false</Correcta>
  </Respuesta>
  - <Respuesta>
    <Texto>Vómito</Texto>
    <Correcta>>true</Correcta>
  </Respuesta>
</Respuestas>
</Pregunta>
```

Listado 8. Marcado en XML que permite configurar uno de los StillSet.

En este listado se observa una parte del archivo XML que compara las respuestas digitadas por el usuario con las del sistema. En este caso es la primer pregunta que se despliega, donde se puede notar que tiene cinco posibles respuestas y que la correcta es la última.

4.5 PRUEBAS DE CAJA NEGRA

El objetivo de este tipo de pruebas es encontrar la mayor cantidad de posibles fallas al software, este tipo de pruebas tiene éxito si se encuentran defectos, pero fracasa si estos se presentan y no fueron detectados.

En las pruebas de caja negra existen algunos casos de prueba los cuales pretenden demostrar que las funciones del software son operativas, donde una entrada produce una salida correcta; estas pruebas se aplican sobre la interfaz del software, dejando a un lado el comportamiento interno, el diseño y estructura del programa [48].

En el caso del videojuego se procede a seleccionar unos elementos de prueba, los cuales se despliegan en la interfase durante el momento en que se este ejecutando el videojuego, dichos elementos son:

1. Tiempos de respuesta.
2. Movimiento del personaje jugador (Simón).
3. Movimiento del perro guía (Orah).
4. Movimientos de soldados enemigos.
5. Diálogos entre personajes jugador y no jugador.
6. Cambio de escenas (Stills).
7. Evaluación al jugador (Prueba de conocimiento sobre la tuberculosis pulmonar.).
8. Funcionamiento del menú.

Luego de identificar los elementos de prueba, se procede a seleccionar las clases de equivalencia, estas clases se dividen en dos tipos, las clases de equivalencia válidas y las inválidas. Las válidas son las que permiten definir el comportamiento ideal del videojuego ante cierta condición externa realizada por el jugador, y en las clases de equivalencia inválidas, se definen los casos erróneos.

CONDICIONES EXTERNAS	CLASES DE EQUIVALENCIA VÁLIDAS	CLASES DE EQUIVALENCIA INVÁLIDAS
Cargar interfaz del juego (Elemento de prueba 1)	Tiempo de carga entre 0 y 60 segundos.	Tiempo de carga mayor a 60 segundos.
Movimiento adelante (↑) (Elemento de prueba 2)	El personaje se desplaza 1 o mas posiciones sobre el eje Z positivo.	El personaje se mueve en un eje diferente a Z positivo.
	Se calcula la altura del personaje con respecto al terreno.	El personaje se eleva o atraviesa el terreno.
Movimiento atrás (↓) (Elemento de prueba 2)	El personaje se desplaza 1 o mas posiciones sobre el eje Z negativo.	El personaje se mueve en un eje diferente a Z negativo.
	Se calcula la altura del personaje con respecto al terreno.	El personaje se eleva o atraviesa el terreno.

Movimiento derecha (→) (Elemento de prueba 2)	El personaje se desplaza 1 o mas posiciones sobre el eje X positivo.	El personaje se mueve en un eje diferente a X positivo.
	Se calcula la altura del personaje con respecto al terreno.	El personaje se eleva o atraviesa el terreno.
Movimiento izquierda (←) (Elemento de prueba 2)	El personaje se desplaza 1 o mas posiciones sobre el eje X negativo.	El personaje se mueve en un eje diferente a X negativo.
	Se calcula la altura del personaje con respecto al terreno.	El personaje se eleva o atraviesa el terreno.
Movimiento del jugador en cualquier dirección. (Elemento de prueba 3)	El perro guía sigue al personaje a una distancia menor a 100 pixeles.	El perro guía se aleja de Simón o encuentra un obstáculo que impide seguirlo.
	Se calcula la altura del perro con respecto al terreno.	El perro guía se eleva o atraviesa el terreno.
Simón entra al laberinto de la prueba. (Elemento de prueba 4)	Los soldados realizan un movimiento repetitivo que impiden el paso a Simón.	Los soldados se quedan estáticos.
		Los soldados realizan un movimiento que permiten pasar fácilmente a Simón.
		Los soldados atraviesan el terreno y los objetos cercanos a ellos.
El personaje se acerca a un personaje no jugador y presiona la tecla (H) (Elemento de prueba 5)	Se despliega el diálogo apropiado definido en el guión en la parte inferior de la pantalla.	No se despliega el diálogo.
		Despliega un diálogo inválido.
		Despliega el diálogo en otro lado de la pantalla.
		Realiza otra acción diferente a la del diálogo.
	Se despliegan correctamente los Stills definidos para la introducción del videojuego.	No se muestran los Stills definidos para la introducción del videojuego.

Still inicio del juego (Elemento de prueba 6)		Se despliegan con texto o audio incompletos.
		Se ejecutan menos de los Stills definidos para la introducción.
Still fin del juego (Elemento de prueba 6)	Se despliegan correctamente los Stills definidos para la introducción del videojuego.	No se muestran los Stills definidos para el fin del videojuego.
		Se despliegan con texto o audio incompletos.
		Se ejecutan menos de los Stills definidos para el fin del juego.
No se selecciona una respuesta y se selecciona siguiente. (Elemento de prueba 7)	Desplegar un mensaje que le indique al jugador que debe seleccionar una respuesta.	El sistema marca la pregunta como válida.
		El sistema marca la pregunta como inválida
Se selecciona la respuesta correcta. (Elemento de prueba 7)	Guardar adecuadamente el campo de respuesta correcta.	No guarde las respuestas correctas en el campo correspondiente.
	Permitir seguir con la evaluación.	Al seleccionar una respuesta no permita continuar con la evaluación.
Finalizar la prueba (Elemento de prueba 7)	Desplegar el total de aciertos al finalizar la prueba.	No desplegar el mensaje de total de aciertos.
	Desplegar el still de fin del juego	No cargue el still de fin de juego, cargue un still erróneo o una escena del videojuego.
Presionar tecla (▼) en el Menú. (Elemento de prueba 8)	Se cambia a la siguiente opción.	No cambia de opción.
	Si es la ultima opción, vuelve a la primera.	Cambia a una opción errónea.
Presionar tecla (▲) en el Menú.	Se cambia a la siguiente opción.	No cambia de opción.

(Elemento de prueba 8)	Si es la primera opción, salta a la última opción.	Cambia a una opción errónea.
Presionar la tecla (Enter) o hacer doble click sobre una opción del menú. (Elemento de prueba 8)	Cargar la funcionalidad correcta que esta relacionada a la opción.	Cargar una funcionalidad totalmente diferente a la indicada.
		No cargar funcionalidad.

Tabla 14. Clases de equivalencia válida e inválida

Ya se han identificado los elementos de prueba al igual que las clases de equivalencias válidas e inválidas, ahora se procede a identificar los casos de prueba para obtener y evaluar los resultados obtenidos al ejecutar cada una de las condiciones externas de la tabla anterior.

NOMBRE CASO	RUTA O ACCION	RESULTADO
Cargar interfaz del juego.	El jugador presiona “Enter” o hace “Doble-Click” sobre el ejecutable del videojuego.	Cuando la máquina no esta ejecutando otros programas, la interfaz del juego demora en cargar 20 segundos aproximadamente. Pero si la máquina esta realizando otros procesos, este tiempo se puede estar duplicando. Pero en ningún caso logra superar el minuto.
Movimiento del personaje jugador (Simón).	El jugador ya ha cargado el videojuego y desea desplazar a Simón presionando cualquier tecla de movimiento, ya sea hacia adelante, atrás, izquierda o derecha.	Al presionar cualquier tecla de movimiento, Simón se desplaza en el eje correcto, ya sea en los ejes Z, X positivos o negativos. Se calcula adecuadamente el manejo de alturas de Simón con respecto al terreno.
Movimiento del perro guía (Orah).	El jugador dirige a Simón hacia cualquier dirección.	Orah siempre los sigue a una distancia no menor de 100 pixeles.

		<p>No se calcula bien el manejo de alturas con respecto a Orah.</p> <p>El movimiento de Orah no es el esperado.</p>
Movimientos de soldados enemigos.	El jugador entra a realizar la prueba de cruzar el laberinto.	<p>Los soldados se despliegan correctamente por los espacios del laberinto.</p> <p>Los soldados no cruzan los objetos, es decir se mantiene el manejo de colisiones perfectamente.</p> <p>El movimiento de los soldados es algo lento, permitiéndole al jugador evadirlos con un nivel de dificultad fácil.</p>
Diálogos entre PJ y PNJs.	El jugador presiona la tecla (H) al estar en frente de un personaje no jugador.	Se despliega el diálogo correcto, con el audio adecuado y en la ubicación seleccionada (Parte inferior de la pantalla).
Cambio de escenas (Stills).	Cuando el jugador inicia y finaliza el juego, se deben cargar una serie de still que brindan cierta información sobre el videojuego.	<p>Se inicializa el juego y se verifica que cargue los tres still que despliegan la historia del videojuego en su totalidad.</p> <p>Se finaliza el videojuego y se carga el still correcto que despliega la información en su totalidad sobre la culminación de este.</p>
Evaluación al jugador. (Prueba de conocimiento sobre la tuberculosis)	El usuario pasa la prueba del laberinto y obtiene la	Se despliega el cuestionario. Cuando no se selecciona una respuesta, el sistema no permite

<p>pulmonar.)</p>	<p>información de la gema. El sistema debe desplegar el cuestionario relacionado a la tuberculosis pulmonar y finalmente mostrar el resultado obtenido.</p>	<p>pasar a la siguiente pregunta y despliega un mensaje informativo al usuario. Se verifica que guarde correctamente la opción digitada por el usuario. Carga en el orden correcto la serie de preguntas previamente definidas. Al finalizar la prueba, el sistema despliega la puntuación obtenida.</p>
<p>Funcionamiento del menú.</p>	<p>Al estar en el menú el usuario puede seleccionar una opción por medio de las flechas y ejecutarla.</p>	<p>Al iniciar el videojuego se visualiza el menú. Al presionar flecha (↓) y si existen mas opciones abajo, se cambia de opción en ese sentido, si es la última opción, pasa a la primera opción. Al presionar arriba (↑) y si existen mas opciones arriba, se cambia de opción en ese sentido, si es la primera opción, pasa a la última opción. Una vez decidida la opción a ejecutar se presiona Enter y se carga la interfaz que permite la funcionalidad correcta de esa opción.</p>

Tabla 15. Evaluación de los Casos de Pruebas

La evaluación de los casos de prueba presentados en la Tabla 15 demuestra que el videojuego es un sistema que realiza de manera adecuada las funciones definidas en el guión del videojuego. Solo el caso que evalúa el movimiento del perro guía (Orah) muestra una falla en el cargue del personaje y en el manejo de alturas al desplazarse por los terrenos definidos en el mismo. Es claro tener en cuenta que en el mundo del entretenimiento y en este caso en particular como son los videojuegos, las verdaderas pruebas son realizadas por usuarios finales que son un grupo de personas privilegiadas que tienen la oportunidad de probar el videojuego realizando su rol de jugadores y quienes finalmente, pueden o no descubrir fallas en las versiones finales listas para salir al mercado.

5 PRUEBA DE CAMPO

5.1 DESCRIPCIÓN GENERAL

En este capítulo se describirá la experiencia de campo realizada, inicialmente se presentará el marco de referencia definiendo la muestra objetivo del experimento y el espacio utilizado para el desarrollo del mismo; después se hará una descripción del diseño del experimento donde se definirá claramente lo que requiere, también se describirá la ejecución de la práctica y los resultados obtenidos, así como el análisis de los mismos.

5.1.1 Marco de Referencia

Para la validación del proyecto se han realizado dos pruebas en la Institución Educativa el Colegio San Francisco de Asís, ubicada en el municipio de Popayán, departamento del Cauca, Colombia; la primera fue una charla de tipo tradicional y la segunda fue la interacción con el videojuego.

La experiencia ha involucrado 12 estudiantes de grado sexto de básica primaria con un promedio de edad de 11 años y 1 docente de Biología de la Universidad del Cauca quien es actualmente el coordinador académico del colegio mencionado.

Algunas de las actividades desarrolladas con el fin de dar inicio a la investigación fueron:

Primero, se optó por buscar un colegio privado pensando en la posibilidad de encontrar mejores herramientas tecnológicas y espacios adecuados que facilitarían la realización de la prueba, esto con el fin de explorar el entorno donde se realizaría la experiencia de campo.

Segundo, se seleccionó la muestra objetivo del experimento, conformada por 12 estudiantes del grado sexto (6) B del Colegio San Francisco de Asís y con edades entre los 10 y los 14 años, los estudiantes fueron seleccionados al azar sin tener en cuenta ningún tipo de cualidades o características en ellos pues fueron seleccionados sin conocerlos previamente.

Tercero, de la muestra objetivo fueron seleccionados 6 niños para conformar el grupo de control y otros seis conformaron el grupo experimental.

Cuarto, se identificó que el coordinador académico era la persona adecuada para realizar la charla magistral por sus conocimientos en Biología y su experiencia en el tema.

5.1.2 Diseño del experimento

Se advierte que lo primero que se realizó en esta prueba de campo fue una exploración del conocimiento de los niños sobre la temática tratada en el videojuego, es decir, sobre los síntomas y factores de riesgo de la tuberculosis pulmonar, el docente al iniciar la charla lo primero que les preguntó a los niños fue si tenían algún conocimiento sobre la enfermedad tuberculosis pulmonar a lo cual los niños respondieron tímidamente no conocer sobre este tema, solo uno de ellos se atrevió a decir que era una enfermedad relacionada con los pulmones, teniendo en cuenta esta valoración inicial se continuó con el diseño del experimento.

La definición del experimento para el presente proyecto consistió en la realización de una sesión de trabajo práctico con los estudiantes de básica primaria de la institución Colegio San Francisco de Asís y una sesión teórica con estudiantes de la misma institución y el mismo promedio de edades, con el fin de establecer elementos importantes que permitan definir si los niños entre los 10 y 14 años pueden aprender mejor, con el uso de videojuegos educativos, que con otros métodos tradicionales como por ejemplo, la asistencia a charlas magistrales de un tema en particular. La Figura 30 muestra una foto de los niños que se encuentran realizando la prueba práctica con el videojuego Simon y la tuberculosis.



Figura 32. Niños del grupo experimental en interacción con el videojuego.

La Figura 31 muestra una foto de los niños que se encuentran realizando la prueba tradicional definida en una charla magistral.



Figura 33. Niños del grupo de control en charla magistral.

5.1.2.1 Definición del Experimento

La práctica no pretende ser un diseño experimental, pues la muestra objetivo es relativamente pequeña y no se tomó una cantidad numerosa de datos que permitan hacer un análisis más efectivo que tal vez arroje mejores resultados pero se puede tener en cuenta para un trabajo futuro.

Lo que realmente se quiso demostrar con esta experiencia de campo es determinar si la utilización del videojuego y del audio facilita el aprendizaje de la temática referente a la tuberculosis pulmonar para estudiantes entre 10 y 14 años de un colegio privado de la ciudad de Popayán como el San Francisco de Asís.

La práctica se llevó a cabo con 12 estudiantes de básica primaria, de la Institución Educativa Colegio San Francisco de Asís de la ciudad de Popayán, los cuales fueron divididos en dos grupos de 6 niños, seleccionados sin tener en cuenta ningún tipo de característica en especial ya que fueron seleccionados al azar.

Para la realización de esta práctica se selecciono el tema “Los síntomas y factores de riesgo de la tuberculosis pulmonar”, ya que es el tema que se plantea en el videojuego de aventura con el cual se pretende facilitar el aprendizaje de este tema en particular, para los niños del grado sexto B.

Inicialmente se realizó una verificación del entorno y de la infraestructura donde los estudiantes llevaron a cabo la práctica, buscando la homogeneidad de las condiciones ambientales en tiempo y espacio para obtener resultados válidos.

Los dos grupos participantes realizaron sus pruebas definidas como se mencionó anteriormente es decir el grupo de control recibió la charla tradicional y el grupo experimental interactúo con el videojuego, y se observó en ellos su comportamiento frente a cada una de las pruebas identificando el interés puesto en las mismas. Finalmente se les realizó a todo el grupo o muestra objetivo, un cuestionario conformado por cinco preguntas, las mismas para ambos casos y así poder realizar la comparación entre los resultados, de esta manera determina si verdaderamente el videojuego Simon y la tuberculosis facilita el aprendizaje sobre los síntomas y factores de riesgo de la Tuberculosis Pulmonar.

Como se mencionó anteriormente, los niños no tenían conocimiento de la tuberculosis pulmonar así que se iniciará con las pruebas (charla magistral e interacción con el videojuego) las cuales les brindará la información necesaria para resolver el cuestionario. Los resultados de este cuestionario permitirán realizar una observación O1 y O2 a los grupos de control (RG1) y al grupo experimental (RG2). El grupo experimental será el que se ponga a prueba mediante la interacción con el videojuego y el grupo de control serán los que recibirán la charla tradicional. Finalmente se realizó una Posprueba a los participantes de ambos grupos obteniendo la observación O1 y O2 para el grupo de control y grupo experimental respectivamente, de esta manera se podrá analizar los resultados obtenidos de las observaciones y establecer si existe o no, una mejora en el aprendizaje del tema teniendo en cuenta que no era conocido por la muestra del experimento. A continuación se muestra la representación de la prueba:

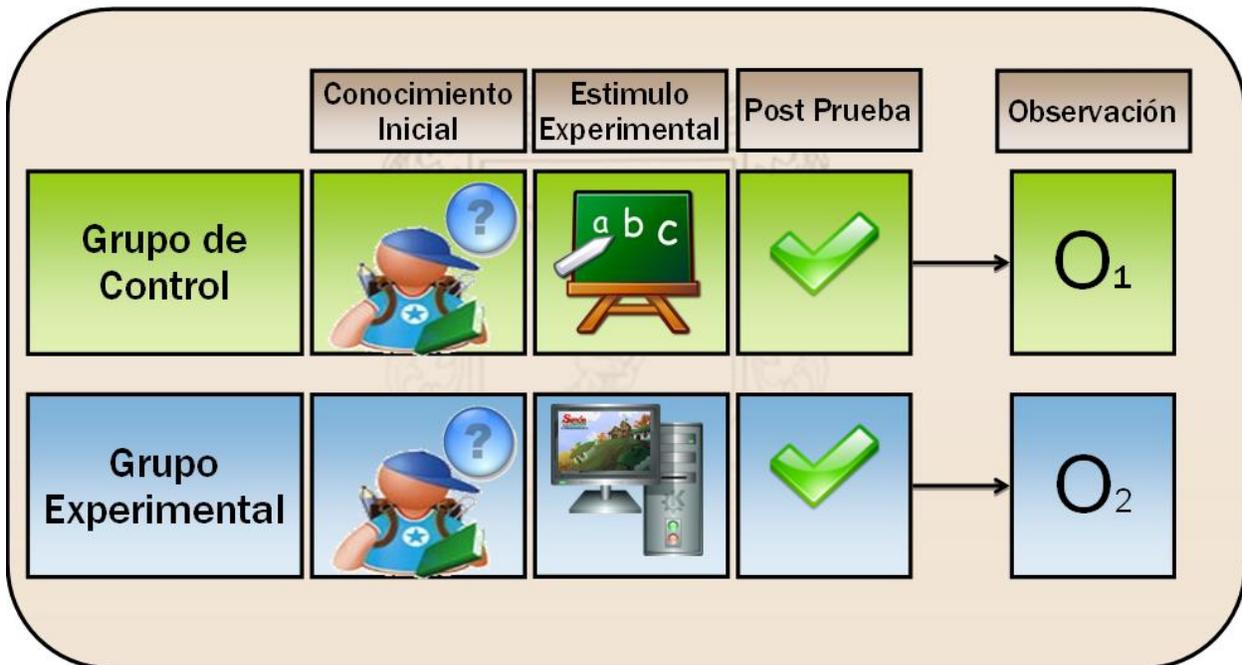


Figura 34. Diseño de la prueba.

5.1.2.2 Ejecución de la práctica

Fue necesario poner al tanto del experimento de campo al docente encargado de dar la charla quien además fue el gestor que consiguió el permiso, las aulas, el tiempo y la disposición de los alumnos esto se hizo con el fin de lograr las mejores condiciones para la ejecución de la práctica

Para la ejecución de la prueba se realizaron las actividades en el siguiente orden:

- 1. Capacitación del docente:** Aunque el docente que dictó la charla tenía gran conocimiento sobre los síntomas y factores de riesgo de la Tuberculosis Pulmonar, se le brindó material para soportar su conocimiento y así los niños del grupo de control obtuvieran la misma información que la presentada en el videojuego Simon y la Tuberculosis la cual sería presentada para el grupo experimental.
- 2. Equivalencia de los grupos en las pruebas:** Se garantizó que los niños de ambos grupos tanto el de control como el experimental se encontraran con las mismas condiciones y comodidades de espacio y tiempo. Ver figura 34.



Figura 35. Niños en igualdad de condiciones para la charla magistral.

- 3. Ejecución de la práctica grupo de control:** Para la realización de la práctica en el grupo de control fue necesario establecer de manera precisa los tiempos requeridos para cada actividad, a continuación se presenta una tabla resumen. Ver Tabla 16.

ACTIVIDAD	TIEMPO
Charla magistral sobre síntomas y factores de riesgo de la Tuberculosis Pulmonar	40 minutos
Aplicación del cuestionario	10 minutos

Tabla 16. Tiempos estimados para la realización de la prueba teórica

4. **Ejecución de la práctica grupo experimental:** Para la realización de la práctica en el grupo experimental, al igual que en el grupo de control, fue necesario establecer de manera precisa los tiempos requeridos para la prueba, a continuación se presenta una tabla resumen. Ver Tabla 17.

ACTIVIDAD	TIEMPO
Interacción con el videojuego Simon y la tuberculosis	40 minutos
Aplicación del cuestionario	10 minutos

Tabla 17. Tiempos estimados para la realización de la prueba práctica

5. **Presentación de pruebas:** los grupos de control y el experimental presentaron un examen con un contenido idéntico sobre el contenido temático presentado solo la presentación de ambos cuestionarios era diferente. Ver Figuras 35 y 36



Figura 36. Presentación del cuestionario presentado al grupo experimental. Ver ANEXO D



Figura 37. Presentación del cuestionario presentado al grupo de control. Ver ANEXO D.

6. Análisis de resultados: Finalmente se procedió a realizar el análisis de los resultados para la presentación del informe de los mismos.

5.1.2.3 Resultados

Una vez definidos el experimento y ejecutada la práctica procedemos a mostrar los resultados.

Se contó con un total de 12 estudiantes para el desarrollo de la actividad los cuales se dividieron en dos grupos diferentes, el grupo que interactuó con el videojuego Simon y la tuberculosis se le denominó grupo experimental y el grupo que recibió la charla se le llamó grupo de control.

Iniciando la actividad se realizó un perfil del estudiante donde se consignan el nombre, la edad y el curso en que se encuentran actualmente.

Con estos datos se procedió a realizar las prácticas tal como se había planeado inicialmente, el grupo de control recibió una charla donde se trataron temas como los síntomas y factores de riesgo de la tuberculosis pulmonar y al grupo experimental se le puso a jugar el videojuego Simon y la tuberculosis con audio.

Posteriormente se realizó la actividad de evaluación de la práctica donde se les entregó a los alumnos un cuestionario con las siguientes preguntas relacionadas con los síntomas y factores de riesgo de la Tuberculosis Pulmonar.

1. ¿Cual de las siguientes opciones NO es un Factor de Riesgo de la Tuberculosis Pulmonar?
 - Desnutrición
 - Alcoholismo
 - Indigencia
 - Drogadicción
 - Enfermedad del corazón

2. ¿Cual de las siguientes opciones NO es un Síntoma de la Tuberculosis Pulmonar?
 - Tos
 - Dolor de cabeza

- Dolor en el pecho
- Sudoración
- Vómito

3. ¿Una persona puede contagiarse de Tuberculosis Pulmonar cuando?

- Al compartir utensilios de personas contagiadas de Tuberculosis
- Al inhalar partículas expulsadas por un enfermo de Tuberculosis al hablar, toser o estornudar
- Al contacto con la ropa de un enfermo con tuberculosis
- Nunca se puede contagiar
- Se usa tapabocas

4. ¿La Tuberculosis Pulmonar puede ser transmitida por vía aérea?

- Si
- No

5. ¿De las siguientes opciones cual de ellas es el Síntoma principal de la Tuberculosis Pulmonar?

- Tos que persiste por más de 15 días
- Fiebre alta
- Pérdida del apetito
- Diarrea
- Dolor en el pecho

Los resultados están en las tablas 16 y 17

NOMBRE DEL ESTUDIANTE	EDAD	CURSO	PREG. 1	PREG. 2	PREG. 3	PREG. 4	PREG. 5	RESULTADOS
Guillermo José Q	11	6B	5	0	5	5	5	4
Luis Felipe Gonzáles	11	6B	5	0	5	5	5	4
Diego Hernán Farfan	10	6B	5	5	5	5	5	5
Brayan Astudillo	12	6B	5	5	5	5	5	5
David Solano	11	6B	5	0	5	5	5	4
Carlos Guillermo Zúñiga	11	6B	5	0	5	0	5	3

Tabla 18. Perfil grupo de control

Edad Promedio: 11 años

NOMBRE DEL ESTUDIANTE	EDAD	CURSO	PREG.	PREG.	PREG.	PREG.	PREG.	RESULTADOS
			1	2	3	4	5	
Fabio Alejandro	11	6B	5	0	0	5	0	2
Delmer Estiven Ortega	10	6B	5	0	5	5	5	4
Juan Manuel Lozano	11	6B	5	0	5	5	5	4
Cristian Camilo Castillo	11	6B	0	5	5	5	5	4
Sebastian Manzano Muñoz	13	6B	5	0	5	5	0	3
Miguel Angel	11	6B	5	5	5	5	0	4

Tabla 19. Perfil grupo experimental

Edad Promedio: 11 años

5.1.2.4 Análisis de los resultados

Lo que se realizó en esta instancia del proyecto fue una comparación de los datos de los resultados del grupo de control con los datos obtenidos con el grupo experimental, se notó que la diferencia se presentó aunque en una pequeña medida en algunos casos favoreciendo el grupo de control y en otros casos favoreciendo el grupo experimental. A continuación veremos los gráficos correspondientes a esas comparaciones discriminados por la pregunta.



Figura 38. Medición del conocimiento.

Pregunta 1

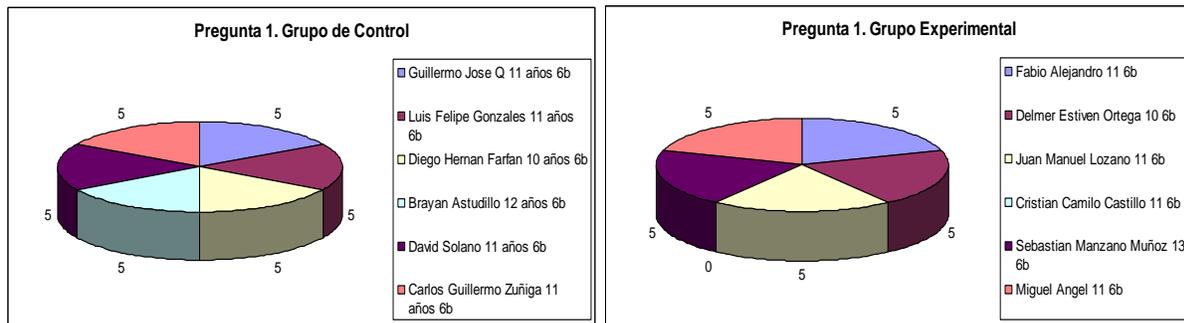


Figura 39. Comparación de los datos obtenidos en la pregunta 1 por el grupo de control y el grupo experimental

En esta primera pregunta el grupo de control mostró una ligera ventaja sobre el grupo experimental el valor promedio de esta pregunta fue:

Grupo de Control: 5

Grupo Experimental: 4.16

Pregunta 2

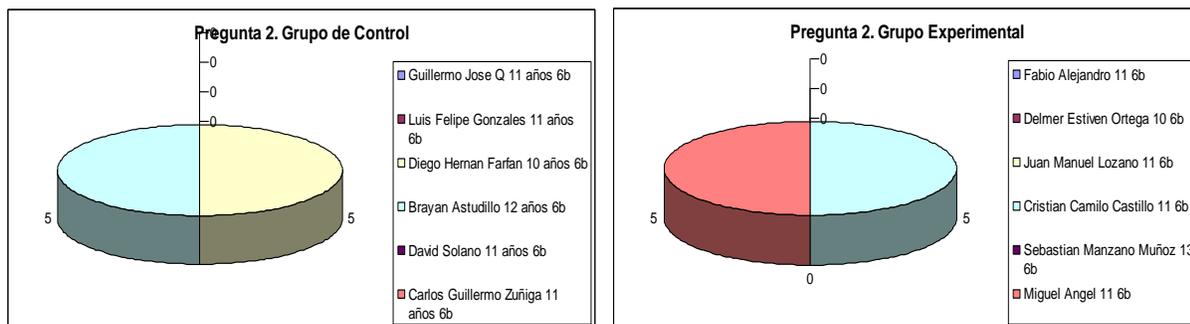


Figura 40. Comparación de los datos obtenidos en la pregunta 2 por el grupo de control y el grupo experimental

En esta segunda pregunta el grupo de control y el grupo experimental obtuvieron el mismo valor promedio el cual fue:

Grupo de Control: 1.66

Grupo Experimental: 1.66

Pregunta 3

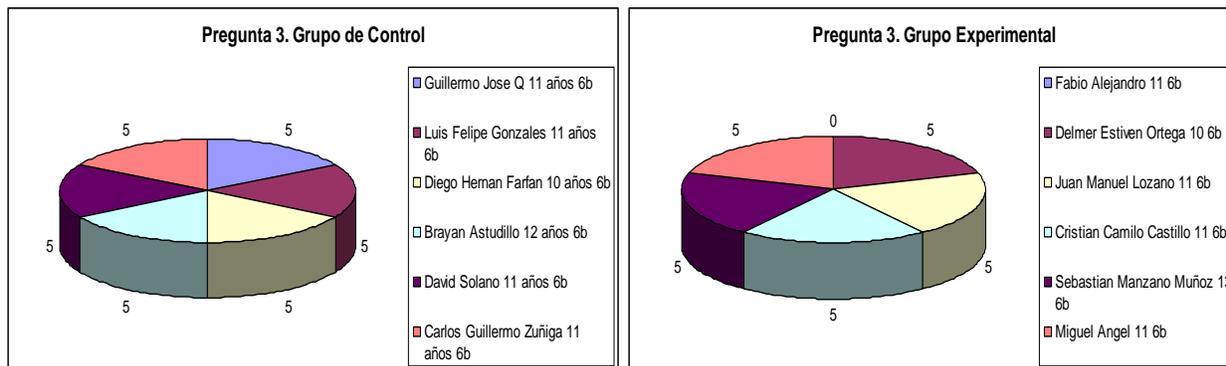


Figura 41. Comparación de los datos obtenidos en la pregunta 3 por el grupo de control y el grupo experimental

En esta tercera pregunta el grupo de control mostró una ligera ventaja sobre el grupo experimental al igual que en la pregunta 1. El valor promedio de esta pregunta fue:

Grupo de Control: 5

Grupo Experimental: 4.16

Pregunta 4

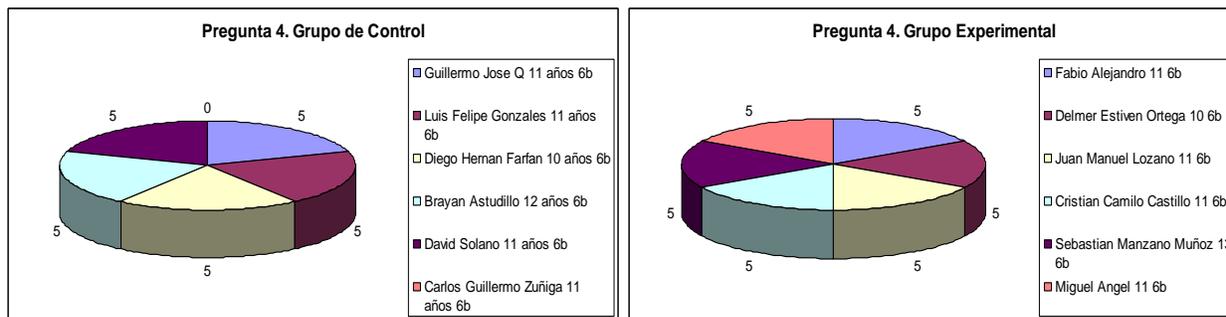


Figura 42. Comparación de los datos obtenidos en la pregunta 4 por el grupo de control y el grupo experimental

En esta cuarta pregunta fue la única donde el grupo que mostró la ventaja fue el grupo experimental. El valor promedio de esta pregunta fue:

Grupo de Control: 4.16

Grupo Experimental: 5

Pregunta 5

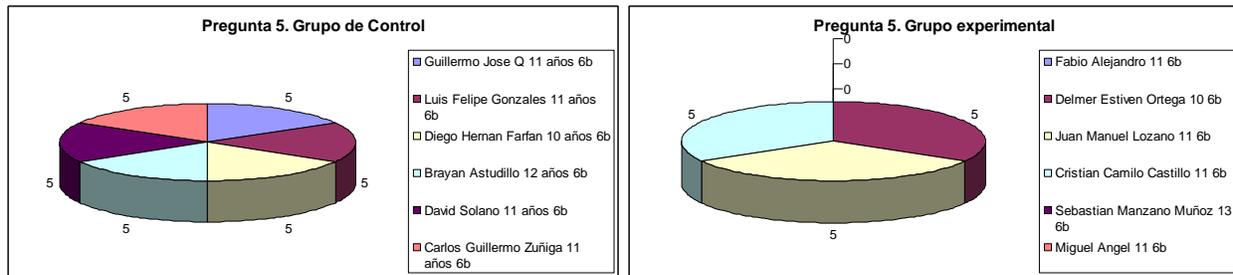


Figura 43. Comparación de los datos obtenidos en la pregunta 5 por el grupo de control y el grupo experimental

En esta última pregunta fue donde realmente se notó una diferencia alta en los datos y de nuevo el grupo de control demostró mejores resultados. El valor promedio de esta pregunta fue:

Grupo de Control: 5

Grupo Experimental: 2.5

Se puede concluir que los métodos tradicionales siguen siendo más útiles que los métodos modernos para enseñar o educar sobre un tema específico como los síntomas y factores de riesgo de la tuberculosis pulmonar. Sin embargo una alternativa diferente como el uso de videojuegos, demuestra que permite al alumno adquirir la información necesaria relacionada a la temática en la que se enfoca esta herramienta. Esto se puede verificar al observar los resultados de la prueba por parte de los estudiantes que interactuaron con el videojuego.

Con los resultados obtenidos se logró evidenciar que la utilización del videojuego en condiciones similares a las del experimento, facilita el aprendizaje acerca de los síntomas y factores de riesgo de la tuberculosis pulmonar, lo cual puede ser un punto de partida para futuros desarrollos en diferentes temáticas y para otras poblaciones objetivo con las mismas condiciones que se presentaron en este desarrollo.

6 CUMPLIMIENTO DE LOS OBJETIVOS

Con el fin de expresar los resultados finales de cada uno de los objetivos, se presenta a continuación una explicación sencilla de cada uno de ellos a medida que se fueron realizando a través de la implementación de la aplicación.

No. De Objetivo: 1 Objetivo General	
Descripción del Objetivo:	Proponer un modelo que incorpore audio al motor OGRE y desarrollar un caso de estudio, a través de un videojuego, que permita a niños entre 10 y 14 años de edad, aprender acerca de los síntomas y factores de riesgo de la tuberculosis pulmonar.
Productos Esperados:	<ol style="list-style-type: none"> 1. Modelo de incorporación de audio al motor gráfico. 2. Desarrollo de una prueba de campo 3. Videojuego
Resultados Obtenidos:	<ol style="list-style-type: none"> 1. Documento relacionado al modelo de inclusión de audio al motor gráfico MOGRE. 2. Desarrollo y documentación de una prueba de campo realizada en el colegio San Francisco de Asís con niños cuyas edades son mayores a los diez años y menores a los catorce. 3. Documentación e implementación de un videojuego educativo con imágenes en 3D, enfocado al aprendizaje de los síntomas y factores de riesgo de la tuberculosis pulmonar. 4. Realización del artículo "MODELO DE AUDIO PARA INCORPORACION A MOTOR GRÁFICO DE JUEGOS" 5. Documento relacionado al tema de los síntomas y factores de riesgo de la tuberculosis pulmonar.
Medios de Verificación:	<ol style="list-style-type: none"> 1. En el Capítulo 3 de este documento se encuentra la Metodología propuesta para incorporar audio al motor gráfico. 2. En el Capítulo 5 de este documento se encuentran la Metodología que se utilizo para realizar la prueba de campo. 3. El capítulo 4 de este documento, se puede encontrar todo lo relacionado al respecto. 4. en la dirección www.enlaceinformatico.unicauca.edu.co se puede verificar la existencia de este artículo.
Estrategias y/o Observaciones:	<ol style="list-style-type: none"> 1. Modelo de incorporación de audio al motor gráfico: Hoy en día los desarrolladores de videojuegos necesitan de herramientas que ofrezcan una gran multifuncionalidad. El modelo creado permite al desarrollador adicionar la librería de audio deseada y reproducir los formatos de audio que esta librería ofrece. Esto sin alterar la funcionalidad de las demás librerías que han sido previamente adicionadas.

	<p>El modelo inicialmente se realizó sólo con una librería (BASS), al terminar las pruebas de audio con esta librería, se procedió a adicionar una segunda (DirectSound), demostrando así que el modelo funciona y que se le pueden seguir adicionando muchas librerías más.</p> <ol style="list-style-type: none">2. Prueba de campo: Esta prueba se elabora gracias a la participación de un grupo de niños, donde un docente les transmitirá sus conocimientos sobre el tema de los síntomas y factores de riesgo de la TB mientras que a otro grupo de niños se les brindara la misma información pero a través de la interactividad con el videojuego. Es importante realizar una previa indagación sobre cual es el conocimiento de la temática por parte de los grupos para poder obtener mejores resultados al culminar la prueba.3. Videojuego: Es muy importante al pensar en desarrollar un videojuego tener en cuenta el trabajo interdisciplinario, esto se debe a que el papel que desempeñó el diseñador gráfico en este proyecto fue de vital ayuda. Además hay que saber elegir las herramientas apropiadas para implementarlo, ya que esto hace que al momento de presentarse un problema, se cuente con una basta información (como por ejemplo en la web) sobre como solucionarlos. En Popayán aunque poco a poco va creciendo el grupo de personas que estén relacionadas con el área de desarrollo de videojuegos, aun este grupo es algo limitado, razón por la cual hace que se complique el poder tener acceso a información y a soluciones rápidas y apropiadas. El video juego se realiza en base a un guión y con unos modelos en 3D entregado por el diseñador gráfico. Todo esto utilizando el motor MOGRE al cual se le aplicó el modelo de integración de audio, obteniendo así el producto final.
--	---

No. De Objetivo: 1 Objetivo específico	
Descripción del Objetivo:	Crear e implementar un modelo que permita la incorporación de audio a OGRE, de forma tal que pueda ser utilizado en futuros trabajos para el desarrollo de nuevos videojuegos y aplicaciones multimedia en 3D.
Productos Esperados:	<ol style="list-style-type: none"> 1. Arquitectura del Modelo de Integración de audio
Resultados Obtenidos:	<ol style="list-style-type: none"> 1. Arquitectura del Modelo de Integración de audio 2. Herramienta reproductora de audio. 3. Modelado XSD. 4. Modelado XML.
Medios de Verificación:	<ol style="list-style-type: none"> 1. En el Capítulo 3.1 ARQUITECTURA DEL MODELO DE AUDIO MOGRE se visualiza el modelo actual y el modelo cuando se le adicionen más librerías. 2. En el Capítulo 3.4 IMPLEMENTACION se la funcionalidad de dicha herramienta y su interface. 3. En los anexos C se puede consultar sobre la funcionalidad de este tipo de modelado en el videojuego. 4. En el capítulo 4.4.1.1 de este documento, se puede constatar sobre el uso de los archivos XML.
Estrategias y/o Observaciones:	<p>Con respecto a estos objetivos se crean estrategias en cuanto a:</p> <ol style="list-style-type: none"> 1. Arquitectura del Modelo de Integración de audio: al iniciar el proyecto se realizo una basta exploración tecnológica para la correcta elección de las librerías de audio inicial. Se eligen las que más documentación y funcionalidades poseen, en este caso son BASS y DirectSound, con las cuales una aplicación puede reproducir varios formatos de audio, efectos de sonido y ejecutar muchas acciones como adelantar pista, retroceder pista, pausa, etc. 2. Herramienta reproductora de audio: se hace necesario crear la herramienta reproductora de audio como un elemento que ayude a testear y probar si la arquitectura de modelado de audio en realidad funciona, y así evitar futuros percances al momento de aplicar sonidos y efectos de sonidos en la etapa de desarrollo del videojuego. 3. XML y XSD: Al crear y cargar un mundo con sus respectivas características, es mucho más útil y fácil hacerlo utilizando estos medios. El desarrollador ahorra tiempo y disminuye la complejidad del código en el programa.

No. De Objetivo: 2 Objetivo Especifico.	
Descripción del Objetivo:	<p>Crear un nivel de un videojuego que cumpla las siguientes especificaciones:</p> <ul style="list-style-type: none"> • Sea elaborado a partir de un guión basado en la detección de síntomas y factores de riesgo de la tuberculosis pulmonar. • Considere las características básicas de un juego de aventura 3D, tal como un escenario ambientado según la trama del juego, personajes tridimensionales genéricos, uso del teclado para la interacción con los personajes, entre otras. • Utilice la implementación del esquema que incorpora el audio al motor gráfico.
Productos Esperados:	<ol style="list-style-type: none"> 1. Videojuego. 2. Guión. 3. Imágenes 3D.
Resultados Obtenidos:	<ol style="list-style-type: none"> 1. Videojuego. 2. Guión. 3. Imágenes 3D.
Medios de Verificación:	<ol style="list-style-type: none"> 1. En el Capitulo 4 DISEÑO E IMPLEMENTACION DEL VIDEOJUEGO se hace referencia a todos los pasos que se siguen para la implementación del videojuego. 2. En el Capitulo 4.2.3 GUION DEL VIDEOJUEGO esta la documentación sobre el guión en el cual se basa el videojuego. 3. En el Capitulo 4.2.2 DESCRIPCION GENERICA DEL VIDEOJUEGO se pueden visualizar las imágenes en 3D que fueron utilizadas para poblar el mundo del videojuego.
Estrategias y/o Observaciones:	<p>Con respecto a estos objetivos se crean estrategias en cuanto a:</p> <ol style="list-style-type: none"> 1. Videojuego: Utilizar documentos XML para que permita cargar de manera fácil y rápida los diferentes objetos que se utilicen en el videojuego 2. Guión: Trabajar coordinadamente con el diseñador gráfico, para que cada una de las partes tengan claro cual es el objetivo a lograr al crear el videojuego basado en este documento. 3. Imágenes 3D: Investigar previamente el formato de las imágenes que carga el motor, debido a que el tiempo para poder exportar los modelos creados por el diseñador gráfico al formato deseado puede extenderse demasiado.

No. De Objetivo: 3 Objetivo Especifico.	
Descripción del Objetivo:	Determinar, mediante la realización de una prueba de campo de tipo tradicional con estudiantes entre 10 y 14 años de un colegio público o privado de la ciudad de Popayán, si la utilización del videojuego y del audio facilita el aprendizaje de la temática referente a la tuberculosis pulmonar.
Productos Esperados:	<ol style="list-style-type: none"> 1. Documentación de la Prueba de Campo. 2. Resultados de la Prueba.
Resultados Obtenidos:	<ol style="list-style-type: none"> 1. Documentación de la Prueba de Campo. 2. Resultados de la Prueba. 3. Documentación relacionada a los síntomas y factores de riesgo de la tuberculosis pulmonar.
Medios de Verificación:	<ol style="list-style-type: none"> 1. En el Capítulo 5 PRUEBA DE CAMPO se hace referencia a todos los pasos que se siguen para llevar a cumplimiento la experiencia de campo con los niños de un colegio. Posteriormente se pueden ver los resultados de la prueba. 2. En el Capítulo 2.1 TUBERCULOSIS se habla sobre los síntomas y factores de riesgo de esta enfermedad, de aquí sale el documento el cual fue entregado a la persona que dictó la clase tradicional al grupo de niños que no interactuó con el videojuego.
Estrategias y/o Observaciones:	<p>Con respecto a estos objetivos se crean estrategias en cuanto a:</p> <ol style="list-style-type: none"> 1. Documentación de la Prueba de campo: Realizar la prueba en diferentes aulas, para evitar que los grupos de los niños obtengan información sobre la temática proveniente del tipo del modelo educativo diferente al que se les va a aplicar. Las pruebas se realizaron en las instalaciones del colegio San Francisco de Asís, y a pesar de algunos inconvenientes presentados para realizar dichas pruebas como: Protestas en el sector del central de la ciudad, Permiso por parte del coordinador, entre otras, lograron llevar a cabo diferentes sesiones de trabajo, para probar el proyecto con usuarios reales logrando obtener resultados satisfactorios, de la metodología presentada para evaluar los niños sobre la información brindada por el docente y por el videojuego. <p>Esta información recolectada fue analizada y guardada en un archivo Excel, para permitir así crear diagramas más precisos de los resultados.</p>

7 CONCLUSIONES RECOMENDACIONES Y TRABAJO FUTURO

El último capítulo de este documento tiene por finalidad presentar las conclusiones a las que se llegó una vez terminado este trabajo de investigación, así como también plantear una serie de recomendaciones encaminadas a ayudar los futuros trabajos que se realicen a partir de los resultados de este proyecto o que estén relacionados con el área del desarrollo de video juegos. Finalmente, se presentan las actividades futuras a llevar a cabo por parte de los investigadores y que buscan dar continuidad a la labor iniciada con la presente investigación.

7.1 CONCLUSIONES

- El modelo creado para la integración de audio al motor gráfico, permite que MOGRE pueda reproducir diferentes formatos de audio, esto se verifica en el momento en que se le adicionan sonidos y efectos de sonidos al videojuego.
- El videojuego se construye en base a la arquitectura establecida, demostrando así que dicha arquitectura permite no sólo la elaboración de proyectos similares sino que puede ser la base para la creación de trabajos futuros relacionados con el área de los videojuegos.
- En el desarrollo del videojuego se pudo observar y aprender claramente, que la relación interdisciplinaria entre desarrolladores y diseñadores es fundamental para el éxito del mismo. Esto es debido a que los desarrolladores pueden implementar complejas funcionalidades pero gráficos muy pobres, cuya estética y semiótica no se asemejan a los productos que un diseñador puede construir y quien por su parte no está capacitado para la construcción de software altamente funcional. Esto fue evidente en el aula de clase cuando se cursó la electiva de desarrollo de videojuegos, donde los diseños y entregables de los modelos realizados por parte de los ingenieros, carecían de muchas características, las cuales los diseñadores gráficos si habían incluido, esta es la razón por la cual se hace necesario contar con dichas personas al momento de elaborar el videojuego.
- Es importante para la consecución de un buen videojuego de aventura que el desarrollo del guión esté completo y bien planteado antes de iniciar a desarrollar el motor de juego y los recursos gráficos. Esto es porque en los videojuegos de aventura el diseño del guión de

juego es donde se plantean los requerimientos del software y las pautas artísticas que se deben seguir. De esta forma, es necesario que el guión literario esté terminado con la mayor brevedad, así como el diseño de personajes, conversaciones, mundos entre otras. Sin embargo, es posible realizar cambios al guión durante el desarrollo del juego por causas de restricciones del motor o por limitantes de tiempo y espacio, pero se debe lograr en lo posible, que éstos cambios sean de poco impacto para la lógica general del juego.

- El uso de XML y XSD, permite al desarrollador reducir la complejidad en el diseño, en la implementación y en el tiempo requerido para la culminación exitosa del proyecto. Además también hace que el videojuego sea extensible simplemente con editar estos archivos, siempre y cuando se conozca la lógica de éste.
- La propuesta documental XML para videojuegos de aventura <e-engine> ha demostrado, ser una fuerte base para el desarrollo de los mismos, aunque algunos de sus elementos no se implementaron <e-engine> demostró ser flexible al escoger que implementar y como implementarlo en el juego lo cual es decisión del desarrollador.
- Uno de los inconvenientes al crear los archivos XML del motor se encuentra en que es algo complicado para los diseñadores el manejo de posiciones, vectores, velocidades en este tipo de archivos y por otro lado los parámetros de recursos artísticos que se colocan en el archivo podrían no cumplir con los supuestos que el motor asume. Debido a esto, los desarrolladores deben tener una clara y continua comunicación con los diseñadores para que de esta manera se conozcan aspectos concretos de los recursos gráficos, por ejemplo: estructuras de modelos 3D, transparencias en PNG's, manejo de texturas, materiales y luces etc. Por su parte, los diseñadores deben empaparse también de aspectos relativos al motor de juego como manejo de vectores, posiciones, velocidades, estructuras de datos y el manejo del motor básico de renderizado (en este caso OGRE), todo esto con el fin de que al desarrollar los productos de parte y parte también se piense en los limitantes que se puede tener debido a las restricciones que los otros equipos de trabajo tengan.
- La implementación de patrones como Game Loop, Game State, Polimorfismo y Singleton cumplen un papel muy importante para el desarrollo de videojuegos y se hace necesario aprender a utilizarlos y aplicarlos.
- En cuanto a la experiencia de campo, se puede decir que los resultados pueden no ser los más confiables por el pequeño tamaño de la muestra experimental y hay que tener en

cuenta que toda la información que brinda el videojuego, se encuentra disponible en su totalidad solo cuando se termina con éxito el videojuego.

7.2 RECOMENDACIONES

A los investigadores y personas interesadas en dar continuidad al trabajo iniciado con este proyecto se les recomienda proseguir con la implementación de características que aún no están presentes en el juego como el uso de mini mapas, inventarios, persistencia entre otras. Además, sería muy enriquecedor el iniciar el desarrollo de otro tipo de juegos diferentes a los de aventura, tal como los juegos de estrategia, habilidad y acción.

Por todo el trabajo que involucró la construcción de Simon y la Tuberculosis y por todos los conocimientos que se tuvieron que aplicar, se puede afirmar que en el proceso de desarrollo de un juego se deben complementar muchos de los conceptos tratados en las diferentes asignaturas del programa de Ingeniería de Sistemas de la Universidad del Cauca. Por tal motivo se recomienda integrar las actividades realizadas en asignaturas complementarias, como es el caso de la electiva de desarrollo de software orientado a agentes y las asignaturas en las que se tratan tópicos relacionados con matemáticas en 3D, ingeniería de software, teoría de la computación y otros.

De igual manera se recomienda que para proyectos de este tipo se pueda contar, no sólo con el apoyo de diseñadores gráficos, sino con muchas más personas en diversas áreas, como por ejemplo literatos e historiadores que den forma a la historia y en general al guión del juego y matemáticos que ayuden en la tarea de reforzar los conceptos necesarios para un mejor trabajo con objetos en tres dimensiones.

Así mismo se recomienda la creación de una electiva al interior del programa de Ingeniería de Sistemas de la Universidad del Cauca en la que se ofrezca la posibilidad de conocer y desarrollar proyectos en el área de creación de juegos y que además se realice un trabajo interdisciplinario con docentes y estudiantes del programa de Diseño Gráfico, de manera que se puedan complementar las actividades propias del diseño de juegos.

7.3 TRABAJO FUTURO

La labor de la creación del juego no debe quedar sólo con lo hecho hasta ahora en Simon y la Tuberculosis, sino que se debe complementar el juego de manera tal que se brinde mayor funcionalidad y más características que no se pudieron llevar a cabo en este proyecto debido a

limitantes de tiempo y recursos. Para lograr este objetivo, será necesario buscar instituciones que apoyen la investigación y el trabajo innovador en nuestro medio.

Con esto se podría avanzar mucho más en cuanto al proceso de desarrollo de juegos y se podría explorar temas que no fueron incluidos dentro de Simon y la Tuberculosis, tales como:

- Los puzzles: debido a la potencial complejidad de estos, no se implementan en este primer nivel, aunque son un gran aporte de reto para el jugador.
- Cargar/Salvar Partida: Dado que el primer nivel del videojuego es muy básico, sus fines comprenden el jugar el videojuego como elemento para la investigación experimental. Se convino que en la validación de este no se hace necesario Cargar o Salvar partidas ya que sólo se implementa una escena del juego y sólo se jugará una vez.
- Inventario: Aunque es una buena opción de visualización de los elementos del personaje principal, no se implementa el inventario debido a que los objetos recogidos en el primer nivel no son muchos y no se ve la necesidad de administrarlos.
- Como se indicó en los objetivos, sería bueno pensar en la integración del primer nivel del videojuego Síntomas y factores de riesgo de la Tuberculosis Pulmonar con el segundo nivel sobre tratamiento de la Tuberculosis Pulmonar que esta siendo realizado en este momento por otro grupo de tesis de la Universidad del Cauca. Con el fin de extender el videojuego y hacerlo más extenso aún.
- Con el fin de dar continuidad a este trabajo no sería suficiente con expandir el videojuego, se hace necesario realizar una nueva experiencia de campo tal vez de tipo experimental seleccionando una muestra de la población más numerosa para obtener más datos y disminuir el margen de error que se podría presentar.
- Según la información obtenida en la experiencia de campo al interactuar con los alumnos que realizaron la práctica, es bueno pensar en dar mayor variedad de los modelos de Simon y la tuberculosis.

Teniendo en cuenta todas estas recomendaciones se lograría una investigación y aprehensión de conocimientos que redundaría en beneficios personales y en mejora del producto con posibilidades reales de comercialización.

8 GLOSARIO Y BIBLIOGRAFIA

En este capítulo se describe el glosario del presente trabajo y la bibliografía utilizada

8.1 GLOSARIO

Interacción: se refiere a una acción recíproca entre dos o más objetos con una o más propiedades homólogas.

Videojuegos: También conocidos como juegos de video, son el resultado del trabajo de personas especializadas en este campo, donde desarrollan un programa informático o software, creado para divertir a personas que interactúan con la consola donde se encuentra instalado. Los videojuegos recrean entornos virtuales con una variedad de animaciones y personajes, donde el jugador manipula uno o varios elementos de dicho entorno para poder lograr un objetivo.

Enfermedades infectocontagiosas: es la manifestación clínica consecuenta a una infección provocada por un microorganismo —como bacterias, hongos, virus, protozoos, etc.

Tuberculosis pulmonar: es una enfermedad infectocontagiosa bacteriana producida por una micobacteria del complejo *Mycobacterium*: *M. Tuberculosis*, *M. Bovis*, o *M. africanum*. Fue una de las primeras causas de muerte en todo el planeta y dio origen a numerosas estrategias terapéuticas, desde las aguas medicinales, las sangrías hasta los más recientes esquemas de antibioticoterapia combinada.

TB: acrónimo de tuberculosis Pulmonar

Organización Mundial de la Salud: es el organismo de la Organización de las Naciones Unidas (ONU) especializado en gestionar políticas de prevención, promoción e intervención en salud a nivel mundial. Organizada por iniciativa del Consejo Económico y Social de la ONU, se redactan los primeros estatutos de la OMS. La primera reunión de la OMS tuvo lugar en Ginebra, en 1948.

Los principales cometidos de la Asamblea Mundial de la Salud son aprobar el programa y el presupuesto de la OMS para el siguiente bienio y decidir las principales cuestiones relativas a las políticas sanitarias. Tal y como establece su Constitución, el objetivo de la OMS es que todos los pueblos de la tierra puedan gozar del grado máximo de salud que se pueda lograr. La Constitución de la OMS define la salud como "un estado de completo bienestar físico, mental y social", y no solamente como la ausencia de afecciones o enfermedades.

Factores de riesgo: es algo que incrementa la posibilidad de contraer una enfermedad o padecimiento.

OMS: Acróstico de Organización Mundial de la Salud

Computadores para Educar: Computadores para Educar es un programa de la Presidencia de la República, con la participación del Ministerio de Comunicaciones quien en la actualidad ejerce la coordinación nacional y contribuye con la financiación del programa a través del Fondo de Comunicaciones; el Ministerio de Educación quien apoya en la definición de los criterios y distribución equitativa de los equipos, y coordina la articulación con otros programas educativos; el SENA en la capacitación y selección del personal técnico de los Centros de Reacondicionamiento; y varios socios de la empresa privada.

CPE: Acróstico de computadores para educar.

Motor gráfico: Estos son un conjunto de librerías y funciones de alto nivel, implementadas para el manejo de gráficas en tiempo real, estas pueden ser en 2D o 3D. El objetivo del motor gráfico, es facilitar el trabajo al desarrollador permitiéndole una manipulación e implementación realizable de operaciones robustas durante el desarrollo de cualquier tipo de gráficas.

Motor de Videojuego: es el sistema que presenta y manipula el ambiente en tres dimensiones, además controla el sonido y permite la interacción con el jugador

OGRE: (Motor de renderizado gráfico orientado a objetos) es un motor para el renderizado gráfico orientado a objetos, esta escrito en C++ y diseñado para que los desarrolladores de software construyan aplicaciones que involucren gráficas 3D de una manera más fácil e intuitiva

Trabajo interdisciplinario: Estudios u otras actividades (juegos) que se realizan mediante la cooperación de varias disciplinas.

Simulación en tiempo real: Hacer que la aplicación informática muestre las imágenes sintéticas a medida que se van produciendo, de forma que reflejen los cambios producidos por las acciones del usuario sobre el programa (Interactividad). Estos cambios suelen responder a la representación de un fenómeno real cuya evolución temporal se intenta replicar (por ejemplo, el vuelo de un avión manejado por un piloto.)

Prueba de campo: Las pruebas de campo confirman los beneficios, preferencias y valor en la vida real. Ellas identifican problemas con los productos. Forman la parte más relevante del testado y son la base más veraz de las reclamaciones válidas de un producto. Las pruebas de campo requieren muchos datos para obtener una imagen completa.

Prueba de BK esputo: Método utilizado para examinar que tan avanzada esta la enfermedad (TB.) analizando la flema del paciente

Videojuegos educativos: videojuegos que resultan positivos para desarrollar el aprendizaje.

Libre distribución: En el campo de la informática, se trata de un software por el cual no hay que pagar costo alguno por su uso.

Gráfica 3D: es la que esta construida en tres dimensiones: largo, ancho y profundidad. En el caso de los videojuegos cuando estos contienen escenas en 3D, estas escenas son conocidas como realidad virtual, la que ofrece una sensación de realismo al jugador.

DirectX: es una colección de API's (Interfaz de programación de aplicaciones) creadas para facilitar tareas relacionadas con la programación de juegos en la plataforma Microsoft Windows.

Colisiones: Es la relevancia de los bordes en la estimación del tiempo de colisión relativo entre dos objetos

MOGRE: Versión administrada de OGRE es decir esta versión se ejecuta sobre el .NET Framework, o sea aquel compilado en MSIL (MSIL significa Microsoft Immediate Language)

IAudioDevice: Interface que recibe los parámetros necesarios de determinada clase para reproducir audio.

Videojuego de aventura: Videojuegos caracterizados por la investigación, la exploración, la solución de rompecabezas, la interacción con caracteres del juego (personajes), y un enfoque en relato en vez de desafíos basados en reflejos. La vasta mayoría de juegos de aventura son juegos de computadora, aunque los juegos de aventura basados en consola no son desconocidos. A diferencia de muchos otros géneros de juegos, el enfoque en una historia de los géneros de aventura permite en gran medida tomar prestadas cosas de otros medios basados en narrativa, tales como la literatura y las películas. Los juegos de aventura abarcan una amplia variedad de géneros literarios, incluyendo la fantasía, ciencia ficción, misterio, horror, y la comedia. Notables juegos de aventura incluyen Zork, King's Quest, The Longest Journey, The Secret of Monkey Island, Indiana Jones and the Fate of Atlantis, Gabriel Knight, Myst, y The Last Express, sin olvidarse de la saga de Zelda. Casi todos los juegos de aventura están diseñados para un solo jugador, dado que hay un fuerte énfasis en la historia y el personaje se hace difícil el diseño de múltiples jugadores.

Stills: imágenes que se despliegan en la pantalla, las cuales contienen descripciones (en formato texto o símbolos) y audio

Storyboard: Es un conjunto de ilustraciones mostradas en secuencia con el objetivo de servir de guía para entender una historia, previsualizar una animación o seguir la estructura de una película antes de realizarse o filmarse.

Mundo: Núcleo central del videojuego, lugar donde ocurren los eventos de la narración de la historia

8.2 BIBLIOGRAFIA

- [1] **X. BRINGUE**, “El público infantil es el sector que menos televisión consume en España,” disponible al 05 de agosto de 2006 en: <http://www.unav.es/noticias/110806-02.html>
- [2] **D.G. BLÁZQUEZ**, “Benchmark 3D para móviles,” disponible al 08 de agosto de 2006 en: <http://www.exelweiss.com/blog/>
- [3] **R. GOMEZ**, “La tele pierde, consolas y móviles ganan,” disponible al 08 de agosto de 2006 en: http://www.elpais.es/articulo/sociedad/tele/pierde/consolas/moviles/ganan/elpporsoc/20060813elpepisoc_2/Tes/
- [4] **C.A. FONSECA**, “SALUD, ENFERMEDAD Y DESARROLLO SOCIOECONÓMICO,” disponible al 19 de noviembre de 2006 en: http://www.deslinde.org.co/Dsl32/Salud,_enfermedad_y_desarrollo_socioeconomico.htm
- [5] **MINISTERIO DE SALUD - DIRECCIÓN GENERAL DE PROMOCIÓN Y PREVENCIÓN**, “Guía de atención de la Tuberculosis Pulmonar y Extrapulmonar,” disponible al 15 de diciembre de 2006 en: http://www.col.ops-oms.org/prevencion/tuberculosis/guia_tuberculosis.pdf
- [6] **UNIVERSIDAD NACIONAL DE COLOMBIA**, “Sector salud,” disponible al 11 de septiembre de 2006 en: http://www.virtual.unal.edu.co/cursos/ciencias/2000088/lecciones/seccion4/capitulo10/04_10_01.htm
- [7] **ORGANIZACIÓN MUNDIAL DE LA SALUD**, “La OMS lanza una ofensiva para ampliar rápidamente las medidas de lucha contra la tuberculosis y el VIH,” disponible al 04 de febrero de 2007 en: <http://www.who.int/mediacentre/news/releases/2004/pr5/es/index.html>
- [8] **MINISTERIO DE COMUNICACIONES DE COLOMBIA**, “Qué es CPE,” disponible al 26 de enero de 2007 en: http://www.computadoresparaeducar.gov.co/que_es.html

- [9] **UNIVERSIDAD DE BARCELONA**, “Crean una Web para enseñar a padres y educadores cómo deben usar los videojuegos los niños,” disponible al 28 de junio de 2007 en: <http://www.aprendeyjuegaconea.com>
- [10] **PÁGINA PRINCIPAL DE OGRE**, “WIKI,” disponible al 22 de Junio de 2008 en: <http://www.ogre3d.org>
- [11] **N. INSUASTY PLAZA**, LOS ESCOLARES Y LA LECTURA: EL COMPORTAMIENTO LECTOR DE LOS NIÑOS EN LA ESCUELA PRIMARIA EN COLOMBIA, Kapeluz, 1984.
- [12] **C.K. MONTOYA**, “El juguete y el desarrollo físico y psíquico del niño y la niña,” disponible al 16 de julio de 2007 en: <http://www.educacioninicial.com/ei/contenidos/00/0600/602.ASP>
- [13] **I. MARTÍNEZ SARANDONA**, “El juego en la teoría de Jean Piaget,” disponible al 28 de noviembre de 2006 en: <http://sepiensa.org.mx/contenidos/2004/irene/eljuegojeanpiaget/eljuegojeanpiaget.html>
- [14] “Video juego educativo en prevención de riesgos laborales,” disponible al 28 de abril de 2007 en: <http://www.exelweiss.com/desarrollo/videojuegos/pc/el-desafio-de-worky.php>
- [15] “Educación y aprendizaje significativo,” disponible al 26 de Octubre de 2008 en : <http://www.monografias.com/trabajos14/educacsignif/educacsignif.shtml>
- [16] **G. RUEDA PÉREZ**, “Evaluación y manejo de la tuberculosis pulmonar y extrapulmonar,” disponible al 22 de junio de 2006 en: <http://www.medilegis.com/bancoconocimiento/T/Tribunamedicav102n1/pamcvaab.htm>
- [17] **ORGANIZACIÓN MUNDIAL DE LA SALUD**, “Investigaciones por valor de US\$ 30 millones para elaborar nuevas pruebas de detección de enfermedades infecciosas mortales,” disponible al 23 de junio de 2006 en: <http://www.who.int/mediacentre/news/releases/2003/pr42/es/index.html>
- [18] **INDICE LATINO**, “Videojuegos Clásicos,” disponible al 14 de enero de 2008 en: <http://indicelatino.com/juegos/>
- [19] **J.L. NÚÑEZ**, “Desarrollo y programación de un videojuego,” disponible al 12 de julio de 2006 en: <http://www.ucol.mx/bit/noticia.php?id=656>

- [20] **C. JORDI**, “Comparativa de motores gráficos para videojuegos,” disponible al 17 de agosto de 2006 en :<http://ima.udg.es/iiia/GGG/TIC2001-2416-C03-01/docs/Engines.pdf#search=%22motores%20para%20videojuegos%22>
- [21] **UNIVERSIDAD EUROPEA DE MADRID**, "motores de juegos", Disponible al 03 de abril de 2008 en: <http://www.esi.uem.es/jccortizo/motores.html>
- [22] **H.G. HOFFMAN**, “Videojuegos para el dolor,” disponible al 16 de septiembre de 2006 en :
<http://www.terra.com.mx/formato.aspx?ref=0&articuloid=137822&paginaid=1&formatoid=1&canal=tecnologia>
- [23] **DEVMÁSTER.NET**, “DevMaster's Game and Graphics Engines Database,” disponible al 21 de agosto de 2006 en: <http://www.devmáster.net/engines/>
- [24] **DEVMÁSTER.NET**, “Engine Details,” disponible al 21 de agosto de 2006 en:http://www.devmáster.net/engines/engine_details.php?id=2
- [25] **OGRE**, " What Is OGRE?," disponible al 03 de abril de 2008 en: <http://www.ogre3d.org/>
- [26] **OGRE**, " Todo sobre OGRE3D," disponible al 03 de abril de 2008 en: <http://www.ogre3d.org/phpBB2/viewforum.php?f=11>
- [27] **WIKI DE OGRE**, "MOGRE (Managed Ogre)," disponible al 03 de abril de 2008 en: <http://www.ogre3d.org/wiki/index.php/MOGRE>
- [28] **WIKI DE OGRE**, "Projects using MOGRE," disponible al 03 de abril de 2008 en: http://www.ogre3d.org/wiki/index.php/Projects_using_MOGRE
- [29] **D.F. ZALDÍVAR PÉREZ**, “Música bienestar y salud,” disponible al 13 de agosto de 2006 en: <http://saludparalavida.sld.cu/modules.php?name=News&file=article&sid=206>
- [30] **M. MOREIRA CURY**, “La música de los videojuegos: modalidades de uso y su relación con el imaginario social,” disponible al 16 de agosto de 2006 en: http://www.scielo.cl/scielo.php?pid=S0716-27902005020300023&script=sci_arttext
- [31] **J.L. FERNÁNDEZ FASANI**, “Introducción a sonido en videojuegos,” disponible al 08 de diciembre de 2006 en: http://www.codepixel.com/index.php?searchword=sonido1&option=com_search&Itemid=5

- [32] **UNSEENDEVELOPMENTS**, “BASS audio library,” disponible al 28 de octubre de 2006 en: <http://www.un4seen.com/>
- [33] **OPENAL**, “Cross-Plataform 3D Audio,” disponible al 28 de octubre de 2006 en: <http://www.openal.org/>
- [34] **EFRAÍN OCAMPO**, “Hallan que PC y videojuegos perjudican la salud”, disponible al 22 de Enero de 2008 en: <http://www.netmedía.info/articulo-55-7607-1.html>
- [35] **MÁSTER MAGAZINE**, “¿Qué efectos tienen sobre nuestra salud los videojuegos e Internet?”, disponible al 20 de Junio de 2008 en: <http://www.mástermagazine.info/articulo/11927.php>
- [36] **CHRISTIAN ROJO**, “Videojuegos y Salud”, disponible al 9 de Julio de 2008 en: <http://es.videogames.games.yahoo.com/08102007/57/videojuegos-salud.html>
- [37] **ALICIA AULT**, “Videojuegos que favorecen la Buena salud” , disponible al 9 de Julio de 2008 en: <http://www.clarin.com/suplementos/informatica/2005/11/02/f-00501.htm>
- [38] **EDUCACIÓN EN VALORES**, “Re-Mission Un videojuego para adolescentes y jóvenes con cáncer” , disponible al 8 de Agosto de 2006 en: http://www.educacionenvalores.org/article.php3?id_article=1028
- [39] **PEPE SANCHEZ**, “Videojuegos que curan”, disponible al 28 de Diciembre de 2006 en: <http://www.elpais.com/articulo/ocio/Videojuegos/curan/elpepateccib/20061228elpepateccib/8/Tes>
- [40] “Afirman que los videojuegos mejoran la sociabilidad y la capacidad de liderazgo”, disponible al 22 de Diciembre de 2005 en: <http://www.weblog.com.ar/2005/12/22/afirman-que-los-videojuegos-mejoran-la-sociabilidad-y-la-capacidad-de-liderazgo/>
- [41] Referencia 40
- [42] **TODD BARRON**, “Strategy game programming with DirectX 9.0.”, Wordware Publishing Inc.: 2003
- [43] **FRANCOIS DOMINIC LARAMEE**, “The game design process.”, disponible al 28 de Diciembre de 2006 en: <http://www.gamedev.net/reference/list.asp?categoryid=23#32>
- [44] **CHRIS CRAWFORD**, “The art of computer game design”, disponible al 28 de Diciembre de 2006 en <http://www.vancouver.wsu.edu/fac/peabody/game-book/Coverpage.html>
- [45] **GD GAMES**, “Calificación de Juegos: ESRB”, disponible al 1 de Noviembre de 2008 <http://www.gdgames.com.mx/ratings.html>
- [46] **OGRE3D**, “Managing Game States with OGRE”, disponible al 20 de Octubre de 2008 en http://www.ogre3d.org/wiki/index.php/Managing_Game_States_with_OGRE

- [47] **KOEN WITTERS**, “The Game Loop”, disponible al 15 de Septiembre de 2008 en <http://dewitters.koonsolo.com/gameloop.html>
- [48] **P. MORENO-GER, J. L. SIERRA, I. MARTÍNEZ-ORTIZ and B. FERNÁNDEZ-MANJÓN**, “A documental approach to adventure game development”, Science of Computer Programming. 2007