



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

MOVILIDAD 3D EN USUARIOS FINALES NO VIDENTES

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

JULIO FELIPE VILLANE SANSARRICQ

PROFESOR GUÍA:

JAIME HERNÁN SÁNCHEZ ILABACA

MIEMBROS DE LA COMISIÓN:

JUAN FERNANDO ÁLVAREZ RUBIO

DIONISIO ALEJANDRO GONZÁLEZ GONZÁLEZ

SANTIAGO DE CHILE

ENERO 2008

Resumen Ejecutivo

“El Diseño de entornos virtuales basados en audio, tareas cognitivas y métodos de usabilidad pueden ser utilizados para desarrollar y ejercitar procesos cognitivos, tales como relaciones espacio-temporales, memoria, resolución de problemas, movilidad, orientación, abstracción espacial, percepción háptica e integración cognitiva”

El desarrollo de habilidades de movilidad y orientación en personas no videntes es fundamental para el desarrollo de una vida independiente. Sin embargo actividades orientadas en reforzar esta línea de conocimiento requieren interacciones directas con los espacios o lugares, y en muchos casos de la asistencia de un educador o acompañante.

Sin embargo es posible, colocar al alcance de las necesidades la tecnología a través del uso de software mediante el cual se puedan construir espacios virtuales a partir de edificios o construcciones complejas ya existentes, de modo que usuarios no videntes a través de retroalimentación auditiva sean capaces de recorrer estos espacios desconocidos sin necesidad de una interacción directa.

El objetivo central de esta memoria es utilizar una herramienta de generación de escenarios tridimensionales, a través del cual se llevará a una representación virtual de una construcción real que posee varios niveles estructurales o pisos (en particular la estación de metro Universidad de Chile) que, combinado con una serie de pistas auditivas, podrá ser recorrido por un usuario final ciego, y asimilado en su modelo mental para facilitar su orientación y movilidad en edificios cuya estructuración sea compleja.

Para lograr esto se utilizará el *Unreal Engine 2*, que corresponde a un motor para juegos de PC y consolas escrito en C++. Algunas de las características que vuelven a este motor en un elemento interesante en la construcción de escenarios tridimensionales son: los bajos requisitos de hardware que exige, posee un editor incorporado que se utiliza para construir escenarios y la calidad gráfica y de sonido que es capaz de generar es sumamente eficiente y realista, entre otras.

Contenido

1	Introducción.....	9
1.1	Motivación.....	10
1.2	Objetivos.....	11
1.2.1	Objetivo General.....	11
1.2.2	Objetivos Específicos.....	11
1.3	Glosario.....	12
2	Marco Teórico.....	13
2.1	Interfaces para usuarios ciegos.....	13
2.2	Niños Ciegos y Aprendizaje.....	15
2.3	Realidades Virtuales.....	16
2.4	Niños Ciegos y Realidades Virtuales.....	18
2.5	Sonido Localizado.....	20
2.5.1	El retardo temporal.....	20
2.5.2	Efecto Hass.....	20
2.5.3	Longitud de Onda.....	21
2.5.4	Enmascaramiento.....	21
2.6	Interfaces de Audio 3D en discapacitados visuales.....	22
3	Requerimientos.....	23
3.1	Objetivos.....	23
3.2	Características de los usuarios.....	23
3.3	Alcance.....	23

3.4	Requisitos de Hardware/Software	23
3.5	Descripción de <i>Unreal Engine 2</i>	24
4	Diseño	26
4.1	Modelo de Diseño.....	26
4.2	Etapas de Apresto	26
4.3	Metro Universidad de Chile – Nivel Superior	28
4.4	Metro Universidad de Chile – Nivel Superior e Intermedio.....	29
4.5	Retroalimentación Auditiva	30
4.6	Patrones de Sonidos	31
4.6.1	Patrón de Sonido Simple	31
4.6.2	Patrón de Dirección.....	31
4.6.3	Patrón de Definición de Sectores.....	31
4.6.4	Patrón de Bloqueo de Sectores.....	32
5	Etapas de Aplicación de Conocimiento.....	32
6	Implementación.....	33
6.1	Introducción al <i>Unreal Engine</i>	33
6.1.1	Motor Gráfico.....	33
6.1.2	Motor de Sonido	34
6.1.3	Motor Físico.....	34
6.1.4	Interacción de Componentes	34
6.1.5	Recursos	35
6.1.6	<i>UnrealEd</i>	37
6.2	<i>UnrealEd</i> : Funciones importantes	38
6.2.1	Introducción al <i>UnrealEd</i>	38

6.2.2	Componentes del <i>UnrealEd</i>	39
6.2.3	Creación de Mapas mediante Métodos Sustractivos	41
6.2.4	<i>Movers</i> : Elementos Interactivos.....	48
6.2.5	<i>Triggers</i> : Iniciadores de Eventos	50
6.3	Implementación de Escenarios.....	57
6.3.1	Implementación de Pistas Auditivas	57
7	Testeo Cognitivo	64
7.1	Usuarios Finales	64
7.1.1	Descripción General	64
7.1.2	Descripción Específica	65
7.2	Eta de Apresto – Primer Mapa	66
7.2.1	Actividades	66
7.2.2	Resultados	66
7.2.3	Comentarios	66
7.3	Eta de Apresto – Segundo Mapa	67
7.3.1	Actividades	67
7.3.2	Resultados	68
7.3.3	Comentarios	69
7.4	Metro Universidad de Chile – Nivel Superior	69
7.4.1	Actividades	69
7.4.2	Resultados	69
7.4.3	Comentarios	70
7.5	Metro Universidad de Chile – Nivel Superior e Intermedio	70
7.5.1	Actividades	70
7.5.2	Resultados	71

7.5.3	Comentarios	71
7.6	Etapa Final – Desplazamiento en el Metro Universidad de Chile	72
7.6.1	Comentarios	72
8	Conclusiones	73
9	Trabajo Futuro y Expectativas	76
10	Referencias	78
11	Anexos	80
11.1	Anexos de Implementaciones de Patrones de Sonido	80
11.1.1	Patrón de Sonido Simple	80
11.1.2	Patrón de Dirección	81
11.1.3	Patrón de Definición de Sectores	83
11.1.4	Patrón de Bloqueo de Sectores	85

Índice de Ilustraciones

Ilustración 1. AudioDoom: Testeos de Usabilidad	15
Ilustración 2. Fotografías tomadas al momento de la interacción con el software	16
Ilustración 3. Escenarios PTSD y de tratamientos dolorosos	18
Ilustración 4. AudioChile: Interfaz e interacción con el usuario final	19
Ilustración 5. Comparación gráfica de las 3 versiones del motor <i>Unreal Engine</i>	24
Ilustración 6. Captura de pantalla de la aplicación <i>Unreal Engine 2</i> para creación y edición	25
Ilustración 7. Mapas Diseñados para la etapa de Apresto	27
Ilustración 8. Escenario del Nivel Superior de la estación del Metro Universidad de Chile	28
Ilustración 9. Escenario del Nivel Intermedio de la estación del Metro Universidad de Chile	30
Ilustración 10. Flujo e Interacción de Componentes del <i>Unreal Engine</i>	33
Ilustración 11. Ciclo de Juego del <i>Unreal Engine</i> con manejo de eventos	35
Ilustración 12. Organización de paquetes junto con la extensión de archivos correspondiente	36
Ilustración 13. Recursos Utilizados en un Mapa	36
Ilustración 14. Interfaz de Usuario del UnrealEd	39
Ilustración 15. Ventana de Propiedades del Actor Camera	41
Ilustración 16. Dimensiones estándar de los personajes creados en el UnrealEd	43
Ilustración 17. Modificación de propiedades de creación de un Cubo	44
Ilustración 18. Resultados al posicionar y escalar un cubo	45
Ilustración 19. Navegador de Texturas	46
Ilustración 20. Creación de un Cubo mediante sustracción	47
Ilustración 21. Ejecución del Nivel	48
Ilustración 22. Navegador de Mallas Estáticas (<i>Static Meshes</i>)	49
Ilustración 23. Selección de <i>Trigger</i> a través del Navegador de Actores	51
Ilustración 24. Agregar un <i>Trigger</i> en la superficie seleccionada	52
Ilustración 25. Ícono que representa un <i>Trigger</i>	52
Ilustración 26. Selección de <i>ScriptedTrigger</i> a partir del Navegador de Actores	54
Ilustración 27. Ejemplo de <i>ScriptedTrigger</i>	56
Ilustración 28. Ejemplo disfuncional de Ubicación Espacial del usuario vía retroalimentación auditiva	58
Ilustración 29. Diagramación de Zonas del escenario Metro Universidad de Chile – Piso Superior	62

Ilustración 30. Diagramación de Zonas del escenario Metro Universidad de Chile – Piso Intermedio	63
Ilustración 31. Foto tomada a la entrada del Colegio Hellen Keller de la comuna de Ñuñoa	64
Ilustración 32. Maqueta utilizada para la segunda parte de la etapa de apresto	68
Ilustración 33. Fotografías tomadas en la segunda parte de la etapa de apresto	69

Índice de Tablas

Tabla 1. Listado de Usuarios Finales participantes en los testeos	65
Tabla 2. Tabla de Categorización para la 1ra parte de la Etapa de Apresto	66
Tabla 3. Resultados obtenidos por alumno para la primera parte de la etapa de Apresto	66
Tabla 4. Tabla de Categorización para la 1ra parte de la Etapa de Apresto	68
Tabla 5. Resultados obtenidos por alumno para la primera parte de la etapa de Apresto	69
Tabla 6. Tabla de Categorización el test del escenario del nivel superior de la estación de metro	70
Tabla 7. Resultados obtenidos por alumno para la etapa de test del escenario del nivel superior de la estación de metro	70
Tabla 8. Tabla de Categorización para el Test basado en escenario multi nivel	71
Tabla 9. Resultados obtenidos por alumno en el Test basado en el escenario multi nivel	71

1 Introducción

En la actualidad existe un gran variedad de aplicaciones que a través de simulaciones de entornos reales (lo que se denomina entornos virtuales) buscan el logro de diferentes objetivos en sus usuarios finales. Los objetivos de estas aplicaciones van desde el diseño de edificios previos a su construcción, animaciones con fines comunicacionales como fotografías y cortometrajes, también usados para construir comerciales y películas, hasta aplicaciones con fines meramente recreativos.

La simulación de ambientes reales ha sido especialmente útil en industrias donde la práctica en un entorno real puede significar inmensurables inversiones monetarias o el potencial riesgo de pérdidas humanas. Es así como aplicaciones de este tipo son utilizadas para entrenar pilotos de aviones comerciales, para entrenar a operadores de la torre de control en los aeropuertos o incluso para entrenamiento militar que involucra el uso de maquinaria sumamente costosa como tanques o aviones de combate.

Por otro lado, las aplicaciones que se enfocan en estos fines recreativos han evolucionado fuertemente a través de esta última década a la par con los avances en tecnología y con ello se acercan cada vez más a representar en forma fiel un escenario real, haciendo uso tanto de recursos gráficos como de sonido. Además, a través de estas aplicaciones es posible la interacción de numerosos jugadores en forma simultánea e inclusive remota.

Adicionalmente se han construido aplicaciones que asisten la construcción de estos ambientes virtuales, que según la especificación de los requerimientos y/o intenciones del usuario permiten construirlos en forma ágil manipulando y controlando una gran variedad de variables en forma simplificada.

Un tema que podría parecer absolutamente tangencial al de los ambientes virtuales es el de las personas con discapacidad, en particular niños y jóvenes con discapacidad visual, que actualmente se encuentran excluidos del uso de aplicaciones basadas en ambientes virtuales, pero sin embargo existen varios trabajos desarrollados que los consideran en conjunto dentro de su desarrollo.

1.1 Motivación

“El Diseño de entornos virtuales basados en audio, tareas cognitivas y métodos de usabilidad pueden ser utilizados para desarrollar y ejercitar procesos cognitivos, tales como relaciones espacio-temporales, memoria, resolución de problemas, movilidad, orientación, abstracción espacial, percepción háptica e integración cognitiva”

“La noche de la ignorancia, de la insensibilidad, es la única niebla impenetrable”, frase célebre de Hellen Keller, de nacionalidad norteamericana, sordomuda desde la infancia, se convirtió en una apasionada escritora y en un modelo a seguir para muchas personas con discapacidad. Luchó por mejorar las posibilidades de crecimiento de personas con discapacidad, en base a su testimonio y experiencia.

En particular, en el caso de las personas con discapacidad visual los entornos virtuales basados en audio se perfilan como una fuente de estudio donde los usuarios son capaces de practicar y comprender y practicar conceptos temporales y espaciales, así como la resolución de problemas, y a través de ello poder practicar y mejorar sus capacidades y destrezas. Estas habilidades y consecuencias dentro del aprendizaje de los usuarios no videntes han sido estudiadas en extenso a través de varios software [1, 2, 3, 4, 5, 6, 7], la mayoría de ellos enfocados principalmente en el área cognitiva.

La principal motivación de este trabajo será el establecer, testear y evaluar distintos patrones que orienten a un usuario final no vidente a desplazarse por un ambiente virtual a través de la utilización de un motor de aplicaciones, en particular el *Unreal Engine 2*, elegido por la cantidad de aplicaciones recreativas que han sido desarrolladas a partir de él, por la cantidad de plataformas que cubre el motor, por los bajos requisitos de hardware que exige dada la calidad con la que genera los escenarios y porque además existe una nueva versión del motor que ha sido publicada este año (*Unreal Engine 3*), que utiliza un sistema de edición muy similar con la versión previa y que se encuentra a la vanguardia en cuanto a desarrollo en distintas plataformas y en la calidad visual y auditiva generada.

A partir de ello se pretende establecer patrones basados en la construcción bajo las herramientas provistas por el sistema, cuya aceptación por parte del usuario final sea lo más alta posible con miras a que nuevos trabajos basados en el motor del *Unreal* puedan apoyarse en este estudio, y no sea necesario establecer nuevos estudios para llegar a las mismas conclusiones.

Una motivación secundaria (pero no menos importante) consistirá en la creación de escenarios basados en lugares reales pero desconocidos por los usuarios finales no videntes de modo de establecer si es posible que a partir de la aplicación es posible aprender a desarrollar destrezas de movilidad y orientación sin haber interactuado en forma directa.

Para ello se utilizará una de las alternativas más potentes que existe actualmente en el mercado como lo es *Unreal Engine 2* [8], creado por *Epic Games* [9], la cual cuenta con un editor que posee una gran

capacidad para generar ambientes en 3 dimensiones con una alta calidad gráfica y de sonido, sin exigir fuertes requisitos de hardware, y además con una versión liberada, de uso libre y gratuito para proyectos con fines educativos y no comerciales.

Es importante decir que los productos de software que existen actualmente que han sido creados para personas con discapacidades no se encuentran a la vanguardia de los avances tecnológicos por lo que para usuarios sin discapacidad se hace más difícil que participen junto a sus pares discapacitados visualmente pues esto implica sacrificar la calidad gráfica que poseen en otras herramientas o juegos a las cuales ya se encuentran acostumbrados. Con el uso de *Unreal Engine 2* o cualquier herramienta que permita crear ambientes virtuales potenciando la parte gráfica existen potencialidades en las cuales los usuarios sin discapacidades en el área de la visión no deberán sacrificar calidad gráfica para compartir en actividades de interacción dentro del mismo entorno virtual con sus pares no videntes. Con lo anterior se puede pensar en crear aplicaciones basadas en entornos virtuales a partir de herramientas como *Unreal Engine* donde no sólo se invite a participar únicamente a usuarios no videntes sino que además se pueda provocar en cierta medida inclusión e integración social de este tipo de usuarios con sus pares videntes.

1.2 Objetivos

1.2.1 Objetivo General

- Implementar un ambiente 3D real a través del cual generar orientación y movilidad en un espacio real pero desconocido, en un usuario final ciego utilizando la herramienta *Unreal Engine 2*

1.2.2 Objetivos Específicos

- Estudiar y analizar las potencialidades del *Unreal Engine 2*
- Diseñar e implementar escenarios 3D utilizando del editor de mapas y motor de sonidos de la Herramienta
- Evaluar la construcción de un modelo mental del escenario construido usando el motor *Unreal Engine 2*
- Testeo y evaluación

La aplicación *Unreal Engine 2* consiste en una herramienta sumamente potente y relevante en el ámbito del desarrollo de software recreativo a nivel mundial, que se destaca por su capacidad de crear entornos complejos otorgando herramientas que, sin la necesidad excesiva de recursos de hardware, entregan un gran realismo en su utilización.

1.3 Glosario

Ciego

Según la Organización Mundial de la Salud, la ceguera es aquella visión menor de 20/400 ó 0.05, considerando siempre el mejor ojo y con la mejor corrección. Se considera que existe ceguera legal cuando la visión es menor de 20/200 ó 0.1 en el mejor ojo y con la mejor corrección; o bien si presenta un campo visual donde el diámetro mejor no exceda los 20 grados, aun presentando una agudeza visual normal.

Son ciegas las personas que tienen sólo percepción de luz sin proyección, o quienes carecen totalmente de visión

Cognición

Proceso mental que ayuda a construir conocimiento.

Entorno Virtual

Hace referencia a la simulación de mundos o entornos en los que un individuo puede interactuar en forma artificial en situaciones que pretenden asemejarse a la realidad o crear una completamente nueva.

Hiperhistoria

Corresponde a la confluencia de varios elementos que involucran la presencia de un mundo virtual constituido de objetos y personajes, donde un individuo o incluso un conjunto de individuos debe asumir el rol de alguno de estos personajes, y donde además existe una historia narrativa intencional que mueve el desarrollo de las interacciones entre las componentes del mundo virtual.

Juego Multiusuario

Corresponde a juegos que han sido diseñados para que exista interacción de varios jugadores al mismo tiempo y en el mismo ambiente virtual. Ofrece la oportunidad de interacción en tiempo real y forma remota en objetivos que deben ser alcanzados de forma cooperativa y/o competitiva.

Motor de Sonido Espacial

Corresponde a un elemento de software que permite la representación de fuentes sonoras a través de sonido espacializado.

Script

También denominado guión o archivo de procesamiento por lotes, corresponde a un programa usualmente simple, que generalmente se almacena en un archivo de texto plano. Los guiones son casi siempre interpretados, pero no todo programa interpretado es considerado un guión. El uso habitual de los guiones es realizar diversas tareas como combinar componentes, interactuar con el sistema operativo o con el usuario

Sonido Espacializado

Sonido generado a partir de una serie de parlantes ubicados en posiciones estáticas pero que al emitir sonidos producen un efecto sobre el oyente en que son capaces de cubrir todo el espacio a su alrededor.

Unreal Engine

Motor de juegos para PC y para consolas creado por la compañía de *Epic Games* a partir del lenguaje de programación C++. Actualmente se encuentran disponibles varias versiones que engloban distintas plataformas y la mayoría de las consolas competitivas del mercado.

2 Marco Teórico

2.1 Interfaces para usuarios ciegos

Al buscar involucrar tecnología con usuarios no videntes se debe proponer un nuevo tipo de interfaz, completamente distintas al tipo de interfaz visual a las que estamos acostumbrados. Es por ello que se desarrolló AudioDoom [1], una aplicación desarrollada con el objetivo de estudiar interfaces eminentemente basadas en sonido para usuarios finales ciegos. Corresponde a una versión basada en sonido del juego computacional Doom.

La idea principal de este proyecto era conocer si existía la posibilidad de desarrollar un ambiente virtual interactivo acústico que permitiera una navegación activa sin hacer uso de pistas visuales. Es por ello que la hipótesis del proyecto descansaba sobre la idea de que los niños ciegos serían capaces de desarrollar estructuras cognitivas de representación espacial a partir de la interacción con ambientes virtuales basados en sonido.

Los usuarios finales con los cuales fue testeado el software fueron alumnos del jardín infantil “Las Luciérnagas” cuyas edades fluctuaban entre los 8 y 11 años.

Para efectos de la investigación se establecieron 4 etapas a través de las cuales se evaluó el software con el aprendiz, éstas son: presteo, exploración, adaptación y apropiación.

La etapa de presteo consiste en un primer acercamiento al software donde el usuario interactúa de modo de conocer las posibilidades que se le otorgan. Esta etapa pretende que el usuario desarrolle autonomía y confianza en la interacción con el software, introduciendo primero los conceptos requisitos para llevar a cabo lo anterior (como por ejemplo la presentación de las partes principales del computador) en caso de que el aprendiz así lo requiera por no tener suficiente experiencia.

La segunda etapa de exploración corresponde a una interacción enfocada en el software prestando especial atención a las características del espacio virtual por el cual navega reconociendo las partes principales del escenario y los movimientos de giro del personaje. Adicionalmente el aprendiz debe representar a través de su propio desplazamiento las direcciones y movimientos que efectuó con el personaje en su interacción.

En la tercera etapa de adaptación se intercalaron 2 actividades: la interacción del aprendiz con el software y la construcción de un modelo sobre una mesa de arena con las características principales del escenario y con los elementos que se encuentran dentro de él (haciendo uso de elementos adicionales para identificarlos) distribuyéndolos según su posición.

En la cuarta y última etapa de apropiación se asume que el alumno ya está consciente y conoce el escenario y sus componentes, base sobre la cual y después de una breve interacción con el software debió construir (haciendo uso de juegos de piezas legos) la representación mental del entorno virtual con el cual habían interactuado.

Los resultados fueron sumamente positivos, pues los modelos construidos resultaron ser muy similares al construido en forma virtual por la aplicación, con lo cual quedó validada la hipótesis, principalmente a través de la última etapa en la que participaron los aprendices y donde se pudo evaluar la complejidad y calidad en las representaciones materializadas de las imágenes mentales construidas a partir del software.



Ilustración 1. AudioDoom: Testeos de Usabilidad

2.2 Niños Ciegos y Aprendizaje

Desde hace varios años que se estudia el impacto del uso de tecnologías en usuarios con discapacidades: juegos computacionales basados en audio, programas multimedia que asisten el aprendizaje de la escritura y lectura hasta bibliotecas acondicionadas con sintetizadores de voz son partes de las herramientas que se encuentran al alcance en nuestro país.

Actualmente el sistema educacional se encuentra en campaña para integrar en los colegios a niños con discapacidad (en particular los no videntes) otorgando la posibilidad de acceso a la educación a través de la tecnología, apoyando y fortaleciendo su desarrollo intelectual.

Se ha comprobado que el uso de estas tecnologías tienen una importante significancia en el desarrollo cognitivo de niños ciegos, asistiendo la enseñanza y acortando cada vez más la brecha existente en la calidad de educación entre niños videntes y no videntes.

En esta línea se han desarrollado varios trabajos que facilitan el aprendizaje de distintas materias en niños ciegos, en particular “AudioMath” [13], un software interactivo desarrollado para desarrollar habilidades matemáticas en la forma de juego similar al Memorice, en donde el objetivo es buscar expresiones equivalentes de un set de opciones.

Es así que se consigue estimular y ejercitar conceptos como: valor posicional, secuencias, descomposición aditiva, descomposición multiplicativa, multiplicación y división.

Adicionalmente se estimula el establecimiento de relaciones de correspondencia y equivalencia, desarrollo de la memoria y distinciones de nociones temporales y espaciales.

En el software multimedia llamado “La Granja de Theo y Seth” [14] también se estimula el aprendizaje matemático a través de metáforas en las cuales los usuarios son capaces de interactuar con distintas operatorias matemáticas.



Ilustración 2. Fotografías tomadas al momento de la interacción con el software

- A. Testeo de usabilidad e impacto cognitivo de “AudioMath”.
- B. Testeo de usabilidad e impacto cognitivo de “La Granja de Theo y Seth”.

En ambos software se evaluó la usabilidad y el impacto cognitivo de los usuarios obteniendo cambios positivos en cuanto al conocimiento matemático. Lo más destacable es que los cambios más acentuados se establecieron en usuarios cuya visión era nula.

2.3 Realidades Virtuales

Acorde con los avances de la tecnología se han desarrollado varios programas con distintas motivaciones en los cuales se simula una realidad a través de la cual uno o varios usuarios interactúan y reciben respuestas muy parecidas a las que se podrían recibir en la realidad.

La Realidad Virtual corresponde a una tecnología que permite a uno o varios usuarios interactuar con un entorno simulado por un computador, que puede asemejarse a la realidad o no. La mayor parte de las realidades virtuales ya desarrolladas se basan en la interacción visual principalmente (ya sea a través de la pantalla de un computador o a través de dispositivos estereoscópicos) y adicionalmente se incluyen otro tipo de interacción auditiva, táctil, etc.

En particular en el trabajo “Expose, Distract, Motivate and Measure: Virtual Reality Games for Health” [10] corresponde a un resumen de una serie de trabajos que se han realizado a partir de

la visión de cómo la creciente popularidad de los juegos digitales y además de los grandes avances en cuanto a realismo gráfico y de sonido, han logrado crear realidades virtuales (VR, *Virtual Reality*) sumamente interactivas y adictivas.

Bajo esta visión se propuso y concretaron estudios donde el fin de estas aplicaciones se pudieran orientar a diferentes áreas, como por ejemplo aplicaciones basadas en juegos con fines educacionales o de entrenamiento, o bien aplicaciones para tratar a pacientes en áreas como las de salud mental y rehabilitación.

Los atributos que se destacan en esta serie de trabajos son exposición (*exposure*), distracción (*distraction*), motivación (*motivation*) y control (*measurement*).

Exposición, el primero, hace referencia a la capacidad de utilizar versiones de juegos existentes de modo de exponer al usuario a un ambiente de modo de contextualizarlo y provocar en el ciertos estímulos relevantes para terapias de costumbre para desordenes de ansiedad como lo son las fobias y el desorden de estrés postraumático (PTSD, *Post Traumatic Stress Disorder*).

Distracción, de modo que aplicaciones del tipo en cuestión son capaces de proveer una distracción de tipo activa, con lo que es posible enfocar la atención de pacientes fuera de procedimientos médicos dolorosos e incómodos como tratamiento de quemaduras, aplicaciones intravenosas y quimioterapia.

Motivación, de modo que en programas de rehabilitación para pacientes que sufren de apoplejía o daño cerebral donde ellos deben efectuar una serie repetitiva de ejercicios en forma frecuente se les pueda agregar un cierto grado de motivación a los participantes.

Control, este atributo hace referencia a que en este tipo de aplicaciones las condiciones, dinámica y estímulos multisensoriales pueden ser controlados en forma precisa de modo de establecer estudios o dosificar el grado de interacción con el paciente.

Dentro de los estudios realizados en el tema por colaboradores en este estudio notaron, en forma anecdótica, que pacientes jóvenes adultos con daño cerebral traumático que demostraban dificultades al realizar las tradicionales tareas cognitivas de rehabilitación podían fácilmente invertir varias horas consecutivas jugando el juego de PC "*SimCity*". Estas observaciones sugerían que los diseños de aplicaciones de realidad virtual con orientación clínica podían verse favorecidos al examinar las fórmulas que usaban los diseñadores de juegos digitales interactivos exitosos, ellas hacen alusión al flujo y variación de los estímulos generados

en el transcurso del juego conectado además con un objetivo progresivo y una estructura de recompensas.

A partir de estos estudios se han desarrollado aplicaciones que involucran uno o más atributos de los mencionados anteriormente para:

- Terapia de exposición con foco en los veteranos de la guerra de Iraq con desorden de estrés postraumático (PTSD) [Ilustración 3 (A)]
- Distracción del dolor en niños que reciben algún tipo de tratamiento doloroso y quimioterapia [Ilustración 3 (B)]
- Motor de rehabilitación para personas con algún tipo de disfunción en su sistema nervioso central (apoplejía y daño cerebral)
- Evaluación y entrenamiento cognitivo/funcional en niños con desórdenes asociados a déficits de atención, síndrome de Down y ansiedad social.

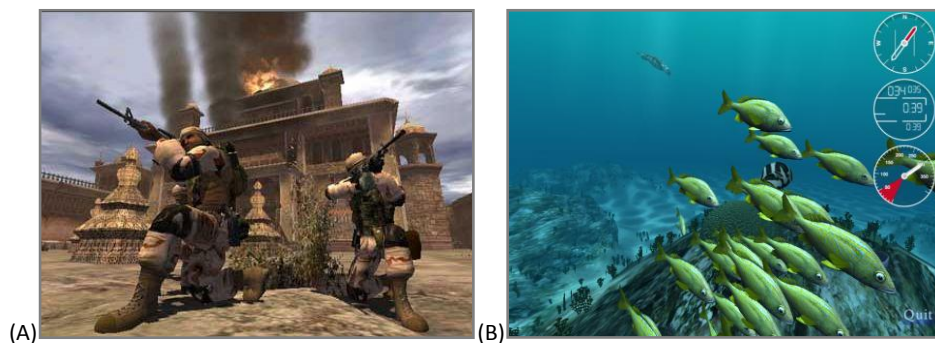


Ilustración 3. Escenarios PTSD y de tratamientos dolorosos

- A. Escenario de guerra generado para utilizarlo en tratamientos de veteranos de guerra con PTSD.
- B. Escenario de un juego usado para distraer a niños en algún tratamiento doloroso o quimioterapia.

2.4 Niños Ciegos y Realidades Virtuales

En la actualidad existen varios software que utilizan realidades virtuales como base de la interacción de usuario.

AudioChile [4] es un proyecto cuyo objetivo principal se basa en el diseño, implementación y evaluación de una aplicación multimedia de tipo RPG, diseñada principalmente para usuarios no videntes, en un ambiente tridimensional y con apoyo de sonido plano y espacial.

Mediante la aplicación se analizó la resolución de problemas por medio de estrategias de planificación como búsqueda aleatoria, ensayo y error y búsqueda sistemática.

El desarrollo del juego se establece mediante la adopción de un personaje donde el usuario va conociendo los distintos aspectos del mundo virtual donde deberá realizar su aventura, en una ambientación, que como su nombre lo indica se establece en el país de Chile, en los lugares más conocidos y característicos de este país, en particular Chiloé, Chuquicamata y Valparaíso, y con ello además permitir que el aprendiz pueda conocer aspectos de la geografía y tradiciones del país.

Para el desarrollo de esta aventura el personaje deberá ir reuniendo una serie de ítems que se podrán ir almacenando en una mochila y así poder utilizarlos o interactuar con ellos en el transcurso y avance del juego. Estos ítems los necesitará para ir completando avances dentro del desarrollo del juego, siendo requisitos para avanzar en el desarrollo de la historia.

Todas las interacciones con los ítems de la mochila, la navegación de los menús y acciones están apoyadas por retroalimentación auditiva, la cual además es utilizada en forma espacializada para asistir la movilidad del personaje en lo relacionado con obstáculos, personajes y límites del entorno.

Los movimientos se pueden hacer exclusivamente en ángulos rectos y extendidos en pro de la simplicidad de acción para el usuario final que se ha definido en la aplicación.



Ilustración 4. AudioChile: Interfaz e interacción con el usuario final

A partir de las conclusiones de este trabajo fue posible señalar que el uso sistemático del software apoyado por la orientación metodológica de tareas cognitivas, permite a los alumnos ciegos aplicar destrezas que ayudan a la resolución de problemas. Además el impacto del software AudioChile en los

alumnos con discapacidad visual nos indica que varios de ellos pudieron anticipar problemas, planificar y aplicar diferentes estrategias de solución de problemas, así como también explicar la estrategia utilizada para resolver los problemas propuestos en el software y aplicarla a otros contextos.

2.5 Sonido Localizado

El sonido localizado corresponde a una habilidad auditiva que permite identificar el punto de origen de una fuente sonora. Esta habilidad es frecuentemente utilizada en la construcción de ambientes virtuales 3D para agregar realismo a la interacción del usuario.

Esta sensación tridimensional que percibe el sistema auditivo se relaciona con la diferencia de amplitud y tiempo que recibe cada oído. Es decir, la localización de los sonidos en el espacio se consigue a través del procesamiento separado de la información que se recibe en cada oído y con la posterior comparación de fase y nivel entre ambas señales recibidas. La fase hace referencia a un estado o situación instantánea de la onda (que varía cíclicamente), mientras que el nivel del sonido a la intensidad o cantidad de energía que contiene la onda acústica.

Para determinar la dirección del sonido el cerebro tiene en cuenta 3 factores que interactúan:

1. El retardo temporal y el efecto Hass
2. La longitud de onda
3. El enmascaramiento

2.5.1 El retardo temporal

Se debe a que un mismo sonido producido por una misma fuente sonora genera percepciones distintas en los oídos. Esto se debe a que los pabellones de nuestros oídos se encuentran separados en forma física y además con una orientación opuesta. Esto provoca que las ondas sonoras deban efectuar un recorrido levemente más largo para alcanzar el oído más alejado de la fuente que para alcanzar el oído ubicado más cercano a la fuente.

Una vez que se transforma el sonido de ambos oídos a una serie de impulsos que finalmente son recibidos por el cerebro e interpretados para dirimir la posición del origen de la fuente con respecto al oyente.

El retardo temporal es más evidente cuando se ha producido un sonido por impulso, por ejemplo un clic o una explosión.

2.5.2 Efecto Hass

También conocido como efecto de precedencia o efecto de prioridad, el efecto Hass afecta a la percepción auditiva de las personas, y fue descrito por el médico alemán Helmut Hass, a quién se debe el nombre.

El efecto describe cómo, a nivel de percepción, si varios sonidos independientes llegan a nuestro cerebro en un intervalo inferior a los 50 ms, éste los fusiona y los interpreta como uno solo. Esto se debe a que el cerebro deja de percibir la dirección y entiende los sonidos posteriores como un eco o reverberación del primero.

Esta interpretación el cerebro la hace en dos formas distintas:

1. Si el retardo llega a un intervalo menor a 5 ms, el cerebro localiza el sonido en función de la dirección que tuviera el primer estímulo aunque los otros provengan desde partes diametralmente opuestas.
2. Si el retardo está entre los 5 y los 50 ms, el oyente detecta un único sonido, pero de intensidad doble y localiza la fuente a medio camino entre todas.

Para que el retardo (o efecto Hass) no determine en nuestro cerebro la dirección del sonido (es decir, para que se perciba el sonido como proveniente de un punto central), la señal atrasada debe tener más nivel que la primera.

La llamada curva de Hass indica la intensidad (expresada en dB) necesaria para lograr una equivalencia en cuanto al retardo en milisegundos entre dos señales. Esta curva de Hass se utiliza en acústica, entre otras cosas, para mantener el estéreo en recintos.

2.5.3 Longitud de Onda

Los sonidos por encima de los 1.000 Hz que tengan una longitud de onda pequeña (inferior a 30 cm), sólo serán escuchados por uno de los dos oídos. Esto se debe a que la cabeza funciona como una pantalla relativa y evita que una parte del sonido alcance al oído que está situado en el lado opuesto a la dirección del sonido.

A la diferencia de fase provocada por la diferente distancia se suma así la diferencia de intensidad, amplitud o nivel acústico, para facilitar la localización espacial de la fuente sonora.

2.5.4 Enmascaramiento

Cuando el oído se encuentra expuesto a dos o más sonidos simultáneos, existe la posibilidad de que uno de ellos enmascare a los demás. Para ser más preciso, cabe definirlo como un efecto producido en la percepción sonora cuando se escuchan 2 sonidos de distinta intensidad al mismo tiempo. Cuando esto

sucede el sonido de menor intensidad resultará inaudible ya que el cerebro sólo procesará el sonido enmascarador.

El sonido de nivel más alto poseerá un efecto de enmascaramiento mayor si el de nivel más suave tiene una frecuencia cercana.

2.6 Interfaces de Audio 3D en discapacitados visuales

Se ha demostrado que la retroalimentación auditiva puede mejorar la experiencia y percepción de profundidad en el usuario. La mayoría del software utiliza sonidos para presentar información que de otra manera sería difícil de presentar. La retroalimentación auditiva resulta particularmente útil cuando se desea concentrar la atención del usuario en lo que está ocurriendo en el programa: se efectuó un experimento en base a un ambiente de programación al cual se le agregó un sonido particular para distintos tipos de rutinas, bloques de código, errores semánticos, etc. Los usuarios fueron capaces de detectar fácilmente ciclos infinitos así como identificar procesos que se ejecutaban en paralelo.

Actualmente existe una tendencia a que las interfaces se diseñen gráficamente y evitando el uso de texto. Esta tendencia ha entorpecido el uso de este tipo de software en personas discapacitadas visualmente quienes, previo a esta tendencia, eran capaces de controlar razonablemente bien la interfaz a través de uso de sintetizadores de voz en base a la lectura de los contenidos en pantalla.

Se ha comprobado que el sonido puede ambientar a usuarios discapacitados visuales en ambientes dos dimensionales a través de estudios y software de prueba [1, 2, 3, 4, 5, 6], y es por ello que resulta interesante conocer cuál es la mejor forma de orientar a los usuarios finales en ambientes que incluyan varias dimensiones y grados de libertad.

3 Requerimientos

3.1 Objetivos

Diseñar, implementar y evaluar aprendizaje (construcción de modelos mentales) de entornos virtuales 3D que permita al usuario orientarse en un ambiente desconocido utilizando como herramienta de construcción el Unreal Engine 2.

3.2 Características de los usuarios

El tema de estudio está orientado a usuarios con discapacidad visual total y parcial cuyas edades oscilen entre los 12 y 18 años de edad.

Adicionalmente se requerirá que los usuarios posean habilidades para desplazarse en forma independiente mediante el uso de bastón y no tengan conocimientos de la estructuración de la estación del metro Universidad de Chile.

3.3 Alcance

La implementación del escenario principal sobre el cual se realizará el estudio se basará en la diagramación simplificada de los dos niveles superiores de la estación del metro “Universidad de Chile”, la cual fue elegida por su geometría simétrica y grandes dimensiones. No se consideraran los locales comerciales que se encuentran en su interior sino que sólo los pasillos y escaleras.

3.4 Requisitos de Hardware/Software

Los requerimientos de hardware establecidos para este documento corresponden a los de la herramienta *Unreal Engine 2*:

- Pentium III o AMD Athlon con un procesador de 1.0 GHz (Se recomienda Pentium® o AMD con un procesador de 1.2GHz o superior)
- 128MB en RAM (Se recomiendan 256 MB en RAM o superior)
- Se requiere que al menos existan 5.5 GB de espacio libre en disco
- Lector CD-ROM de 8x o lector de DVD
- Tarjeta de sonido compatible con Windows®

- Tarjeta de video de 32 MB (Se recomienda una tarjeta *NVIDIA* o *ATI hardware T&L* de 64 MB)
- *DirectX*® versión 9.0b instalado

3.5 Descripción de *Unreal Engine 2*

El Software principal a utilizar para la implementación de los escenarios definidos en este trabajo será el *Unreal Engine 2* que corresponde a un motor para juegos de PC y consolas creado por la compañía *Epic Games* (cuya página web corresponde a <http://www.epicgames.com/>). Este motor nace de una implementación realizada a partir de un juego llamado *Unreal* (creado en el año 1998) para luego convertirse en la base de muchos juegos, en particular juegos pertenecientes al género de ROL y de perspectiva en tercera persona.

Está escrito en el lenguaje de programación C++ a partir de lo cual se han generado versiones para las plataformas PC (*Microsoft Windows*, *GNU/Linux*), *Apple Macintosh* (*Mac OS*, *Mac OS X*) y la mayoría de las consolas (como *DreamCast*, *Xbox*, *Xbox 360*, *Playstation 2*, *Playstation 3*, *Wii*).

Adicionalmente esta herramienta ofrece varias herramientas adicionales de gran ayuda para diseñadores y artistas que por ejemplo permiten modificar diseños de mallas o animaciones.

Existen actualmente 3 versiones de *Unreal Engine* liberadas en los años 1.998, 2.002 y 2.007. La versión con la que se trabajará corresponde a la segunda versión (de ahí su nombre).



Ilustración 5. Comparación gráfica de las 3 versiones del motor *Unreal Engine*

A continuación se muestra una imagen que corresponde al software de edición que se utiliza para el diseño de escenarios e interacciones:

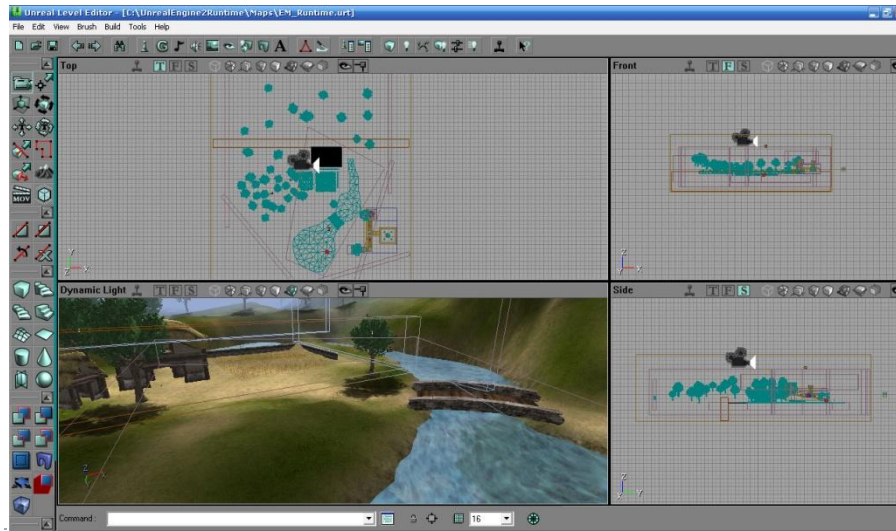


Ilustración 6. Captura de pantalla de la aplicación *Unreal Engine 2* para creación y edición

4 Diseño

4.1 Modelo de Diseño

El modelo de Diseño estará basado en las siguientes etapas:

- **Etapas de Apresto:** un primer acercamiento a la interacción de los usuarios con la herramienta de modo que permita confirmar y evaluar supuestos, y sobre la cual guiar el desarrollo de las etapas posteriores.
- **Etapas de Interacción Simple:** que corresponderá a la interacción de un escenario basado en el nivel superior de la estación de metro Universidad de Chile.
- **Etapas de Interacción por Niveles:** que corresponderá a la interacción de un escenario basado en los niveles superior e intermedio de la estación de metro Universidad de Chile.
- **Etapas de Aplicación:** que corresponderá a la aplicación práctica del modelo mental del escenario con el que previamente se ha interactuado en forma virtual, realizando un recorrido por la estación de metro Universidad de Chile en forma independiente.

4.2 Etapa de Apresto

Para la etapa de apresto se diseñaron 2 escenarios de baja complejidad, en los cuales existían 3 habitaciones conectadas por pasillos. El usuario inicia en una de las tres habitaciones y debe encontrar, a partir de pistas auditivas, los pasillos que lo guiarán a las demás habitaciones.

Se ubicarán sonidos en las entradas de las conexiones de habitaciones con pasillos para que el usuario pueda identificar conexiones (como posibles alternativas en la elección de su camino). Además se incluirán sonidos que le indiquen la dirección hacia donde lo lleva el Pasillo que hagan uso de la capacidad de sonido especializado y que orienten al usuario tanto en profundidad como en dirección.

Los mapas diseñados para esta etapa fueron los siguientes:

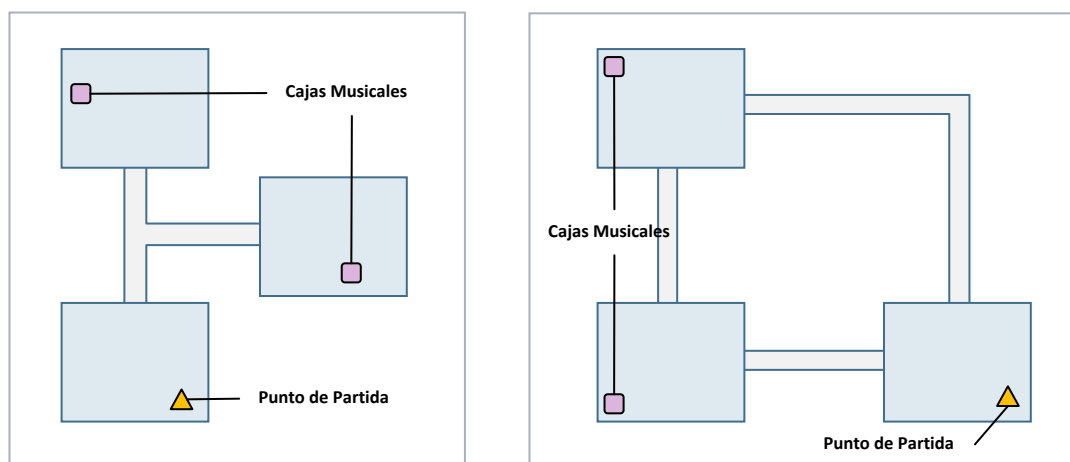


Ilustración 7. Mapas Diseñados para la etapa de Apresto

El primer mapa (el que se muestra a la izquierda de la Ilustración 7) se diseñó para testear el reconocimiento de intersecciones de pasillos donde el usuario debe elegir entre una de las alternativas que tiene para seguir su camino para después devolverse y analizar la alternativa que se dejó de lado en una primera decisión.

El segundo mapa (el que se muestra en el lado derecho de la Ilustración 7) se diseñó para testear la identificación de ciclos en la estructuración de las habitaciones y pasillos.

La motivación bajo la cual el usuario debe interactuar con el mapa hace referencia a buscar 2 cajas que se encuentran en 2 de las 3 habitaciones y que emiten una canción corta al momento de acercarse a ellas.

En el caso del segundo mapa se le solicitará al usuario que una vez identificadas las 2 cajas ubique en una maqueta con la misma estructura del escenario la posición de las cajas a modo de poder establecer el grado de ubicuidad que establece en su modelo mental del escenario a partir de la interacción con el software.

En este segundo mapa se le entregará al usuario retroalimentación auditiva para que encuentre las salidas de las habitaciones y en las intersecciones de pasillos, además de indicarle si se encuentra en un pasillo o en una habitación.

Lo que se pretende evaluar en esta etapa es conocer si los usuarios son capaces de representar mentalmente su recorrido y algunos puntos importantes dentro del escenario, lo que se logra al

pedirles que indiquen estos puntos dentro de una maqueta con la misma estructuración que la del escenario.

4.3 Metro Universidad de Chile – Nivel Superior

El siguiente paso a evaluar con nuestros usuarios es la interacción con un escenario más complejo, para ello se utilizará el escenario del nivel superior del Metro Universidad de Chile.

El escenario se estructura como se muestra a continuación:

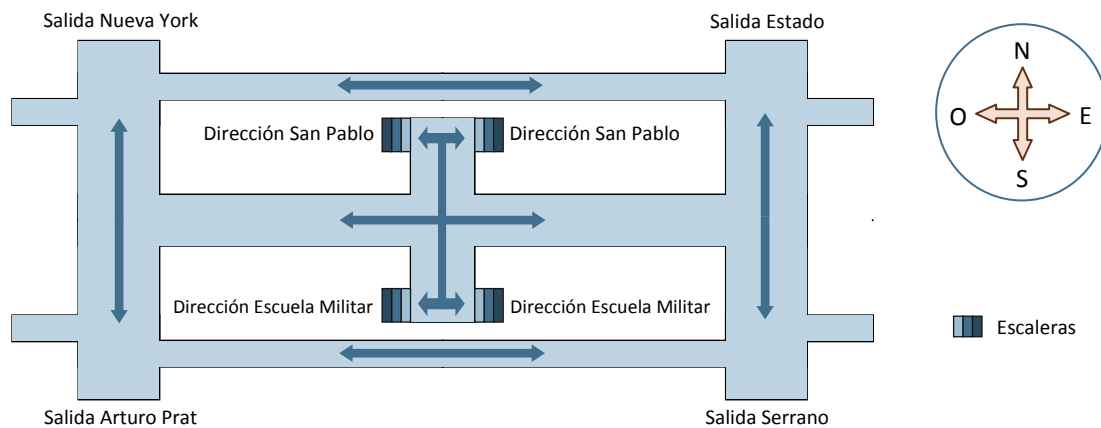


Ilustración 8. Escenario del Nivel Superior de la estación del Metro Universidad de Chile

Este escenario es una versión simplificada del nivel superior de la estación, pues no considera los locales comerciales que existen y sólo se consideran paredes orientadas horizontal y verticalmente como se indica en la figura.

A partir del Pasillo Central surgen 2 alternativas que sirven para acceder al ingreso del metro con dirección a los metros San Pablo y Escuela Militar que permiten acceder al nivel intermedio de la estación descendiendo por alguna de las 4 escaleras que se encuentran simbolizadas en la ilustración.

La retroalimentación auditiva en estos escenarios estará centrada en posicionarlo a través del uso de puntos cardinales y definiendo nombres distintos identificando cada uno de los pasillos de modo de facilitar la ubicuidad del usuario.

En esta etapa se pretende evaluar si el usuario es capaz de construir mentalmente el escenario a partir de la interacción con el software que utiliza retroalimentación auditiva a medida que se

recorre. Para ello se le pedirá que a partir de un origen constante (ubicado en la salida a Arturo Prat, que corresponde a la salida ubicada en el sector suroeste del escenario, y además siempre orientado mirando hacia el norte) se desplace por el escenario llegando a distintos puntos del mapa. Para ello se dejará que inicialmente el usuario interactúe con la aplicación para luego realizar las tareas que permitan evaluar la construcción mental del escenario.

Las tareas estarán enfocadas a cubrir todo el mapa a través de recorridos donde el usuario deberá llegar a una de las cuatro salidas del metro que se encuentran en el escenario a partir del uso de algún pasillo en particular, por ejemplo:

1. El punto de partida se ubica en la salida suroeste del escenario (salida Arturo Prat)
2. El usuario deberá tomar el pasillo central
3. El destino será la salida noreste del escenario (salida Estado)
4. Una vez arribado el destino este será tomado como un nuevo punto de partida
5. El usuario deberá hacer uso del pasillo norte
6. El destino final será la salida noroeste del escenario (salida Nueva York)

Una vez realizada la actividad se le pedirá al usuario que reconstruya mentalmente el recorrido que ha realizado verificando si a partir del uso del software se ha logrado arraigar un modelo mental que represente el escenario y el recorrido que ha utilizado, utilizando la terminología (nombre de pasillos y puntos cardinales) utilizados en la construcción del escenario.

4.4 Metro Universidad de Chile – Nivel Superior e Intermedio

El siguiente paso queda establecido por la ampliación del escenario del Nivel Superior del Metro Universidad de Chile por medio de la inclusión una versión simplificada del Nivel Intermedio a través del cual el usuario podrá recorrerlo identificando principalmente su estructura, la posición de las boleterías y los torniquetes a través de los cuales accedería al andén del metro.

La estructuración del escenario se indica a continuación:

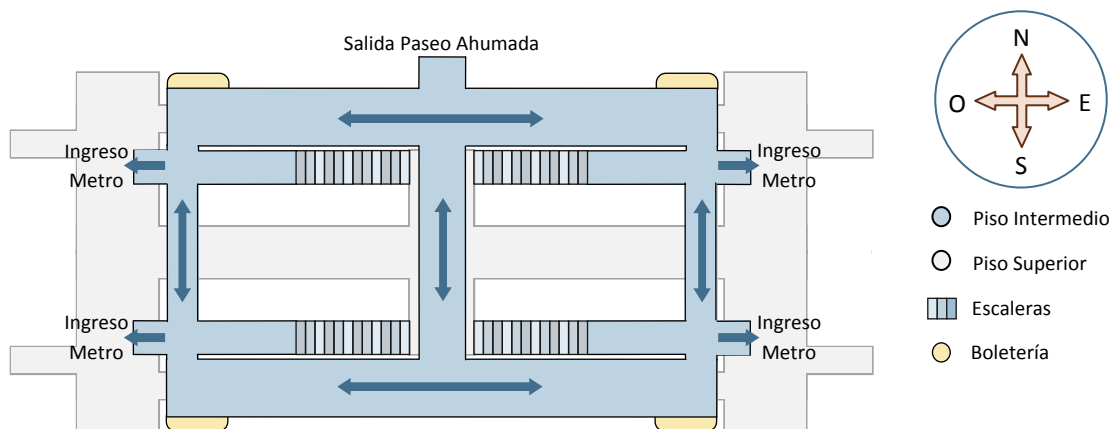


Ilustración 9. Escenario del Nivel Intermedio de la estación del Metro Universidad de Chile

El usuario descenderá a partir del nivel superior haciendo uso de las escaleras desde el piso superior, lo cual implica que la elección de la dirección en la cual desea viajar en el metro (dirección San Pablo o dirección Escuela Militar) estará dada por una elección previa realizada cuando el usuario se encuentra en el Piso Superior y decide tomar algunas de las escaleras del sector norte del Pasillo Central (que correspondería a la dirección San Pablo) o bien al sector sur del Pasillo Central (que correspondería a la dirección Escuela Militar).

La retroalimentación auditiva seguirá la misma línea que en el piso Superior, guiando el posicionamiento del usuario a través de pistas auditivas que hacen uso de referencias a puntos cardinales e identificando los sectores importantes del nivel.

En esta etapa la actividad evaluativa tendrá como intención completar la tarea realizada a partir del uso del escenario anterior, el usuario a partir de alguna de las salidas del nivel superior deberá encontrar las escaleras para descender al nivel intermedio (habiéndose asignado previamente una dirección del metro hacia la cual orientarse) encontrar la boletería más cercana para luego ingresar a la entrada al metro.

4.5 Retroalimentación Auditiva

La retroalimentación estará basada en pistas auditivas que entreguen al usuario información suficiente para que pueda ubicarse espacialmente, entre ellas:

- Identificación de colisión con los límites del escenario.

- Ubicación espacial, es decir, pistas auditivas que orienten al usuario en cuanto a su posición dentro del escenario.
- Descripción de alternativas en caso de llegar a una intersección de pasillos.
- Cercanía a Puntos Importantes.

4.6 Patrones de Sonidos

A partir de las posibilidades analizadas en el *Unreal Engine* y análisis de usabilidad se diseñaron los siguientes patrones de sonido:

4.6.1 Patrón de Sonido Simple

Consiste en generar pistas auditivas gatilladas por la cercanía a ciertos puntos definidos dentro del escenario, por ejemplo, para señalar puntos importantes del escenario.

4.6.2 Patrón de Dirección

Consiste en utilizar elementos móviles dentro del escenario, cuyo movimiento se active cuando el usuario alcance ciertos sectores definidos dentro del escenario, por ejemplo la entrada a un pasillo. El elemento móvil se desplazará en la dirección de alguna alternativa de movimiento importante, por ejemplo en la dirección de un pasillo al momento de acercarse a la entrada. Al mismo tiempo que el elemento se mueve generará un sonido, con la intención de que el usuario pueda percibir dirección y profundidad al hacer uso del motor de sonido.

Estos elementos interactivos se encontrarán y se desplazarán por lugares que no afecten el movimiento del usuario. Una vez realizado el movimiento el elemento volverá a su posición original sin emitir ningún sonido, lo que implicará que el sonido no podrá ser activado en forma inmediata, sino que tendrá un retraso asociado para volver a activarse.

4.6.3 Patrón de Definición de Sectores

Consiste en utilizar pistas auditivas de modo que el usuario identifique el ingreso y salida de ciertos sectores del mapa. La construcción de este tipo de patrón requerirá almacenar en una variable un valor que indique cuando el usuario se encuentre dentro (Activado) o fuera (Desactivado) del sector. El valor Activado será adquirido por la variable al momento en que el usuario ingrese al sector, al mismo tiempo se generará un sonido que indique el ingreso a un sector en particular. En cambio el valor Desactivado será adquirido al momento en que el

usuario ingrese a un nuevo sector (lo que indicará que se ha abandonado el sector original), junto con ello se activará la condición en ese nuevo sector y por lo tanto la pista auditiva de ingreso le indicará el abandono del sector inicial y el ingreso al nuevo sector.

4.6.4 Patrón de Bloqueo de Sectores

En algunos casos se requerirá que existan bloqueos en el escenario que impidan que el usuario se desplace dentro de algunos sectores definidos, hasta que se hayan cumplido ciertas condiciones. Por ejemplo, en el escenario que involucra el nivel intermedio de la estación de metro se utilizará este tipo de patrón para bloquear el acceso al metro hasta que el usuario haya accedido a la Boletería.

La forma en que se implementará este patrón será muy similar al Patrón de dirección, la diferencia radicará en que el elemento se encontrará bloqueando el acceso a algún sector en particular y no regresará a su posición inicial una vez que fue activado, dejando desbloqueado el acceso.

5 Etapa de Aplicación de Conocimiento

Una vez que los usuarios han tenido la suficiente interacción con el software el siguiente paso está orientado a evaluar si el modelo mental que han generado de la estación del metro es suficiente como para que los usuarios se valgan por sus propios medios en el escenario real.

Para ello se llevará a los usuarios al metro Universidad de Chile, se les ubicará en cerca de una de las salidas del nivel superior del metro y se le pedirá que efectúe la misma tarea que realizó en la etapa anterior del proceso en la cual debe descender al nivel intermedio por las escaleras norte o sur (dependiendo de la dirección en la cual desee tomar el metro), ubicar la boletería más cercana y luego identificar el lugar donde se encuentran los torniquetes que dan acceso a los andenes del metro.

Con esto se conseguirá evaluar el nivel de fidelidad que ha generado el software en el modelo mental del usuario y si la herramienta es efectiva en el aprendizaje de estructuras complejas como lo son escenarios de varios niveles, en particular los dos niveles superiores de la estación del metro Universidad de Chile.

6 Implementación

6.1 Introducción al *Unreal Engine*

Unreal Engine es un sistema que administra escenarios, texturas, mallas estáticas y otros aspectos artísticos de juegos, combinándolos en forma fluida dentro de una obra interactiva.

Unreal Engine está formado principalmente por 3 motores: motor gráfico, motor de sonido y motor físico, donde cada uno de ellos funciona en forma independiente a los otros 2. Estos 3 motores o sistemas independientes se vinculan para dar origen a un único motor, permitiéndoles funcionar juntos en forma sincronizada.

En la figura se muestra un diagrama que representa la forma en que las componentes principales se relacionan en su funcionamiento:

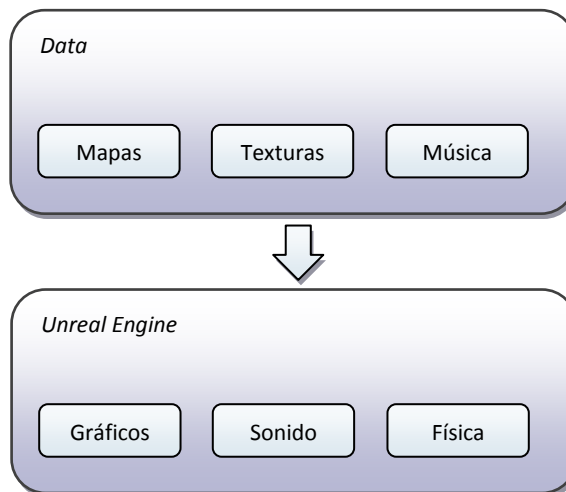


Ilustración 10. Flujo e Interacción de Componentes del *Unreal Engine*

6.1.1 Motor Gráfico

El motor gráfico del *Unreal Engine* es el encargado de desplegar en el dispositivo de video (una pantalla o un televisor) lo que está ocurriendo con el juego.

Este motor es el encargado de tomar toda la información relativa a los entes involucrados dentro del sistema (como el escenario y los jugadores) y traducirlos en tiempo real a un elemento reconocible para finalmente ser observado en la pantalla.

6.1.2 Motor de Sonido

El motor de sonido es el que hace posible manipular distintos niveles de sonidos con una gran calidad. Recrea efectos de sonido y sonidos ambientales en forma tan precisa como sea posible, haciendo uso del hardware de sonido que se encuentre instalado.

Permite configurar opciones de lugar y tiempo en que estos sonidos deben ser ejecutados a través de la configuración de ciertas propiedades.

6.1.3 Motor Físico

Una parte importante al representar ambientes y escenarios que sean fieles a la realidad es simular las mismas leyes que rigen la física del mundo real, como por ejemplo el rebote y giro de un objeto que golpea una pared.

El Motor Físico se encarga de simular una física definida a partir de una serie de reglas establecidas. A este motor físico también se le suele denominar “*Karma*” o incluso a veces “Motor Matemático” por la definición de reglas y funciones matemáticas que modelan la realidad.

6.1.4 Interacción de Componentes

Al igual que la mayoría de los núcleos de los juegos, el Unreal Engine utiliza lo que se denomina un “Ciclo de Juego”.

El Ciclo de Juego es una parte del programa que se repite constantemente a través del juego de modo de mantener la lógica y que todas las componentes sean actualizadas correctamente. Es dentro de este ciclo donde se chequean todas las propiedades de los usuarios, como por ejemplo si ha realizado algún movimiento en alguna dirección.

Adicionalmente este Ciclo de Juego se encarga de redibujar la pantalla (para mantener la fluidez del juego), reproducir sonidos e incluso enviar y recibir información a través de la red.

Además de utilizar un Ciclo de Juego, *Unreal Engine* es capaz de manejar un sistema guiado por eventos. Esto significa que el sistema administra una cola que mantiene una serie de eventos

donde las distintas componentes pueden registrar acciones, como cuando el usuario presiona una tecla en particular, o cuando un usuario se comunica con otro a través de la red. Estos eventos son administrados por medio de prioridades, lo que significa que eventos con alta prioridad serán procesados antes que otros eventos con menor prioridad.

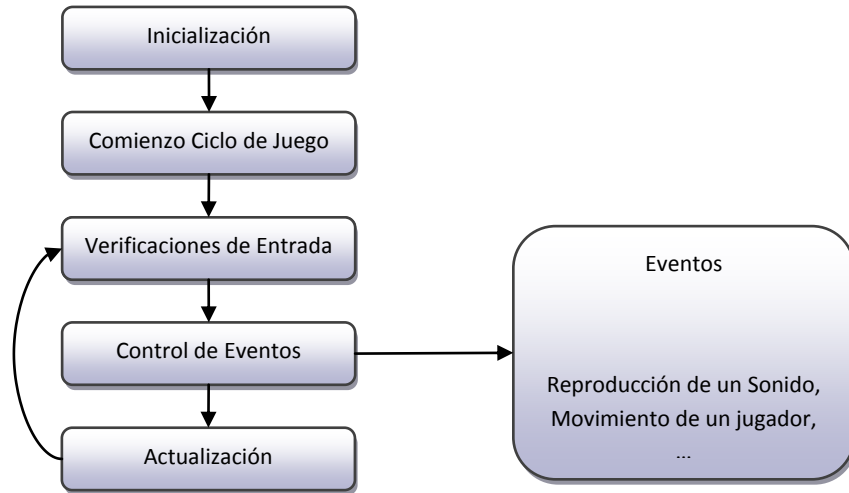


Ilustración 11. Ciclo de Juego del *Unreal Engine* con manejo de eventos

Los eventos representan actividades críticas que deben ser ejecutadas para no afectar la fluidez, y por lo que el *Unreal Engine* se encargará de realizar estas tareas en forma previa a la actualización de valores.

6.1.5 Recursos

La mayor parte de la información que *Unreal Engine* utiliza se encuentra almacenada en una serie de paquetes. Un paquete puede ser pensado como una colección de información relacionada, tal como información de texturas, mallas estáticas, animaciones, sonidos, etc.

Se debe tener en mente que no todos los recursos que se utilizan son almacenados en paquetes, por ejemplo los escenarios son almacenados dentro de mapas.

A continuación se muestra una ilustración que combina los distintos tipos de recursos con las extensiones de archivos en las que se almacena la información:

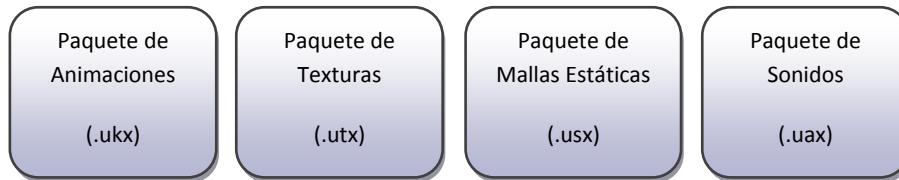


Ilustración 12. Organización de paquetes junto con la extensión de archivos correspondiente

A continuación se describen los recursos más relevantes involucrados, se omitirá principalmente lo que tenga que ver con texturas y animaciones, ya que para este estudio en particular no se hace necesario ahondar en tales temas.

Mapas o Niveles

Los mapas o niveles son uno de los recursos que utiliza el *Unreal Engine* que no es almacenado en paquetes. En ellos se construye el ambiente en el cual se ejecutará el juego. Un mapa es el encargado de combinar los distintos recursos involucrados en el *Unreal Engine*, como las animaciones, las texturas, las mallas estáticas y los sonidos. El mapa no inserta directamente los recursos almacenados en los paquetes sino que registra referencias hacia los paquetes, por lo cual, si un paquete de sonidos que es utilizado en el mapa es borrado, al intentar reproducir los sonidos no se obtendrá ningún resultado.

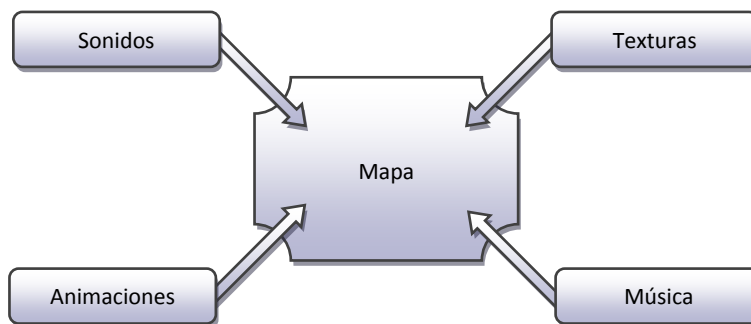


Ilustración 13. Recursos Utilizados en un Mapa

Sonidos

Al igual que las texturas, los sonidos deben ser creados en aplicaciones externas o puede provenir desde archivos de audio que se encuentren previamente disponibles. Los sonidos en el mapa son reproducidos en respuesta a eventos específicos, como al abrir una puerta o al pasar por una cierta región del escenario. Algunos sonidos, como los sonidos ambientales son

reproducidos cuando el usuario se encuentra dentro de un cierto rango. Los sonidos se encontrarán almacenados en paquetes con extensión “.**uax**”.

Mallas Estáticas

Las mallas estáticas (en inglés “*static meshes*”) consisten en modelos poligonales altamente optimizados que son importados al juego a partir de paquetes 3D externos, como *Maya* y *3D Studio Max*.

Esto permite la creación de numerosas instancias a partir de una única malla estática poligonal, con (virtualmente) una sobrecarga igual (pérdida en la tasa de *frames*) a que si existiera una única copia dentro del mapa. Por ejemplo, un mapa que posea 50 copias generadas a partir de una misma malla estática funcionará igual de rápido que un mapa que posea una única copia.

Las mallas estáticas son almacenadas en paquetes con extensión “.**usx**”.

6.1.6 UnrealEd

UnrealEd corresponde al sistema principal que permite crear y controlar todo el contenido de un proyecto. Es capaz de crear mapas para un proyecto, importar contenidos y administrar los paquetes de recursos. Además es el único programa que permite abrir y acceder los contenidos de un paquete *Unreal*.

No puede ser utilizado para crear contenidos 3D como mallas estáticas, modelos de personajes, crear texturas propias o grabar efectos de sonido. Por esta razón se hace necesario incorporar el uso de otro tipo de aplicaciones externas que cubran tales necesidades.

6.2 *UnrealEd*: Funciones importantes

En el subcapítulo anterior (exactamente en el punto 6.1.6) se dio a conocer la definición y aplicaciones principales que se pueden realizar con el *UnrealEd*.

UnrealEd fue introducido originalmente en el año 1998, y fue el primer editor que les entregaba a los usuarios retroalimentación en tiempo real, con lo que los diseñadores eran capaces de ver en forma exacta e instantánea cómo se verían sus mapas al momento de ser jugados sin necesidad de dar inicio al juego (proceso que típicamente toma un tiempo relevante).

A continuación se listan las características más rescatables en la construcción de los mapas a utilizar en este estudio en particular, ellos son:

- Introducción al *UnrealEd*
- Componentes del *UnrealEd*
- Creación de Mapas mediante Métodos Sustractivos
- *Movers*: Elementos Interactivos
- *Triggers*: Gatilladores de Eventos

Todas estas características se explican a continuación.

6.2.1 Introducción al *UnrealEd*

Previo al uso de del *UnrealEd* se hace necesario ahondar en algunos conceptos sobre los cuales se basa su utilización:

Actores

Luego de que la geometría del mapa es establecida, cualquier otro objeto ubicado en el nivel se considera un Actor. Los gatilladores, jugadores, las luces o cualquier elemeto que no pertenezca a la geometría corresponderá a un actor.

Los actores son clasificados en 2 subcategorías: “*placeable*” y “*non-placeable*”

Actores “Placeable” versus Actores “Non-Placeable”

Muchos actores son clasificados como “*Placeable*” tales como las posiciones de inicio de los jugadores, luces y “*Movers*”, lo que implica que este tipo de actores tendrán asociada una posición en particular dentro del mapa. Estos actores deberán ser incluidos en forma manual

por el diseñador del mapa además de configurar sus propiedades para controlar su comportamiento y lograr que ejecuten las acciones deseadas.

En el caso de los actores “*Placeable*” cada tipo de actor tendrá una representación visual propia, por ejemplo las luces tendrán el ícono una ampolleta y el punto de partida de los jugadores tendrán el ícono de un joystick como representación.

Por otro lado, los objetos “*Non-Placeable*” no necesitan ser ubicados manualmente dentro del nivel durante su creación. Un ejemplo de ello son los tipos de niveles que se pueden definir dentro del *Unreal Engine* que controlan las reglas del juego.

6.2.2 Componentes del *UnrealEd*

Al ejecutar la aplicación *UnrealEd* nos encontraremos con una pantalla como la que se muestra a continuación:

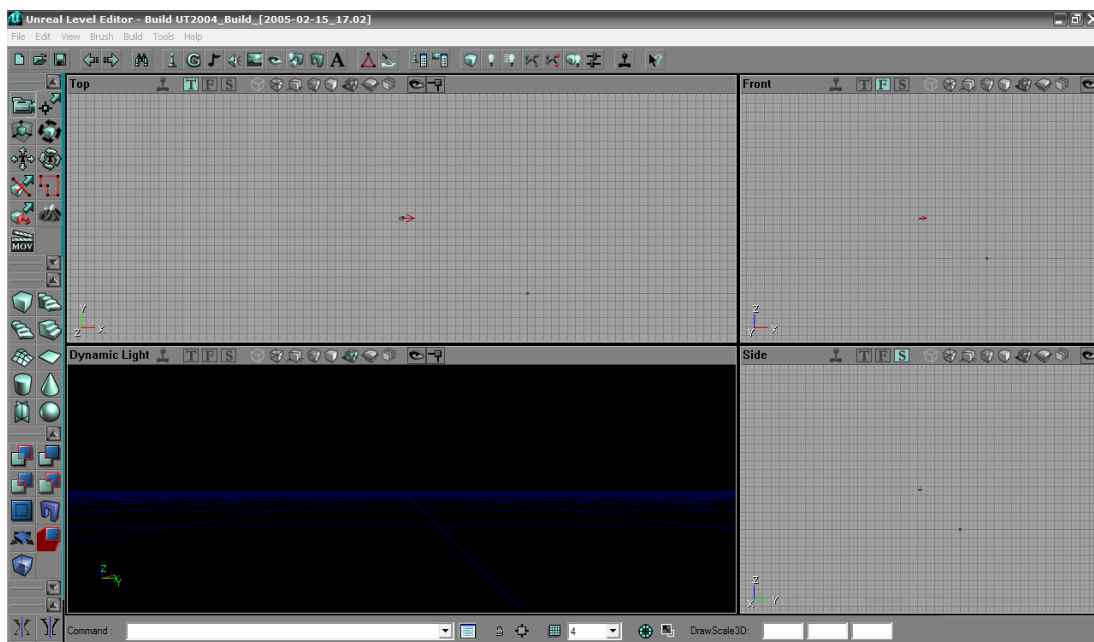


Ilustración 14. Interfaz de Usuario del UnrealEd

A partir de la ilustración se pueden identificar las siguientes componentes:

Barra de Menú Principal

Se encuentra localizada en la parte superior de la ventana, provee en forma casi completa toda la funcionalidad que se encuentra accesible en otros lugares de la interfaz ubicada en un menú de estructura de formato desplegable (*drop-down*).

Contiene rangos de funcionalidad que abarcan los procesos que permiten abrir, guardar e importar archivos, ajustar las vistas, controlar el proceso de reconstrucción y mucho más.

Barra de Herramientas

Se encuentra localizada justo debajo de la Barra de Menú Principal e incluye una serie de íconos que corresponden a tareas de uso común en el desarrollo de la construcción del mapa. Estas tareas incluyen abrir y guardar archivos, abrir navegadores, reconstruir ciertos aspectos del nivel y ejecutar el escenario *Unreal* de modo de poder testear el nivel implementado hasta el momento.

Caja de Herramientas

Corresponde a una barra de herramientas vertical ubicada en el lado izquierdo de la interfaz. Entrega acceso a una amplia variedad de funciones que son usadas en forma frecuente en la construcción del mapa. Estas funciones incluyen ajustes de cámaras, agregar o sustraer masas, agregar Actores especiales y mucho más.

Vistas

Las Vistas corresponden al aspecto más crucial del *UnrealEd* que provee de “ventanas” hacia el nivel que se está diseñando. Las Vistas son provistas a partir de 2 distintos sabores: con perspectiva y ortográficas. Las Vistas con perspectiva proveen una sensación de profundidad en 3 dimensiones. Las Vistas ortográficas son bidimensionales, tal como un set de planos.

En el caso de la Vista con perspectiva se puede alternar entre una variedad de modos como *Wireframe* (Modo Alambre), Solo Texturas e Iluminación Dinámica; vistas basadas en la parte del nivel en la cual se encuentre enfocada la atención en el momento.

Barra de Consola

Se encuentra en la parte inferior de la ventana y ofrece acceso a características propias de una línea de comandos, que puede ser usada para ingresar distintos comandos al *Unreal Engine*. Adicionalmente se puede utilizar la Barra de Consola para escalar actores dentro del mapa.

Navegadores

Distintos navegadores se encuentran disponibles en el *UnrealEd* para acceder a distintos recursos que se necesiten en la implementación del mapa. Cada recurso posee un navegador propio: Texturas, Actores, Mallas, Animaciones, Mallas Estáticas, Sonido, etc.

Ventana de Propiedades

En el *UnrealEd* se encuentran disponibles variados tipos de Ventanas de Propiedades que son utilizadas para controlar los distintos parámetros de los objetos.

Dentro de estas Ventanas de Propiedades la más importante y más utilizada a través de este estudio es la Ventana de Propiedades de Actores.

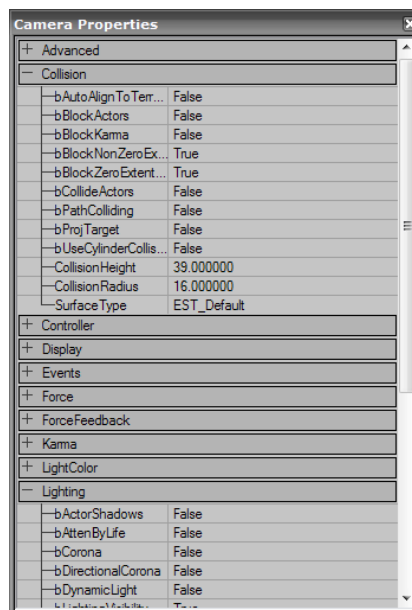


Ilustración 15. Ventana de Propiedades del Actor Camera

Por ejemplo (como se muestra en la ilustración) la ventana de propiedades del Actor *Camera* despliega diferentes listas de propiedades. En caso de un actor distinto los tipos de propiedades pueden variar con respecto a la ilustración.

Cada propiedad se encuentra organizada en secciones desplegables de modo que no se haga necesario efectuar “*scroll*” a través de las distintas páginas de propiedades.

6.2.3 Creación de Mapas mediante Métodos Sustractivos

Antes de la existencia de *UnrealEd* los mapas debían ser creados de la misma manera en que las estructuras arquitectónicas son creadas en el mundo real: creando un piso, luego agregándole paredes y finalmente un techo. Incluso una simple habitación cúbica requería de crear y ubicar no menos de 6 objetos poligonales. Uno podría pensar en que este método mencionado utiliza una forma aditiva de crear áreas cerradas. Esto significa que estos objetos son agregados a un mundo vacío de modo de finalmente estructurar un mapa.

El método que utiliza *UnrealEd* para generar mapas corresponde a un modelo totalmente opuesto al mencionado en el párrafo anterior. Los mapas *Unreal* son creados a partir de un mundo que está formado por una gran masa sólida, por lo cual el método de creación será sustractivo en vez de aditivo. Esencialmente, toda creación de niveles *Unreal* comienza con la sustracción de un volumen desde una masa sólida que compone al mundo, algo parecido a crear habitaciones cúbicas en el centro de una montaña.

Este método posee dos ventajas principales por sobre el acercamiento aditivo:

- La velocidad de creación incrementa en forma dramática. En la metodología aditiva se necesitaba la creación de 6 objetos separados para poder completar una simple habitación cúbica y con la preocupación de que estos objetos quedaran perfectamente alineados. En la versión sustractiva se podrá crear la misma habitación cúbica en un único paso a través de la sustracción de un cubo desde el mundo real, sin tener que preocuparse por el alineamiento de los bordes.
- La segunda ventaja es la reducción de fallas de alineamiento entre murallas, lo que en el método aditivo generaba un efecto determinado HOM (*Hall Of Mirrors*) en donde a través de pequeños espacios el jugador era capaz de ver la parte externa a las estructuras, lo que resultaba en un error gráfico poco natural y realista.

El método sustractivo no significará que no se podrán agregar detalles al mapa en forma aditiva, sino que el punto de partida debe ser de tipo sustractivo, para luego poder aplicar métodos aditivos sobre los vacíos formados.

Escalas Estéticas

Un factor importante a considerar en el diseño de niveles es cómo determinar el tamaño de los elementos, pues dentro del *UnrealEd* no existen nociones de distancia y tamaño como las que utilizamos habitualmente, sino que se utiliza un estándar de medida propio, que corresponde a las Unidades *Unreal* ("*Unreal Units*").

Sin embargo existe una fórmula que puede guiar el escalamiento de objetos bajo la cual se pueden traducir distintos proyectos. La mayoría de los personajes del *Unreal* son modelados a partir de una altura en común (ver ilustración).



Ilustración 16. Dimensiones estándar de los personajes creados en el UnrealEd

Dado este parámetro uno puede establecer un cierto rango de flexibilidad, pero en general uno puede asumir esta altura como una altura estándar.

Se puede asumir que si los personajes poseen una altura de 96 Unidades *Unreal* y en general una persona mide 1,70 mts se puede asumir que:

$$1 \text{ metro} = 96/1,70 = 56 \text{ Unidades Unreal}$$

Por lo tanto si uno desea construir una habitación que tenga 3 metros de alto, bastará con darle una altura de 168 Unidades *Unreal* de alto. Por lo tanto efectuar conversiones desde el mundo real hacia un escenario o mapa de modo de crear estructuras que tengan las mismas dimensiones que en la realidad es un proceso meramente mecánico una vez obtenidas las medidas reales.

Posicionamiento y Escalamiento de Actores

Para explicar la aplicación de estas acciones se describe un pequeño tutorial:

1. En la Caja de Herramientas se debe hacer clic en el ícono del Cubo (📦). Si se desea darle una dimensión en particular al cubo se puede efectuar un clic derecho sobre el ícono, lo que provocará que aparezca un diálogo como el que se indica en la ilustración.

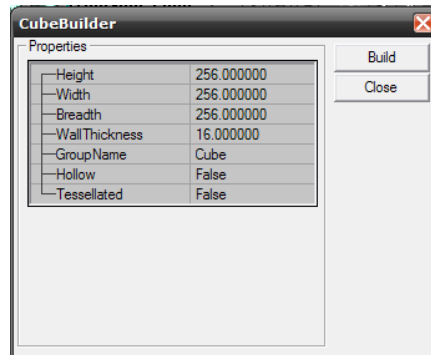



Ilustración 17. Modificación de propiedades de creación de un Cubo

2. Una vez que se ha creado el Cubo y se ha seleccionado en la vista superior (Top) para desplazarlo a través del mapa se debe presionar la tecla Shift junto con el BIM (Botón Izquierdo del Mouse) mientras que desplazamos el mouse hasta que logremos ubicarlo en la posición deseada.
3. Para escalar el Cubo debemos utilizar la opción “Actor Scaling” () que se encuentra en la caja de herramientas. Una vez seleccionado el Cubo y haciendo uso de una combinación de la tecla Control junto con el BIM mientras que se desplaza el mouse, se logrará escalar el Cubo según el eje X del mapa.
4. Si se desea escalar según el eje Y se debe utilizar el BDM (Botón Derecho del Mouse) en vez del BIM. Si se presionan ambos botones del mouse en forma simultánea se podrá escalar en ambas direcciones.
5. Si se desea posicionar o escalar el Cubo en relación al eje Z se deberá realizar el mismo proceso pero en alguna de las vistas *Front* o *Side*. En la vista Dinámica (que es la que despliega una versión tridimensional y a tiempo real) también se pueden efectuar este mismo tipo de acciones con la desventaja de que no es posible realizar movimientos o escalamientos con gran precisión.

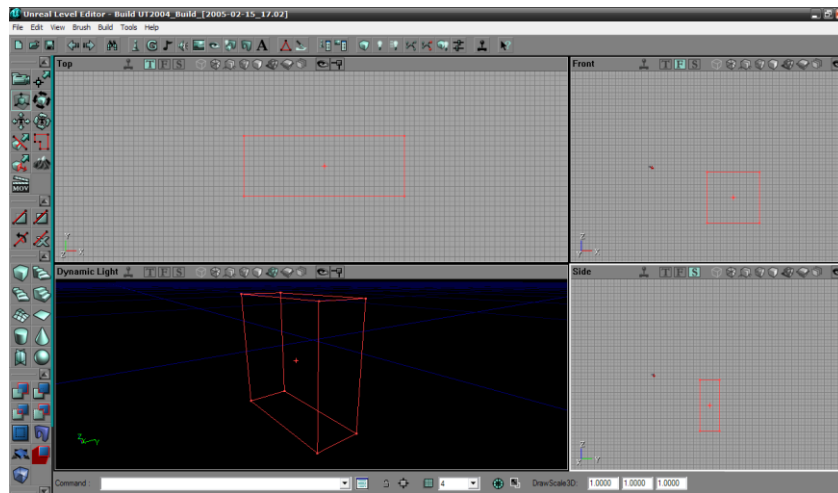


Ilustración 18. Resultados al posicionar y escalar un cubo


Para desplazarse en medio de las vistas se debe mantener presionado cualquier botón del mouse y al mover el cursor se conseguirá el mismo efecto en la vista. De esta forma podremos desplazar el mapa en las direcciones de los ejes X e Y (manteniendo constantes los valores según el eje Z) en el caso de la vista superior (*Top*), en las direcciones X y Z (manteniendo constantes los valores según el eje Y) en el caso de la vista frontal (*Front*) y en las direcciones Y y Z (manteniendo constantes los valores según el eje X) en el caso de la vista lateral (*Side*). Para efectuar desplazamientos en la dirección no considerada se debe efectuar los mismos movimientos con el mouse pero apretando ambos botones en forma simultánea.

El desplazamiento en la vista con perspectiva difiere del movimiento de las vistas que sólo manejan 2 dimensiones:

- Al presionar el BIM y desplazar el mouse logramos desplazamientos en los ejes X e Y del escenario por medio de giros en torno al eje Z en combinación con avances y retrocesos.
- Al presionar el BDM y desplazar el mouse se logra girar la visión de la cámara en torno a los ejes X, Y y Z.
- Al presionar ambos botones y desplazar el mouse lograremos desplazamientos en el eje Z, manteniendo constantes los valores en los ejes X e Y.

Construcción de una primera Habitación

Los pasos para la construcción de una primera habitación serían los siguientes:

1. Primero se debe crear un Cubo, dimensionarlo y ubicarlo en el lugar deseado según las indicaciones del punto anterior.
2. Seleccionar una textura con la cual se creará cada pared interior del Cubo. Como la finalidad de este trabajo no está orientado al trabajo con texturas se utilizará siempre la misma textura de trabajo que representa una pared. Para acceder al Navegador de Texturas se debe hacer clic en el ícono del Navegador de Texturas () que se encuentra en la barra de herramientas. La textura a usar se muestra en la siguiente ilustración:

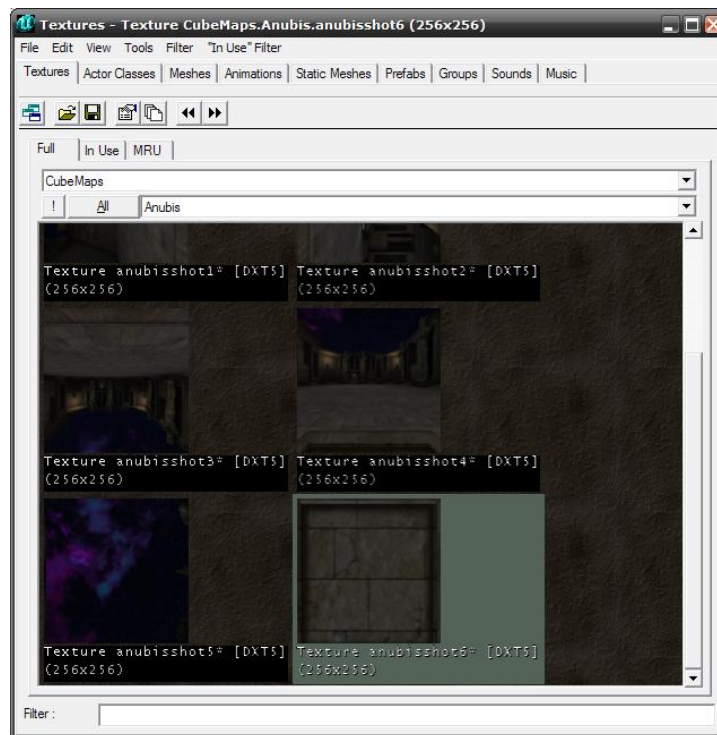



Ilustración 19. Navegador de Texturas

Una vez seleccionada la textura no es necesario mantener el Navegador de Texturas abierto, puede ser cerrado y seguir con el paso siguiente).

3. Una vez seleccionada la textura y con el cubo debidamente escalado y posicionado se debe hacer clic en el botón de sustracción () que se encuentra en la caja de herramientas. Con ello se acaba de crear el primer espacio sustractivo dentro del mapa. A continuación se muestra una ilustración con el resultado de la sustracción de un primer espacio dentro del mapa:

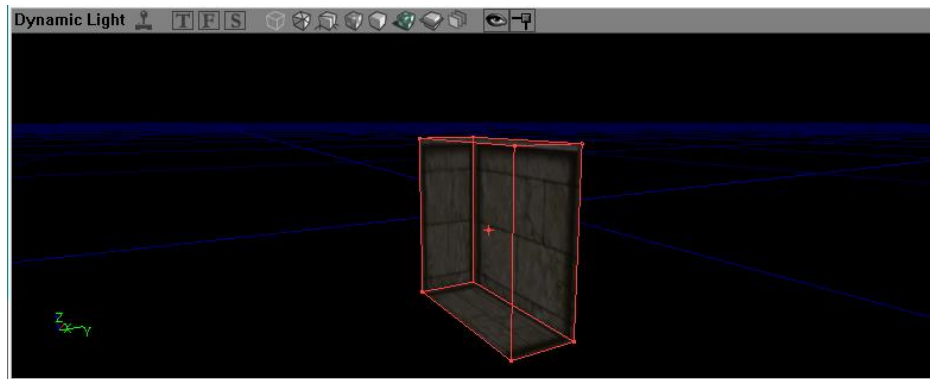


Ilustración 20. Creación de un Cubo mediante sustracción

4. El siguiente paso es agregarle iluminación al cuarto lo cual se realiza a través de establecer iluminación general a todo el escenario. Para se debe acceder a la opción "View" de la Barra de Menú Principal e ingresar a la opción "Level Properties". Se desplegará una Ventana de Propiedades en las cuales podemos modificar los atributos generales del mapa o nivel. Ingresamos al menú desplegable "ZoneLigth" y modificamos los valores del atributo "AmbientBrightness" de 0 a 255 (que corresponde al valor máximo).
5. A continuación se debe establecer un lugar de partida para que el sistema ubique al jugador al momento de iniciar el juego. Para ello se debe (a través de la vista con perspectiva) seleccionar la superficie del piso y apretar BDM y seleccionar la opción "Add Player Start Here" desde el menú que se despliega. Aparecerá el ícono de un joystick en el lugar donde se haya efectuado esta operación (🎮).
6. Para ejecutar el mapa primero debemos construir el escenario a través de la acción "Build All" que se encuentra en la barra de herramientas en el ícono 🏗️, para luego ejecutar el escenario utilizando la opción "Play Map!" que se encuentra en la misma barra de herramientas (🎮). A continuación se muestra una ilustración con el resultado de esta operación:

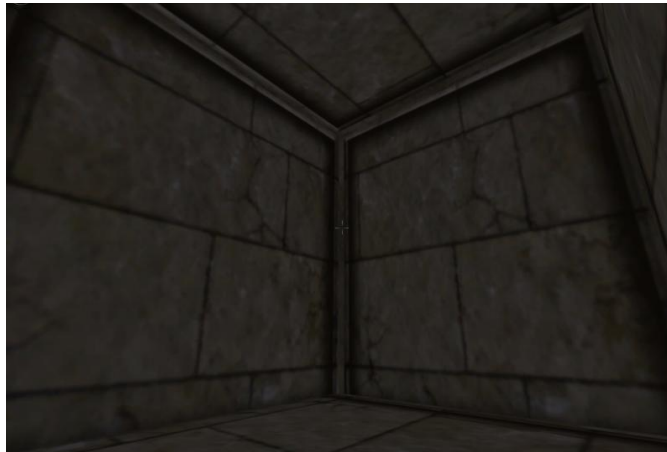


Ilustración 21. Ejecución del Nivel

7. Para salir de la Ejecución del Nivel se debe apretar “*Escape*”, hacer clic en el botón “*Exit Game*” y confirmar la acción.

Bastará con intersectar varias estructuras sustractivas para formar mapas más complejos con combinaciones de habitaciones y pasillos, por lo tanto este conocimiento es suficiente para construir los mapas que han sido diseñados.

6.2.4 *Movers*: Elementos Interactivos

En *Unreal* un *Mover* es un tipo de malla estática animada que se ubica dentro del mapa. Son utilizados para crear una serie de elementos prácticos como puertas y ascensores, o cualquier elemento sólido en el escenario que requiera movimiento.

El movimiento de un *Mover* puede ser activado a partir de una variedad de alternativas. Cada *Mover* posee 8 “*keys*” que representan grabaciones de la posición y la rotación del objeto durante diferentes etapas de la animación.

Por ejemplo se puede registrar en la primera “*key*” de un *Mover* que represente una puerta la posición y rotación adecuadas en que la puerta se encuentre cerrada, para luego agregar una segunda “*key*” que almacene la posición y rotación en el momento que la puerta se encuentre abierta. Con esto, cuando el *Mover* la puerta efectivamente se abrirá.

Una vez que un *Mover* es activado se moverá suavemente a través de los “*key*” que hayan sido definidos, y no se detendrá hasta llegar al último. Una de las alternativas que existen para

activar un *Mover* puede ser gatillada mediante la colisión de un jugador con él. El *Mover* puede ser activado de forma avanzada a partir de un sistema gatillador.

Los pasos para construir un *Mover* son los siguientes:

1. Selección de una Malla Estática a partir del Navegador de Mallas Estáticas. En la Ilustración se muestra la elección de una Malla Estática con forma de cubo, lo cual lo hace fácilmente adaptable mediante posicionamientos y escalamientos como los vistos en el capítulo de Creación de Mapas mediante Métodos Sustractivos (6.2.3). A continuación se muestra una ilustración con la selección de una Malla Estática a partir del navegador de Mallas Estáticas:

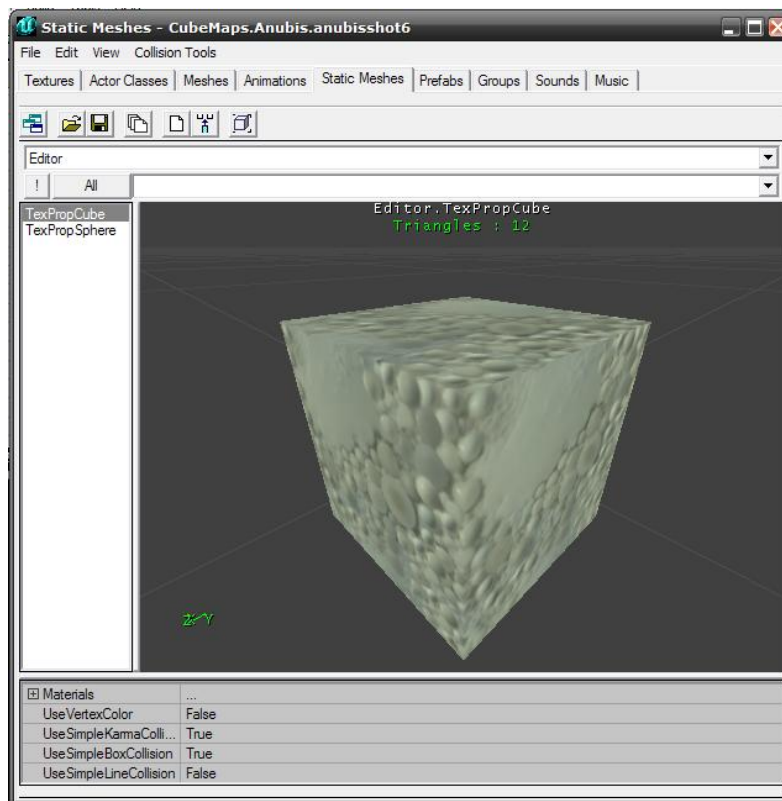


Ilustración 22. Navegador de Mallas Estáticas (*Static Meshes*)

2. Una vez seleccionada la Malla Estática se debe agregar al mapa a través de la acción “Add Mover” que se encuentra en la Caja de Herramientas (). Una vez hecho esto aparecerá dentro de las vistas una malla de color morado que indica su característica de Mover por sobre las Mallas Estáticas Simples que serán de color celeste.

3. Lo siguiente es determinar las distintas “keys” que compondrán la animación y movimiento. La posición en la que se encuentra establecido corresponderá al *key 0*, para agregar nuevos *keys* será necesario utilizar el BDM sobre el *Mover* y cambiar el número de *key* dentro del submenú “*Mover*” para luego desplazar el objeto de modo que ocupe la posición correspondiente.
4. Una vez que hayan sido asignados correctamente todos los “keys” y se elija nuevamente “*Key 0 (Base)*” el *Mover* se deberá desplazar a su posición original, lo que implicará que la información fue almacenada en forma correcta.
5. A continuación se deben configurar las propiedades del *Mover* a través de su correspondiente Ventana de Propiedades dependiendo de la configuración que se desee:
 - Dentro de la Categoría “*Object*” la propiedad “*InitialState*” se puede configurar con varios valores que vienen listados, dentro de ellos destacan:
 - *BumpOpenTime*, lo que provoca que el movimiento es generado cuando un jugador colisiona con la estructura del *Mover*, se mantiene en la posición del *key* final por un cierto lapso de tiempo, y luego vuelve a su posición original.
 - *TriggerToggle*, lo que provoca que el movimiento sea iniciado por medio de un Gatillador, y donde el movimiento del *Mover* es realizado por partes, es decir, cada vez que es gatillado avanza hasta el *key* final y se mantiene en ese lugar hasta que es gatillado nuevamente donde comienza su desplazamiento hasta el *key* inicial.
 - Dentro de la Categoría “*Mover*”, se debe configurar la cantidad de *keys* que se están utilizando en la propiedad “*NumKeys*”.
 - Adicionalmente en la misma Categoría “*Mover*” se puede configurar la propiedad “*StayOpenTime*” con la cantidad de tiempo en segundos que se desea que el *Mover* se mantenga en su posición final antes de devolverse, en el caso de que su “*InitialState*” haya sido configurado como “*BumpOpenTime*”.
 - También se podrá configurar la propiedad “*DelayTime*” que corresponde al tiempo en segundos que espera el *Mover* para iniciar su movimiento.
6. Finalmente se debe reconstruir el mapa y ejecutarlo para probar que todo se encuentre funcionando bien.

6.2.5 Triggers: Iniciadores de Eventos

Los Gatilladores (o *Triggers*) corresponden a Actores que pueden ser ubicados en cualquier lugar dentro del mapa. Cuando son activados dan origen a un evento que es identificado por su nombre. Cualquier Actor que haya sido programado para responder ante ese Gatillador realizará alguna acción preestablecida.

Existen variados tipos de Gatilladores, que pueden ser activados bajo distintas acciones, por ejemplo algunos se activarán cuando un jugador se encuentre a una cierta distancia, otros a partir de *scripts*, etc.

A continuación se describen algunos de los Gatilladores más importantes y que fueron utilizados dentro de la implementación de los escenarios.

Trigger

El “*Trigger*” corresponde a un gatillador que se activa una vez que un jugador se acerca a una cierta distancia de su ubicación dentro del escenario.

Para crear un nuevo “*Trigger*” dentro del escenario se debe acceder al Navegador de Actores y dentro de la opción “*Triggers*” seleccionarlo, para luego hacer clic con el BDM sobre alguna superficie del escenario (como el suelo o una pared) y seleccionar la opción “*Add Trigger here*”.

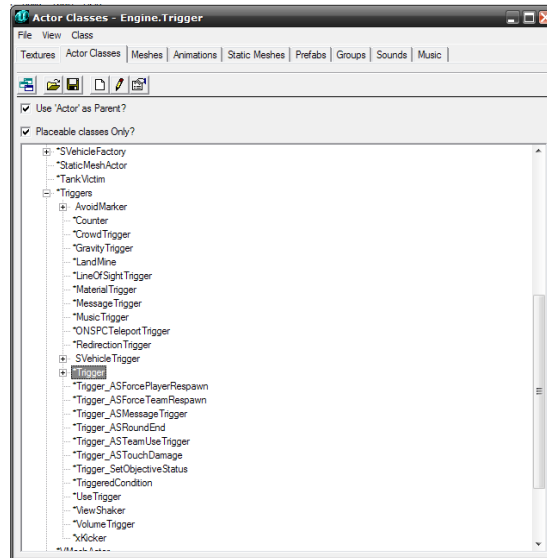


Ilustración 23. Selección de *Trigger* a través del Navegador de Actores

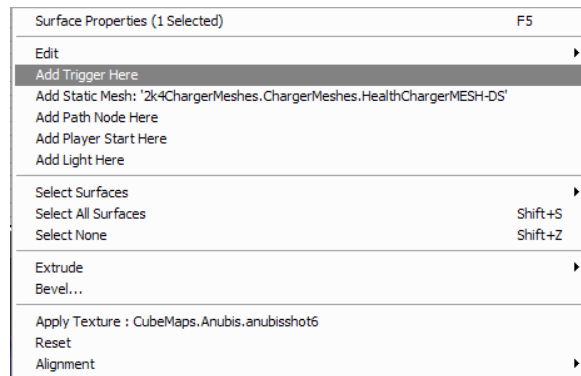


Ilustración 24. Agregar un *Trigger* en la superficie seleccionada

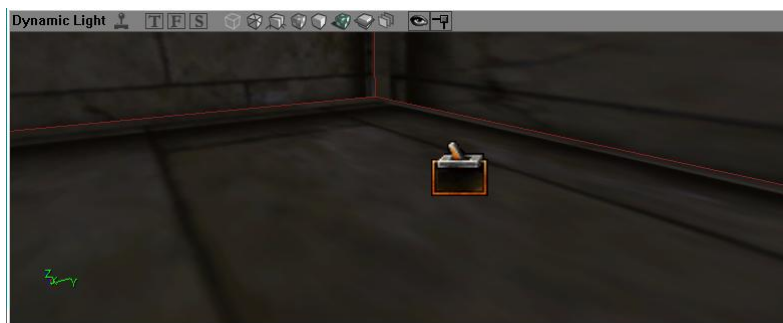


Ilustración 25. Ícono que representa un *Trigger*

Dentro de los atributos más destacables que pueden ser configurados en un *Trigger* a través del Navegador de Atributos (BDM sobre el ícono del *Trigger* y clic sobre “*Trigger Properties*”) destacan los siguientes:


- El atributo *Hidden* que se encuentra dentro del submenú *Advanced* permite configurar la visibilidad del ícono del *Trigger* al momento de ejecutar el escenario. Por defecto tendrá un valor *true* que implicará que no será mostrado al ejecutar el escenario, deberá ser cambiado al valor *false* para que se muestre.
- Los atributos *CollisionHeight* y *CollisionRadius* que se encuentran dentro del submenú *Collision* permiten configurar un volumen cilíndrico que se generará a partir de la posición en la que se encuentra ubicado el *Trigger* y que representará el volumen que al ser intersectado con algún jugador generará el evento.
- En el atributo *Event* que se encuentra dentro del submenú *Events* debe ser registrado el nombre del evento que se generará.

- En el atributo *Message* que se encuentra en el submenú *Trigger* permite registrar un mensaje que será emitido en la ejecución del escenario cada vez que el *Trigger* sea activado.
- El atributo *ReTriggerDelay* que también se encuentra dentro del submenú *Trigger* permite configurar un tiempo que el *Trigger* debe esperar para volver a ser activado.

En general este gatillador será utilizado para activar la secuencia de acciones que pueden ser definidas en un *ScriptedTrigger*.

TriggeredCondition

Corresponde a un tipo de Actor que representa la base para el manejo de condiciones en el diseño de mapas, pues posee un atributo cuyo valor puede oscilar entre dos posibles alternativas: “*True*” y “*False*”.

Para crear una nueva *TriggeredCondition* es necesario acceder al Navegador de Actores y ubicarlo bajo la categoría “*Triggers*” y seleccionarlo, para luego hacer clic con el BDM sobre alguna superficie del mapa y ejecutar la acción “*Add TriggeredCondition here*” (aparecerá un objeto representado por el ícono .


Los atributos más importantes que pueden ser modificados a partir de la Ventana de Propiedades (haciendo clic con el BDM sobre el gatillador *TriggeredCondition* que fue creado y eligiendo la opción “*TriggeredCondition Properties*”) son los siguientes:

- El atributo “*Tag*” dentro del submenú “*Events*” asignará un nombre a la condición de modo de poder ser enlazada.
- Dentro del submenú “*TriggeredCondition*” encontraremos 3 atributos que pueden ser asignados con los valores “*True*” y “*False*”:
 - *bEnabled*: que indicará que la condición se encuentra activa en caso de que su valor sea “*True*”, o inactiva en caso de “*False*”.
 - *bToggled*: que indicará si el valor de “*bEnabled*” puede ser cambiado entre sus 2 posibles valores más de 1 sola vez.
 - *bTriggerControlled*: que indicará si el valor de la condición puede ser alterada a partir de la generación de un evento a partir de un gatillador.

La forma de uso más frecuente es en conjunto con un objeto de tipo *ScriptedTrigger*, el cual se describe a continuación.

ScriptedTrigger

Este gatillador permite construir sistemas más poderosos permitiendo generar múltiples acciones a partir de las cuales se puede generar múltiples eventos, revisar condiciones (utilizando uno o más *TriggeredConditions*), construir ciclos, etc.

Para crear un nuevo *ScriptedTrigger* se debe acceder al Navegador de Actores para luego seleccionar bajo las categorías *Keypoint – AIScript – ScriptedSequence* la opción correspondiente, para luego hacer clic con el BDM sobre alguna superficie del mapa y seleccionar la opción “*Add ScriptedTrigger Here*”. Aparecerá un ícono muy parecido al del *Trigger*, un poco más pequeño y con un signo de exclamación ().

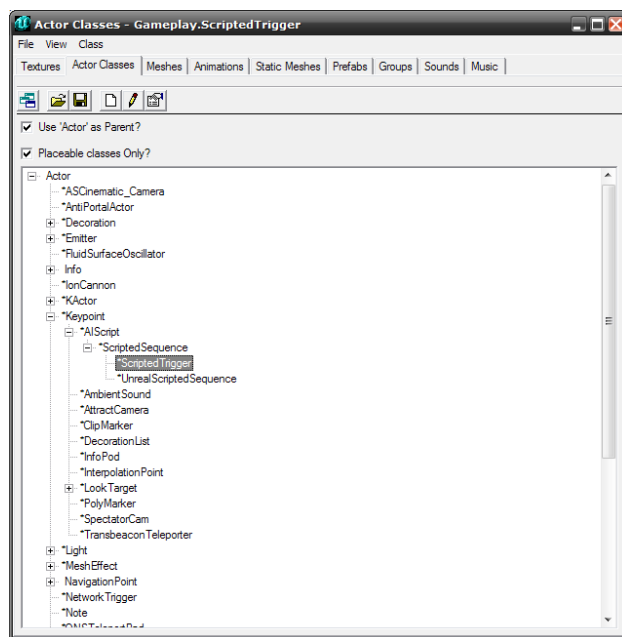


Ilustración 26. Selección de *ScriptedTrigger* a partir del Navegador de Actores

Para abrir la ventana de propiedades debemos hacer doble clic sobre el objeto creado o hacer clic con el BDM sobre el objeto y elegir la opción “*ScriptedTrigger Properties*”.

Al utilizar un *ScriptedTrigger* rara vez se utiliza alguna propiedad perteneciente al submenú *Events*. En vez de ello, las funcionalidades son implementadas dentro del submenú *AIScript*. Dentro del submenú *AIScript* encontramos el atributo *Actions* que corresponderá a un arreglo en donde se podrán configurar una serie de acciones cada vez que el *ScriptedTrigger* haya sido activado.

Para agregar nuevas acciones basta con seleccionar el atributo *Actions* y a un costado derecho aparecerán opciones “*Empty*” para vaciar la lista de acciones y “*Add*” para agregar una nueva, la cual en caso de existir acciones ya creadas se agregara al final.

Una vez generada una acción se debe escoger el tipo de acción que se deseará realizar desde una lista desplegable. Entre estas opciones destacan:

- *Action_DISPLAYMESSAGE*: que permitirá desplegar un mensaje en pantalla
- *Action_IFCONDITION*: que permitirá generar estructuras condicionales, o sea enmarcar un set de acciones que se cumplirán sólo en caso de que una cierta condición sea satisfactoria. Se necesitará identificar un Actor del tipo *TriggeredCondition* a partir del cual se revisará el cumplimiento o no de su condición. Para simbolizar el fin de acciones que englobará la estructura condicional se debe crear una acción de tipo *Action_ENDSECTION*.
- *Action_GOTOACTION*: permite desviar la ejecución de las acciones hacia una acción en particular. Esta acción deberá ser identificada a través de la posición del arreglo que ocupa.
- *Action_PLAYSOUND*: permite reproducir un sonido en particular, el cual debe ser primero seleccionado a partir del Navegador de Sonidos y luego haciendo clic en el sub atributo “*SOUND*” de la acción aparecerá la opción “*Use*” que completará el valor del sub atributo con la ruta al sonido (vía paquete) previamente elegido.
- *Action_TRIGGEREVENT*: permite configurar un evento cuyo nombre debe ser configurado en el sub atributo “*Event*”, el cual será generado una vez que la acción sea ejecutada.
- *Action_WAITFOREVENT*: corresponde a una de las acciones centrales en la creación de la mayoría de los *ScriptedTriggers* pues permite poner en acción el listado de acciones en el momento en que un cierto evento sea activado. Para ello se deberá configurar el nombre del evento que gatillará esta acción en el sub atributo “*ExternalEvent*”.
- *Action_WAITFORTIMER*: permite generar un tiempo de espera entre acciones. Posee un sub atributo llamado “*PauseTime*” que se debe configurar con la cantidad de segundos de espera para ejecutar la siguiente acción.

Se pueden construir set de acciones como el que se muestra a continuación:

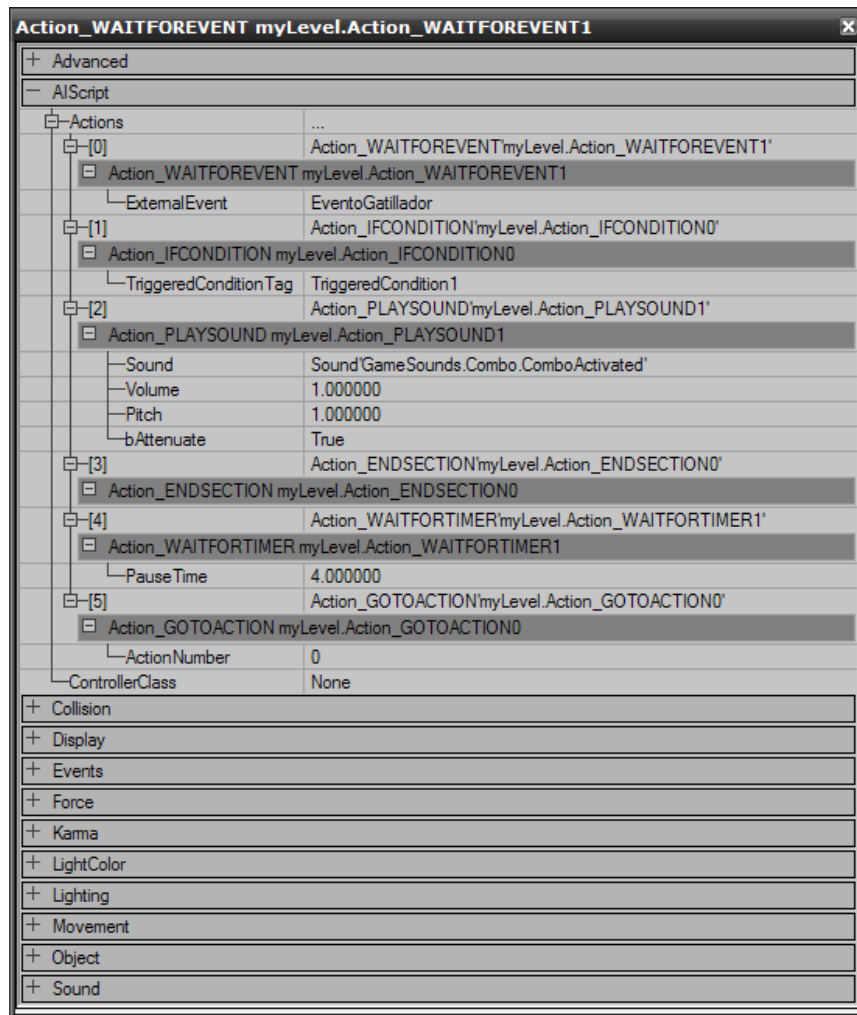


Ilustración 27. Ejemplo de *ScriptedTrigger*

En la Ilustración se muestra un Actor de tipo *ScriptedTrigger* donde las acciones han sido configuradas de la siguiente forma:

- Se espera a que se gatille un evento llamado “EventoGatillador”. Una vez que se gatille se pasa a la siguiente acción.
- Se revisa si la condición almacenada en un objeto de tipo “*TriggeredCondition*” cuyo *Tag* tiene como valor “*TriggeredCondition1*” es cierta o no. En caso de que la condición se cumpla se pasa a la siguiente acción donde se emite un sonido que se encuentra almacenado en “*GameSounds.Combo.ComboActivated*” dentro de los paquetes de

sonido. En caso contrario (que la condición no se cumpla) se salta a la acción que se encuentra a continuación de la acción de tipo *Action_ENDSECTION*.

- Luego se espera 4 segundos antes de avanzar a la siguiente acción.
- Se vuelve a la acción 0, y por lo tanto el objeto “*ScriptedTrigger*” queda a la espera de que se vuelva a ejecutar nuevamente el evento.

6.3 Implementación de Escenarios

En esta sección se explicará la forma en que fueron implementados cada uno de los escenarios ya diseñados modelándolos con la herramienta UnrealEd y utilizando las características descritas en el sub capítulo 6.2.

6.3.1 Implementación de Pistas Auditivas

De acuerdo con el diseño, se deben implementar una serie de pistas auditivas de modo de guiar en forma fluida la interacción del usuario con el nivel diseñado.

A continuación se describe la implementación de cada una de ellas:

Identificación de colisión

Al personaje del usuario se le asignará un sonido descriptivo al momento en que se encuentre desplazándose a lo largo del escenario. La herramienta *Unreal Engine* provee por defecto detección de colisiones que provoca en el personaje un cambio desde el estado en movimiento a uno estático.

Adicionalmente es posible asociarle un sonido en particular a estos dos estados del personaje, como por ejemplo sonido de pasos al estado que representa al personaje en movimiento (que viene configurado por defecto).

Lo anterior provocará que el usuario al desplazarse a través del mapa pueda escuchar sonidos de pasos (que representarán movimiento) y al colisionar con algún obstáculo o pared dejará de escuchar tales sonidos.

Ubicación espacial

La ubicación espacial consiste en orientar posicionalmente al usuario, una alternativa es orientarlo geográficamente comunicándole por ejemplo que se encuentra en la parte noroeste del mapa. Otra alternativa es asignarle nombre a las distintas áreas que componen el escenario

y comunicarle al usuario a través de un mensaje cuando esté realizando transición entre ellas, nombrándole el área al cual esta accediendo.

En el primer caso bastará con usar un *ScriptedTrigger* que ejecute las siguientes acciones:

1. Espere por un evento en particular.
2. Reproduzca un sonido en el cual se encuentra grabado un mensaje que indicará al usuario la posición en la que se encuentra dentro del mapa.
3. Esperar un lapso de tiempo determinado de modo que no se repita en forma recurrente si es que el usuario no se ha alejado de él.
4. Regresar al primer paso.

Para explicar mejor la implementación del segundo caso, suponga que existe un escenario donde un pasillo conecta con una habitación, una alternativa sería construir 2 *ScriptedTriggers* cuyos funcionamiento siga la misma línea que el mencionado para el primer caso: uno que indique el momento en el que se ingresa al pasillo (Reproduce el mensaje “Ingresando al pasillo”) y el otro la entrada de la habitación (Reproduce el mensaje “Ingresando a la habitación”). Deberán ser activados al momento que se ingresa al pasillo o a la habitación dependiendo cual sea el caso.

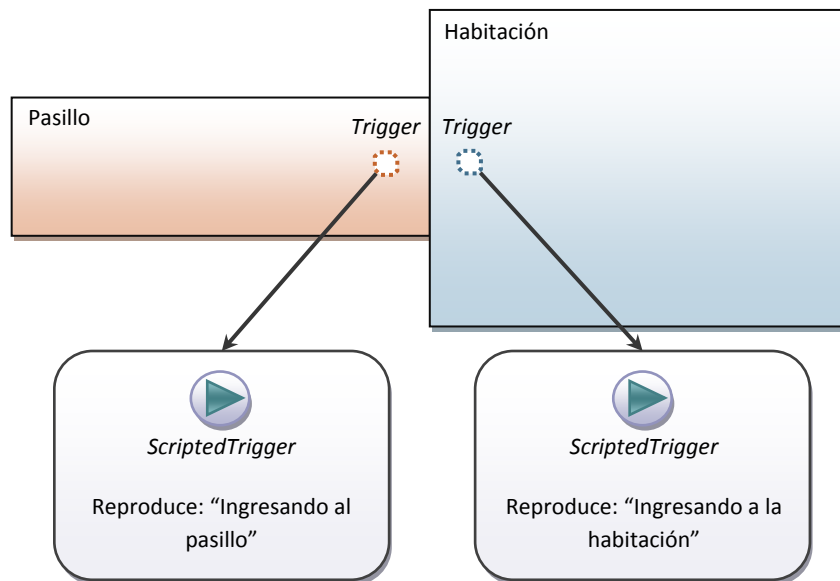


Ilustración 28. Ejemplo disfuncional de Ubicación Espacial del usuario vía retroalimentación auditiva

El problema que generaría este primer aproximamiento es que si el usuario viene desde el pasillo cuando pase por la salida del pasillo se reproducirá “Ingresando al pasillo” y luego cuando entre a la habitación se reproducirá “Ingresando a la habitación”, ocurriendo algo similar en el caso inverso en que el usuario viene desde la habitación al pasillo.

Para solucionar esto se deberá asociar a cada área definida dentro del escenario con un *TriggeredCondition* a través del cual se pueda establecer si el usuario se encuentra fuera del área o no. Volviendo al ejemplo, se deberán asociar 2 condiciones, una al pasillo y otra a la habitación, en donde el invariante general será que sólo un *TriggeredCondition* puede estar desactivado pues el usuario sólo se encontrará en un área a la vez.

Teniendo estas condiciones en cuenta la solución quedará determinada por:

- Cada zona tendrá ubicados *Triggers* en cada una de sus entradas, que generaran un evento propio y común cada entrada de la zona (denominado “Evento de Entrada”).
- Cada zona tendrá asociado un *TriggeredCondition* (denominado “Condición de Zona”), de atributos *bEnabled* igual a *False* en caso de que el usuario inicie su interacción en la zona o *True* en caso contrario. Además se deberán configurar los atributos *bToggle* y *bTriggeredControlled* como *True* de modo que el valor de la condición pueda ser modificado a medida que el usuario se traslada entre zonas.
- Cada entrada de la Zona tendrá asociada un *ScriptedTrigger* con el siguiente conjunto de acciones:
 1. Esperar el “Evento de Entrada”.
 2. Revisar la “Condición de Zona”:
 - En caso de que sea verdadera reproducir el sonido con el mensaje correspondiente a la zona.
 - En caso de que sea falso ir al paso 5.
 3. Deshabilitar todas las “Condiciones de Zona” de zonas contiguas.
 4. Habilitar la “Condición de Zona” propia.
 5. Volver al primer paso del proceso.

Con esto se logrará que el sonido sólo se active al momento de entrar a una zona y no al salir, lo cual podría causar confusión en el usuario.

Descripción de alternativas

Este tipo de retroalimentación auditiva debe ser entregada al usuario en sectores del escenario que sean no triviales, donde las posibilidades de exploración del escenario se amplíen, como por ejemplo al encontrarse con una bifurcación de pasillos.

Para ello se utilizará un *ScriptedTrigger* que ejecute las siguientes acciones:

1. Espere por un evento en particular.
2. Reproduzca un sonido en el cual se encuentra grabado un mensaje que orientará al usuario indicándole referencias para que ubique las alternativas.
3. Esperar un lapso de tiempo determinado de modo que no se repita en forma recurrente si es que el usuario no se ha alejado de él.
4. Volver al primer paso.

Podrán ubicarse uno o más *Triggers* que gatillen el evento que espera el *ScriptedTrigger* que ha sido descrito, dependiendo de si se necesita abarcar un área irregular del mapa.

Cercanía a Puntos Importantes

Lo que diferencia este tipo de retroalimentación auditiva de las otras es que el sonido debe mantenerse en continua reproducción mientras el usuario permanezca dentro de una cierta área y hasta que el punto hacia el cual se le desea guiar sea alcanzado.

Para ello será importante estructurar las siguientes componentes:

- Asociar un *TriggeredCondition* con la zona en que la retroalimentación debe ser activada, cuyo valor del atributo *bEnabled* sea *True* en caso de que el usuario se encuentre dentro de la zona, o *False* en caso contrario. Lo denominaremos “Condición de Zona”
- Asociar otro *TriggeredCondition* con la activación de la retroalimentación auditiva en base a que el usuario haya o no accedido al punto destino. El valor del atributo *bEnabled* será *True* (que además deberá ser su valor inicial) mientras el usuario no haya accedido al punto destino y *False* en caso contrario. Lo denominaremos “Condición de Fin”.
- Uno o varios *Triggers* (dependiendo del área que se deba cubrir) ubicados fuera de la zona de activación que se encargue de desactivar la “Condición de Zona” una vez que el usuario la abandone.

- Uno o varios *Triggers* que se encarguen de activar la “Condición de Zona” una vez que el usuario ingrese a la zona de activación de la retroalimentación auditiva. El evento generado por estos *Triggers* lo denominaremos “Evento de Activación”.
- Uno o varios *Triggers* que se encarguen de desactivar la “Condición de Fin” una vez que el usuario haya alcanzado el objetivo.
- Un *ScriptedTrigger* que ejecutará el siguiente set de acciones:
 1. Esperar a que se genere algún “Evento de Activación”.
 2. Revisar si la “Condición de Zona” y la “Condición de Fin” se cumplen, y en tal caso realizar las siguientes acciones:
 - Reproducir el sonido que orientará al usuario hacia el punto objetivo.
 - Volver al punto 2.
 3. Volver al punto 1.

El *ScriptedTrigger* encargado de reproducir el sonido debe ser ubicado en el punto de destino de manera de aprovechar la generación especializada de sonido que genera el motor de sonido del *Unreal Engine*.

Los *Triggers* que activan y desactivan condiciones deben hacerlo a través de *ScriptedTriggers* que revisen si la condición aún no posee el valor deseado, y en tal caso generar el evento que modifique el valor del *TriggeredCondition* correspondiente. En caso contrario no ejecutar ninguna acción.

6.3.2 Escenarios para Etapa de Apresto

Para la implementación de los escenarios de esta etapa se utilizaron pistas auditivas para la detección de colisiones y de cercanía a puntos importantes.

Dado que la estructura del escenario consta de habitaciones conectadas por pasillos se optó por utilizar pistas auditivas de cercanía a puntos importantes, para indicar la posición y dirección de los pasillos. Para ello se ubicaron *Triggers* de activación en la entrada a cada uno de los pasillos, y un *ScriptedTrigger* (emisor del sonido) ubicado entre la parte intermedia y el final del pasillo de modo de utilizar la generación especializada de sonido que posee el *Unreal Engine* para orientar al usuario en la dirección del pasillo.

6.3.3 Mapa del Metro Universidad de Chile – Nivel Superior

Para la implementación de este escenario se separó en 5 zonas principales como se indica en la siguiente figura:

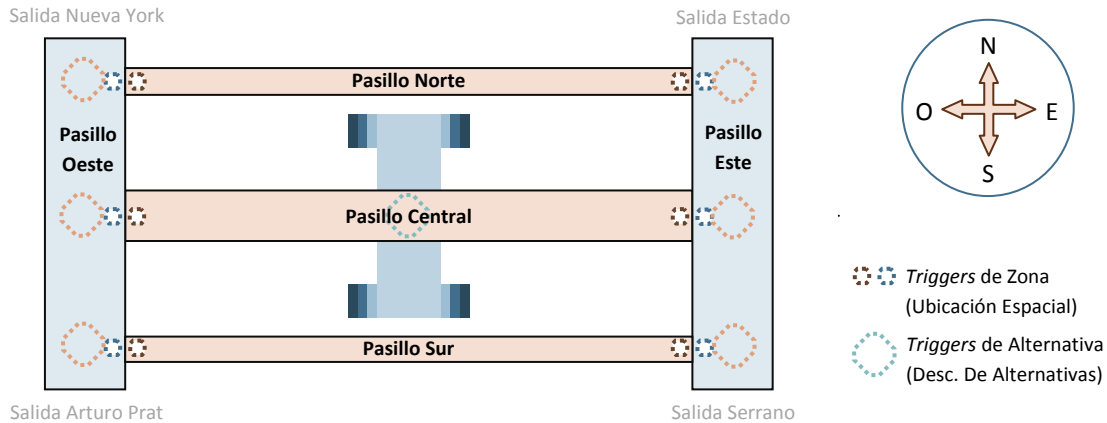


Ilustración 29. Diagramación de Zonas del escenario Metro Universidad de Chile – Piso Superior

Las zonas definidas en referencia a los puntos cardinales reales de la estación y basado en los pasillos que la definen serán entonces: pasillo Norte, pasillo Sur, pasillo Oeste, pasillo Este y pasillo Central. Donde son los pasillos Este y Oeste los que conectan con las salidas del metro y el pasillo Central es el único pasillo a través del cual se puede acceder al nivel intermedio del mapa.

Cada una de las zonas definidas tendrá ubicada en cada salida o conexión con otras zonas “*Triggers de Zona*”, para la generación de pistas auditivas correspondientes a Ubicación Espacial, de manera que se genere un sonido que entregue pistas acerca del ingreso a la zona.

Consideremos el caso del pasillo sur que aparece en la Ilustración 29, donde se encuentran 2 conexiones, la primera que conecta con el Pasillo Oeste y la segunda que conecta con el Pasillo Este. Al ser activado el Trigger ubicado en la conexión con el pasillo oeste, además de reproducir el sonido adecuado con la información de ayuda y desactivar el *TriggeredCondition* que indica que el usuario se encuentra fuera de la Zona Pasillo Sur (pues acaba de ingresar a ella), se deberá activar el *TriggeredCondition* asociado al Pasillo Oeste que indicará que el usuario se encuentra fuera de esa área (pues acaba de salir de tal zona). Esto mismo se deberá repetir en cada transición de zonas y en forma simétrica con la zona con la cual se conecta.

Además en los puntos del escenario donde el usuario tenga más de 1 alternativa de movimiento se proveerá pistas auditivas de Descripción de Alternativas basadas en diálogos de orientación donde se utilizarán los puntos cardinales.

6.3.4 Mapa del Metro Universidad de Chile – Niveles Superior e Intermedio

La implementación del mapa que incluye los niveles superior e intermedio se basará en la versión del mapa construida sólo para el nivel superior: consistirá en una ampliación de él.

En el nivel intermedio también se realizará una división mediante zonas

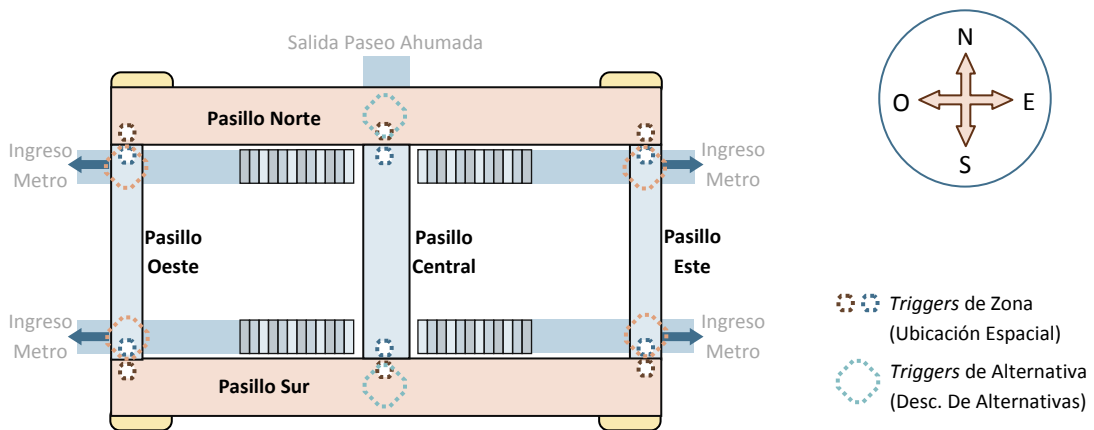


Ilustración 30. Diagramación de Zonas del escenario Metro Universidad de Chile – Piso Intermedio

La implementación de gatilladores y condiciones dentro del mapa seguirá la misma lógica del piso superior ubicados tal como se indican en la Ilustración 30.

Adicionalmente se ubicarán pistas auditivas de Ubicación Espacial a la subida y bajada de las escaleras de modo que el usuario detecte cambios en el nivel del escenario. Para ello se ubicarán *Triggers* en la parte alta y baja de cada una de las escaleras que generen la reproducción de mensajes que indiquen que se encuentra en la parte superior e inferior de la escalera según corresponda.

7 Testeo Cognitivo

7.1 Usuarios Finales

7.1.1 Descripción General

Los Usuarios Finales corresponden a estudiantes del Colegio de Ciegos Hellen Keller con distintos grados de ceguera, que se encuentran cursando entre 6° y 8° de educación básica, y cuyas edades oscilan entre los 12 y 18 años.



Ilustración 31. Foto tomada a la entrada del Colegio Hellen Keller de la comuna de Ñuñoa

Es importante mencionar que los usuarios con los cuales se testeó el sistema poseen conocimientos básicos acerca de tecnologías, pueden manejar varias aplicaciones en forma autónoma basándose en audio y sintetizadores de voz.

7.1.2 Descripción Específica

El listado de Usuarios finales detallado se muestra a continuación:

N°	Identificador	Curso	Edad	Diagnóstico	Grado de Visión	Otros Diagnósticos
1	Usuario 1	7°	14	Desprendimiento de retina en ambos ojos	Ceguera Total	No tiene
2	Usuario 2	8°	15	Retinopatía bilateral.	Ceguera Total	No tiene
3	Usuario 3	8°	14	Catarata congénita.	Baja Visión	No tiene
4	Usuario 4	8°	14	Microcórnea microftalmia.	Ceguera Total	No tiene
5	Usuario 5	7°	12	Distrofia tapeto retinal.	Baja Visión	Visión Excéntrica
6	Usuario 6	8°	14	Catarata congénita. Glaucoma.	Baja Visión	No tiene
7	Usuario 7	8°	13	Degeneración tapeto retinal bilateral. Nistagmus horizontal congénito retinitis.	Baja Visión	No tiene
8	Usuario 8	8°	18	Glioma de Quiasma óptico.	Ceguera Total	Hemiparético izquierdo
9	Usuario 9	6°	12	Amaurosis congénita	Ceguera Total	No tiene

Tabla 1. Listado de Usuarios Finales participantes en los tests

7.2 Etapa de Apresto – Primer Mapa

7.2.1 Actividades

Las actividades para el primer escenario de esta etapa consistieron en:

- Breve introducción
- Explicación del modelo de desplazamiento del sistema y detección de colisiones
- Tiempo de interacción con el software
- Definición de objetivos: que consistía en desplazarse por el escenario encontrando 2 cajas musicales
- Explicación verbal del recorrido realizado

7.2.2 Resultados

Los resultados se categorizaron en una escala del 1 al 5 donde los valores representan:

Valor Escala	Descripción
1	No logra realizar la tarea
2	Logra realizar la tarea con dificultad y no es capaz de describir el camino realizado ni la diagramación del escenario
3	Logra realizar la tarea pero no es capaz de describir fielmente la diagramación del escenario
4	Realiza satisfactoriamente la tarea y logra explicar verbalmente y con acierto el recorrido realizado
5	Realiza satisfactoriamente la tarea y logra explicar verbalmente el recorrido realizado y la posición de las cajas musicales dentro del escenario

Tabla 2. Tabla de Categorización para la 1ra parte de la Etapa de Apresto

Los usuarios que participaron en este testeo fueron:

N°	Identificador	Grado de Visión	Resultado	Comentarios
1	Usuario 2	Ceguera Total	3	
2	Usuario 3	Baja Visión	4	El grado de visión que tiene el usuario no fue relevante para este testeo pues no se le suministró retroalimentación visual, sólo basada en sonido.
3	Usuario 7	Baja Visión	2	El grado de visión que tiene el usuario no fue relevante para este testeo pues no se le suministró retroalimentación visual, sólo basada en sonido.
4	Usuario 8	Ceguera Total	3	

Tabla 3. Resultados obtenidos por alumno para la primera parte de la etapa de Apresto

7.2.3 Comentarios

La detección de colisiones fue totalmente aceptada por los usuarios pues lograban identificar los límites del escenario en forma fluida y buscar alternativas de movimiento que los llevaran a cumplir el objetivo.

Tras el testeo se logró identificar positivamente el reconocimiento de direccionalidad del sonido generado por el motor de sonido del *Unreal Engine* identificado por todos los usuarios al activar fuentes sonoras que se encontraban a distancia del personaje del usuario.

Si bien los usuarios lograron detectar que dentro del escenario existía una bifurcación, donde debían elegir una de las dos alternativas, no resultaba natural realizar el recorrido de regreso y tomar la alternativa restante. La mayoría de los usuarios debió pasar repetidas veces por la intersección para finalmente elegir la alternativa que aún no era explorada donde podían encontrar la segunda caja musical. Esta característica fue mejorada en los siguientes mapas a partir del análisis desarrollado en esta etapa.

Se detectó que las pistas suministradas por el primer mapa confundían al usuario en algunos casos, lo cual motivó varias de las propiedades y tipos de pistas auditivas descritas en este informe para escenarios posteriores.

7.3 Etapa de Apresto – Segundo Mapa

7.3.1 Actividades

Las actividades para el segundo escenario de esta etapa consistieron en:

- Breve introducción (para los usuarios que aún no habían participado en el testeo)
- Interacción con una maqueta que poseía la misma estructura que el mapa
- Explicación del modelo de desplazamiento del sistema y detección de colisiones
- Tiempo de interacción con el software (para los usuarios que aún no habían participado en el testeo)
- Definición de objetivos: que consistía en desplazarse por el escenario encontrando 2 cajas musicales
- Explicación gestual (ayudado de la maqueta) acerca del recorrido realizado
- Ubicación (dentro de la maqueta) de la posición de las 2 cajas musicales previamente encontradas utilizando el software



Ilustración 32. Maqueta utilizada para la segunda parte de la etapa de apresto

7.3.2 Resultados

Los resultados se categorizaron en una escala del 1 al 5 donde los valores representan:

Valor Escala	Descripción
1	No logra realizar la tarea
2	Logra realizar la tarea con dificultad y no es capaz de ubicar en la maqueta las posiciones donde son encontradas las cajas ni describir el recorrido realizado
3	Logra realizar la tarea pero no es capaz de describir fielmente la posición de las cajas, aunque sí el recorrido realizado
4	Realiza satisfactoriamente la tarea y ubica las cajas dentro de las habitaciones correctas y es capaz de describir en forma cercana el recorrido realizado
5	Realiza satisfactoriamente la tarea y y ubica las cajas dentro de las habitaciones correctas en la posición adecuada

Tabla 4. Tabla de Categorización para la 1ra parte de la Etapa de Apresto

Los usuarios que participaron en este testeo fueron:

N°	Identificador	Grado de Visión	Resultado	Comentarios
1	Usuario 2	Ceguera Total	5	
2	Usuario 3	Baja Visión	4	El grado de visión que tiene el usuario no fue relevante para este testeo pues no se le suministró retroalimentación visual, sólo basada en sonido.
3	Usuario 4	Ceguera Total	4	
4	Usuario 6	Baja Visión	5	El grado de visión que tiene el usuario no fue relevante para este testeo pues no se le suministró retroalimentación visual, sólo basada en sonido.
5	Usuario 8	Ceguera Total	3	

Tabla 5. Resultados obtenidos por alumno para la primera parte de la etapa de Apresto



Ilustración 33. Fotografías tomadas en la segunda parte de la etapa de apresto

7.3.3 Comentarios

Las mejoras en las pistas auditivas y la visualización táctil del escenario mejoraron el desempeño general de los usuarios con respecto a la primera parte.

Identificaron en forma más fluida las conexiones entre habitaciones y las cajas musicales ubicadas dentro de estas últimas.

En general se logró identificar el ciclo establecido dentro del escenario sin mayores problemas ni confusiones.

7.4 Metro Universidad de Chile – Nivel Superior

7.4.1 Actividades

Las actividades para este escenario consistieron en:

- Breve introducción
- Interacción del usuario con el software
- Explicación verbal del escenario recorrido por parte del usuario
- Definición de objetivos: desplazamientos dentro del mapa que impliquen el uso de varios pasillos
- Explicación verbal y gestual por parte del usuario del escenario y los desplazamientos realizados

7.4.2 Resultados

Los resultados se categorizaron en una escala del 1 al 5 donde los valores representan:

Valor Escala	Descripción
1	No logra realizar la tarea
2	Logra realizar la tarea con dificultad y no es capaz de describir el camino realizado ni la diagramación del escenario
3	Logra realizar la tarea pero no es capaz de describir fielmente la diagramación del escenario
4	Realiza satisfactoriamente la tarea y logra explicar verbalmente y con acierto el recorrido realizado
5	Realiza satisfactoriamente la tarea y logra explicar verbalmente el recorrido realizado y la posición de las cajas musicales dentro del escenario

Tabla 6. Tabla de Categorización el test del escenario del nivel superior de la estación de metro

Los usuarios que participaron en este testeo fueron:

N°	Identificador	Grado de Visión	Resultado	Comentarios
1	Usuario 1	Ceguera Total	5	
2	Usuario 2	Ceguera Total	4	
3	Usuario 3	Baja Visión	4	El grado de visión que tiene el usuario no fue relevante para este testeo pues no se le suministró retroalimentación visual, sólo basada en sonido.
5	Usuario 5	Baja Visión	5	El grado de visión que tiene el usuario no fue relevante para este testeo pues no se le suministró retroalimentación visual, sólo basada en sonido.
8	Usuario 7	Baja Visión	5	El grado de visión que tiene el usuario no fue relevante para este testeo pues no se le suministró retroalimentación visual, sólo basada en sonido.
9	Usuario 8	Ceguera Total	4	
10	Usuario 9	Ceguera Total	5	

Tabla 7. Resultados obtenidos por alumno para la etapa de test del escenario del nivel superior de la estación de metro

7.4.3 Comentarios

Las pistas auditivas gozaron de gran aceptación y usabilidad dentro de los usuarios pues pudieron desplazarse de forma fluida y sin necesidad de ayuda por todo el escenario.

El cumplimiento de los objetivos fue rápido, directo y fluido una vez que los usuarios habían interactuado con el software.

Los usuarios en general establecieron una diagramación del escenario fiel y sin errores, luego de haber interactuado en forma libre y de haber cumplido con algunos objetivos de movimiento dentro del mapa.

7.5 Metro Universidad de Chile – Nivel Superior e Intermedio

7.5.1 Actividades

Las actividades para este escenario consistieron en:

- Interacción del usuario con el software
- Explicación verbal del escenario recorrido por parte del usuario
- Definición de objetivos secundarios: desplazamientos dentro del mapa que impliquen el uso de varios pasillos y cambios de nivel
- Definición de un objetivo principal: iniciar desde una de las salidas ubicadas en el nivel superior del escenario, llegar a una de las boleterías para luego acceder a alguna de las entradas al metro.
- Explicación verbal y gestual por parte del usuario del escenario y los desplazamientos realizados

7.5.2 Resultados

Los resultados se categorizaron en una escala del 1 al 5 donde los valores representan:

Valor Escala	Descripción
1	No logra realizar la tarea
2	Logra realizar la tarea con dificultad y no es capaz de describir el camino realizado ni la diagramación del escenario
3	Logra realizar la tarea pero no es capaz de describir fielmente la diagramación del escenario
4	Realiza satisfactoriamente la tarea y logra explicar verbalmente y con acierto el recorrido realizado
5	Realiza satisfactoriamente la tarea y logra explicar verbalmente el recorrido realizado y la posición de las cajas musicales dentro del escenario

Tabla 8. Tabla de Categorización para el Test basado en escenario multi nivel

Los usuarios que participaron en este testeo fueron:

N°	Identificador	Grado de Visión	Resultado	Comentarios
1	Usuario 1	Ceguera Total	5	
2	Usuario 2	Ceguera Total	5	
5	Usuario 5	Baja Visión	5	El grado de visión que tiene el usuario no fue relevante para este testeo pues no se le suministró retroalimentación visual, sólo basada en sonido.
8	Usuario 7	Baja Visión	5	El grado de visión que tiene el usuario no fue relevante para este testeo pues no se le suministró retroalimentación visual, sólo basada en sonido.
10	Usuario 9	Ceguera Total	5	

Tabla 9. Resultados obtenidos por alumno en el Test basado en el escenario multi nivel

7.5.3 Comentarios

Los usuarios han aprendido a desplazarse sin ningún inconveniente dentro del escenario y poseen un modelo mental del recorrido determinado como objetivo principal de las actividades a desarrollar.

Son capaces de describir cualquier desplazamiento de un punto del escenario a otro sin necesidad de efectuar el recorrido en forma directa con la aplicación.

7.6 Etapa Final – Desplazamiento en el Metro Universidad de Chile

En la última etapa del testeo los usuarios fueron llevados al Metro Universidad de Chile con el objetivo de que realizaran el mismo recorrido realizado en la etapa anterior utilizando la aplicación, pero ahora en forma real y sin uso de pistas auditivas, pero sí con ayuda de conocimientos previos de desplazamiento en espacios abiertos.

Para ello los usuarios fueron ubicados en la entrada suroeste del nivel superior del Metro Universidad de Chile de cara al Norte (tal como inicia la aplicación) y ellos haciendo uso de su bastón y modelo mental generado a partir del uso del software debían llegar a la boletería noreste ubicada en el nivel intermedio de la estación, luego ubicar la entrada más cercana a los andenes (torniquetes) sin atravesarla, luego subir nuevamente al nivel superior, dirigirse al pasillo este, y volver al punto inicial utilizando el pasillo sur.

7.6.1 Comentarios

El testeo fue realizado entre las 13:30 y las 14:15 hrs de un día lunes, tiempo durante el cual la estación de metro se mantuvo con una cantidad considerable de personas que se movilizaban por el lugar. Esto complicó el testeo pues los usuarios no pudieron desplazarse libremente por la estación y debieron ser ayudados a sortear grupos de personas por el acompañante facilitador a cargo.

Todos los usuarios lograron realizar la actividad sin mayores complicaciones que las generadas por la aglomeración de personas en algunas partes de la estación y la identificación de las intersecciones de pasillos que se veían bloqueados en algunas partes por la cantidad de gente. Salvo 2 casos de confusiones en donde los usuarios perdían su punto de referencia, que era recordado por el facilitador, los alumnos seguían su recorrido.

8 Conclusiones

Actualmente la calidad de los recursos tecnológicos que se encuentran al alcance de usuarios con discapacidad visual es muy baja comparada con versiones comerciales, esto provoca que personas sin discapacidad visual no se vean atraídas a interactuar con este tipo de aplicaciones, y por lo tanto quedan relegadas únicamente a usuarios no videntes. La motivación de este estudio se basa en demostrar que la utilización de herramientas como el Unreal Engine (orientadas principalmente a desarrollar aplicaciones con gran calidad gráfica, de sonido y con un gran realismo) pueden ser eficazmente utilizadas para construir aplicaciones para personas con discapacidad.

Esto podría motivar construcciones posteriores de aplicaciones que incluyan interacciones entre personas con y sin discapacidad visual, siendo igual de atractivas para ambos segmentos.

Se estableció un proceso de avance entre etapas, basadas inicialmente en supuestos y patrones utilizados en programas desarrollados para usuarios no videntes, sobre el cual se fue mejorando cada vez más la solución. El proceso se define a continuación:

- En las primeras pruebas se utilizaron pistas basadas en sonido localizado espacialmente, la cual ayudó en forma efectiva a orientar a los usuarios en cuanto a la direccionalidad de movimiento. Sin embargo este tipo de pistas auditivas no resultó tan eficiente como se esperaba, pues incluso en los escenarios más sencillos se requerían pistas que entregaran mayor información acerca de la ubicación, pues los usuarios no eran capaces de formarse un modelo mental fiel al espacio virtual que estaban recorriendo.
- Es a partir del razonamiento previo que se opta por intentar una nueva versión de pistas auditivas, donde se elige indicar en forma más directa los lugares del escenario en los que se encuentra y dejar de lado la direccionalidad, dando pie a que el usuario analice las alternativas de movimiento por ensayo y error (tal y como lo haría en la realidad). Es por ello que se definen sectores del escenario los cuales informan al usuario el momento de ingreso al sector y el momento de transferencia hacia un nuevo sector con frases como: “Pasillo” que indica el ingreso a un pasillo, “Saliendo Pasillo” que indica la salida del pasillo, “Habitación” que indica el ingreso a una habitación o “Saliendo Habitación” que indica la salida de la habitación. Sin embargo aún así los usuarios no eran capaces de identificar si después de movilizarse por el escenario y volver, por ejemplo, a alguna habitación previamente recorrida, que habían realizado un ciclo. Los ciclos sólo fueron

capaces de identificarlos efectivamente cuando se les permitió interactuar con una maqueta del escenario en forma previa.

- Lo anterior motivó a que el escenario además de identificar transiciones y definiciones de sectores en su interior, se debía asignar un nombre que los identificara en forma única. Para ello, y luego de realizar un ejercicio con la mayoría de los usuarios, se definió que la utilización de puntos cardinales podría resultar sumamente efectiva para que se pudieran formar un modelo mental del escenario. Se construyó un escenario que además de incluir una versión mejorada de las pistas auditivas utilizadas en la segunda etapa de las pruebas, indicaba además mediante referencias en base a puntos cardinales la posición del usuario dentro del mapa. A través de esto el usuario, al recorrer el escenario, se formó un modelo mental e identificó rápidamente los ciclos.
- Al momento de representar cambios de nivel se utilizó el mismo sistema determinado en el punto anterior, definiendo las escaleras cómo secciones del escenario, en donde se le indica al usuario el momento en que ingresa a las escaleras (indicando además si se encuentra subiendo o bajando) y el momento en que abandona el sector (típicamente con un mensaje auditivo tal como "Acabas de bajar las escaleras")

Una característica que se puede deducir a partir de los resultados obtenidos en las etapas de apresto es que el software exige un grado de aprendizaje inicial, es decir, se hace necesario el desarrollo de actividades iniciales en donde los usuarios conozcan el programa y desarrollen habilidades que les permitan desplazarse con mayor fluidez dentro del escenario.

A partir de las actividades desarrolladas utilizando la aplicación se puede concluir finalmente que los usuarios que participaron en los test finales lograron establecer en su modelo mental el escenario de la estación del Metro que fue recorrido en forma virtual. El escenario construido con el *UnrealEd* junto con las pistas auditivas definidas en base a los puntos cardinales fueron suficientes para emular virtualmente el escenario elegido, de modo de entregar a los usuarios la información necesaria para que pudieran desenvolverse, orientarse y movilizarse con fluidez.

Las mayores dificultades al momento de realizar las pruebas en el lugar físico (estación del metro Universidad de Chile) no estuvo en dudas que hayan surgido en los usuarios sobre el modelo mental de la estación, sino que en la cantidad de gente que por ese momento transitaba y que dificultaba de sobremanera el desplazamiento de los usuarios, y que en algunos casos no permitían que ubicaran las intersecciones de pasillos, parte esencial que permite emular el recorrido hecho con el software.

Si bien en estudios anteriores se construyeron aplicaciones muy similares, el valor agregado de este estudio se basa en los siguientes 3 puntos:

- A diferencia de aplicaciones donde los movimientos que puede realizar el usuario dentro del ambiente virtual son discretos, el modelo utilizado en este estudio corresponde a un modelo continuo, es decir, es muy difícil que el usuario al moverse en el escenario repita la posición en la que se encuentra en forma exacta. Esta característica aumenta la complejidad de desplazamiento dentro del escenario a la vez que acerca la interacción a un modelo mucho más fiel a la realidad de desplazamiento de los usuarios.
- Se utiliza el motor *Unreal Engine* para modelar el desarrollo de los escenarios así como la construcción de elementos que entreguen pistas auditivas en el desplazamiento a través del escenario. Para ello se analizaron las componentes más importantes de la herramienta que pueden permitir construir escenarios complejos realizando tareas de relativamente baja complejidad. Esta herramienta trae consigo 2 características que no fueron evaluadas en este estudio pero que vale la pena mencionarlas y dejar latente la inquietud:
 - Desarrollar una aplicación donde varios usuarios puedan interactuar en forma simultánea dentro de un mismo escenario (pues el *Unreal Engine* puede manipular conexiones mediante red).
 - Construir escenarios más atractivos visualmente, donde existan tareas que puedan ser desarrolladas en conjunto por usuarios tanto videntes como no videntes en forma simultánea.
- Los escenarios para la etapa de Test fueron desarrollados a partir de una simplificación de la estación del Metro Universidad de Chile, por lo tanto la índole de la evaluación no sólo se orienta a evaluar la construcción de un modelo mental del un escenario que no posee utilidad para los usuarios, sino que se pretende demostrar los usuarios al utilizar el software son capaces de construir un modelo mental que les permitirá desplazarse en forma más fluida por un entorno que no han conocido en forma directa previamente.

9 Trabajo Futuro y Expectativas

A partir de este estudio se comprobó que es posible aplicar el uso de herramientas que asisten la construcción de Entornos Virtuales (como el *UnrealEd*) de modo de simular escenarios reales relevantes, de modo que puedan ser recorridos y asimilados por usuarios no videntes.

Actualmente el proceso de aprendizaje de movilidad y orientación por parte de niños no videntes se realiza *in situ*, bajo la compañía y supervisión de un educador a cargo, lo cual implica una gran cantidad de esfuerzo y tiempo invertido en una baja cantidad de aprendices.

La utilización de estos espacios virtuales reales podría permitir que usuarios no videntes repasen trayectos o recorridos antes de salir del lugar de origen hacia su destino, permitiendo un mayor grado de independencia y seguridad en el recorrido.

Los usuarios, a través de estos espacios virtuales reales, serán capaces de conocer remotamente muchos más lugares, pues no existirá la necesidad de tener un educador a cargo del proceso, no se correrán peligros al intentar nuevas rutas y se podrá repetir cuantas veces se quiera y a cualquier hora.

El trabajo del educador en ningún caso podrá ser reemplazado por este software, su tarea consistirá en asistir la interacción de los usuarios con el escenario, guiándolos y enseñándoles todo aquello que el software no hace: cómo cruzar una calle, cómo identificar esquinas de calles, identificar peligros, etc. La intención de esta iniciativa consiste en potenciar la labor del educador, aprovechando al máximo el tiempo asignado para el aprendizaje de orientación y movilidad de los alumnos.

Adicionalmente la herramienta *Unreal Engine* permite construir aplicaciones cerradas a partir de escenarios terminados, lo que mejoraría la usabilidad del software dado que en la actualidad se debe abrir una aplicación intermedia (*UnrealEd*), cargar el mapa y luego ejecutarlo, lo cual resulta muy poco amigable.

Otra de las posibilidades que otorga la herramienta es la participación simultánea de distintos usuarios en un mismo escenario, lo cual da pie a la construcción de escenarios donde exista participación conjunta varios usuarios en forma simultánea.

Esto podría permitir definir escenarios donde usuarios deban realizar tareas colaborativas a partir de escenarios con los que previamente hayan interactuado, de modo de medir la aplicación de conocimiento a un nivel secundario.

10 Referencias

1. Sánchez, J., Lumberas, M. & Silva, J. (1998). *Virtual reality and learning: trends and issues. Proceedings of The Fourteenth International Conference on Technology and Education*, Oslo, Norway , Agosto
2. Sánchez, J., Flores, H. (2004). *Memory enhancement through audio. Published in the Proceedings of The Sixth International ACM SIGACCESS Conference on Computers and Accessibility*, Assets 2004, Atlanta , Georgia , USA , October 18-20, pp. 24-31
3. Sánchez, J., Sáenz, M. (2005). *Sound Immersed Virtual Environments for Blind Children. Proceedings of The Fourth International Workshop on Virtual Rehabilitation IWVR '05*. Catalina Island, California, USA. September 19-21, 2005, pp. 192-202
4. Sánchez, J., Sáenz, M. (2005). *3D Sound Interactive Environments for Problem Solving. Proceedings of The Seventh International ACM SIGACCESS Conference on Computers and Accessibility*, Assets 2005, Baltimore, Maryland, USA, October 9-12, pp. 173-178
5. Sánchez, J., Elías, M. (2006). *Blind Children Learning Science through Audio-Based Interactive Software*. Proceedings of VII Congreso Internacional de Interacción Persona-Ordenador, Interacción 2006. Noviembre 13-17, 2006. Universidad de Castilla-La Mancha, Puertollano (Ciudad Real), España, pp. 591-600
6. Sánchez, J., Hassler, T. (2006). *AudioMUD: A Multi-User Virtual Environment for Blind People. Proceedings of The 5th International Workshop on Virtual Rehabilitation, IWVR '06*. New York , USA . August 29-30, 2006 , pp. 64-71
7. Sánchez, J., Zuñiga, M. (2006). *Evaluating the Interaction of Blind Learners with Audio-Based Virtual Environments. 11th Annual CyberTherapy 2006 Conference: Virtual Healing: Designing Reality. June 13 - 15, 2006*. Gatineau, Canada, pp. 66
8. *Unreal Technology Online* – <http://www.unrealtechnology.com/> (Visita: 10 de Abril 2007)
9. *Epic Games Online* – <http://www.epicgames.com/> (Visita: 10 de Abril de 2007)
10. Rizzo, A. (2006) Expose, Distract, Motivate and Measure: Virtual Reality Games for Health. En J. Sánchez (Ed.): *Nuevas ideas en Informática Educativa*. ISBN 956-310-430-7, Volumen 2, pp 1-4, 2006 © LOM Ediciones 2006.
11. Lumberas M., *Hiperhistorias Interactivas para Niños No Videntes, Memoria para optar al grado de magister en ciencias mención computación*, Departamento de Ciencias de la Computación, Universidad de Chile, Santiago, 1998.
12. Busby J., Parrish Z., Van Eenwyk J. (2005) *Mastering Unreal Technology: The Art of Level Design, Epic Games*, capítulos 1-3, 6, 9, 14 y 20. ISBN 0-672-32692-2, © Sams Publishing.

13. Sánchez, J., Flores, H., Sáenz, M. (2005). Blind Children Developing Mathematics Skills through Audio. Published in the *Proceedings of ITI 2nd Internacional Digital Media Conference in Egypt, DIGIMEDIA 2005* . Cairo, Egipto, Marzo 7-8, 2005, pp. 155-166
14. Sánchez, J., Sáenz, M. (2003) La Granja de Theo y Seth, aprendizaje de las Matemáticas en Niños con Discapacidad Visual a través de Audio, *Actas 8vo. Taller Internacional de Software Educativo TISE 2003* , Chile, Noviembre 24-26, 2003

11 Anexos

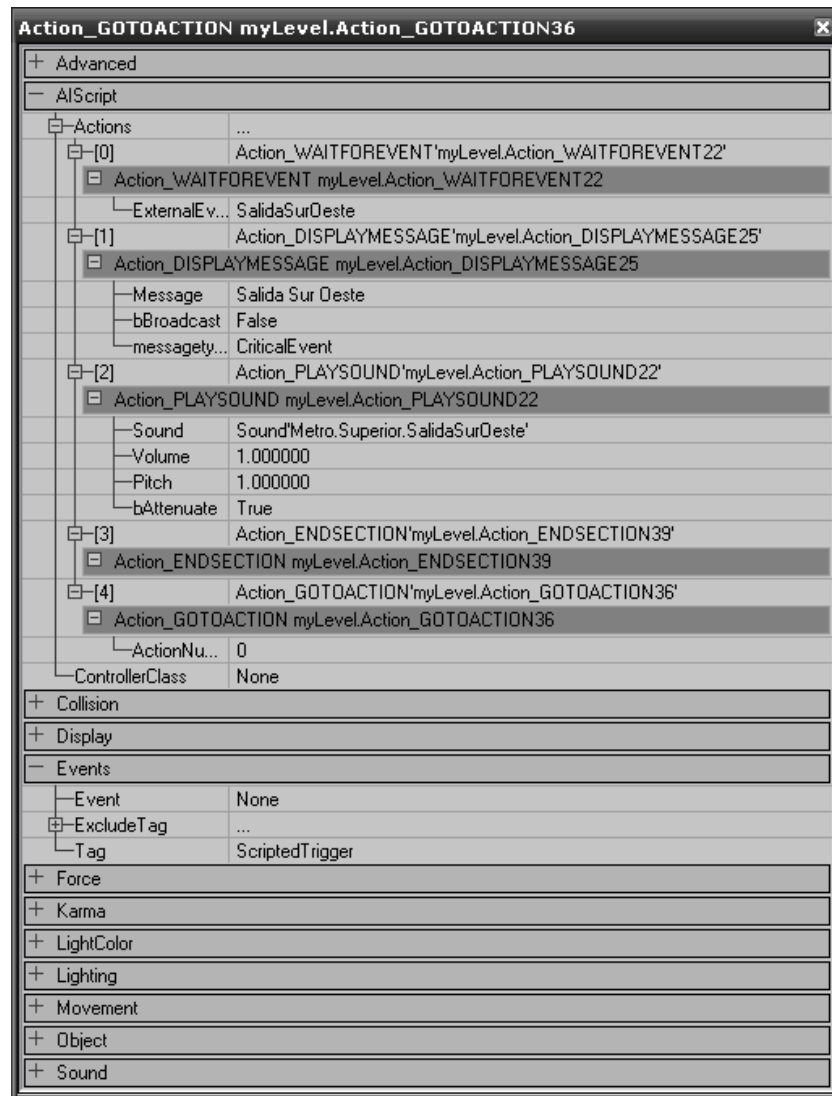
11.1 Anexos de Implementaciones de Patrones de Sonido

11.1.1 Patrón de Sonido Simple

A continuación el gatillador (*Trigger*) que activa el patrón de sonido simple:

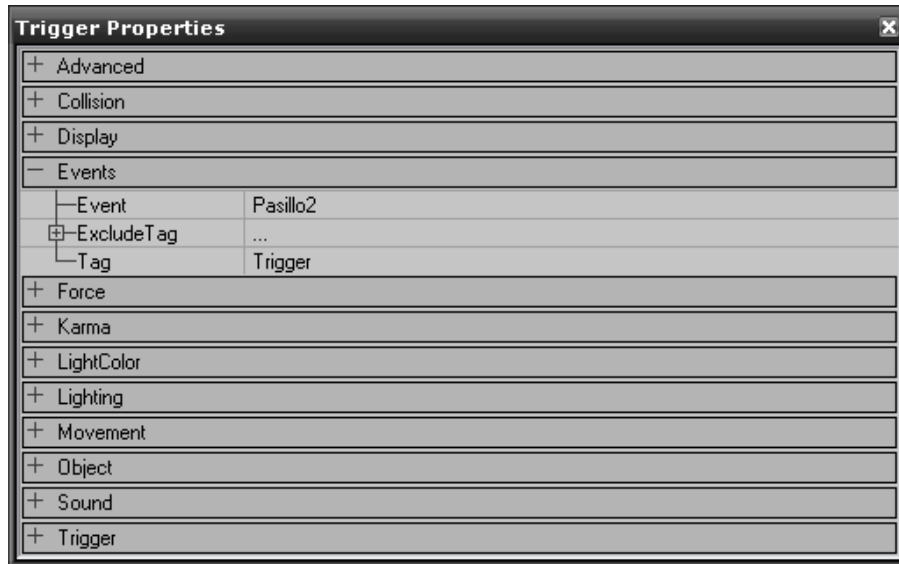


A continuación el *ScriptedTrigger* utilizado para generar un sonido que indica que el usuario se encuentra en la salida suroeste en el escenario utilizado como representación del Metro Universidad de Chile:

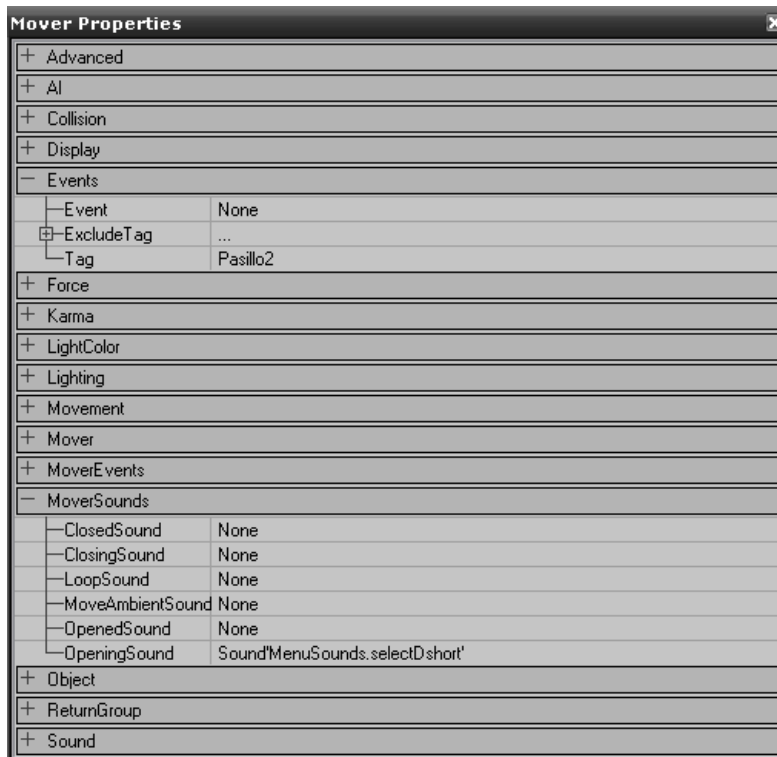


11.1.2 Patrón de Dirección

A continuación se muestra el Evento que gatilla el evento para activar el movimiento de un *Mover* en el escenario:



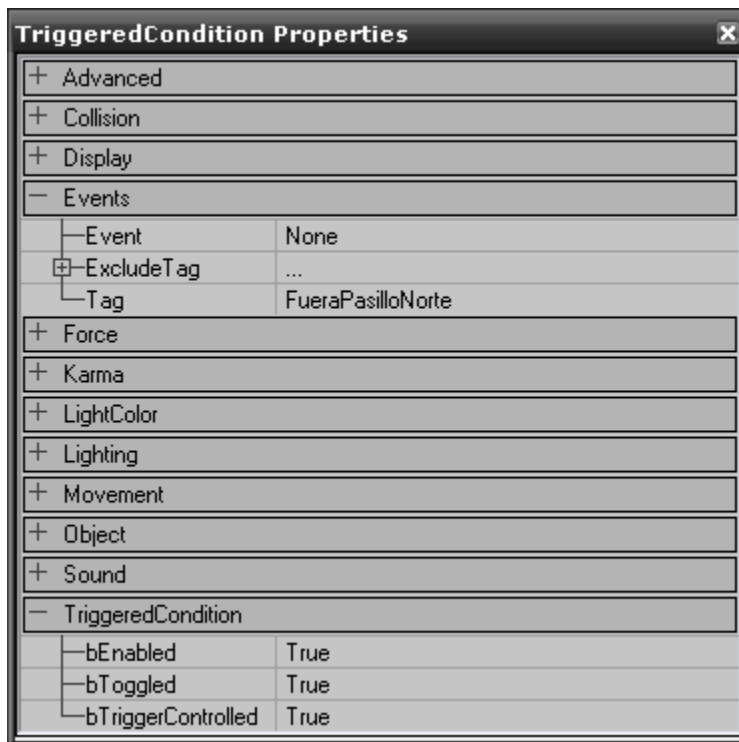
A continuación se muestran las propiedades más importantes determinadas en el Mover de modo de que se genere un sonido en particular (valor almacenado en la propiedad *Opening Sound*) y se ejecute el movimiento:



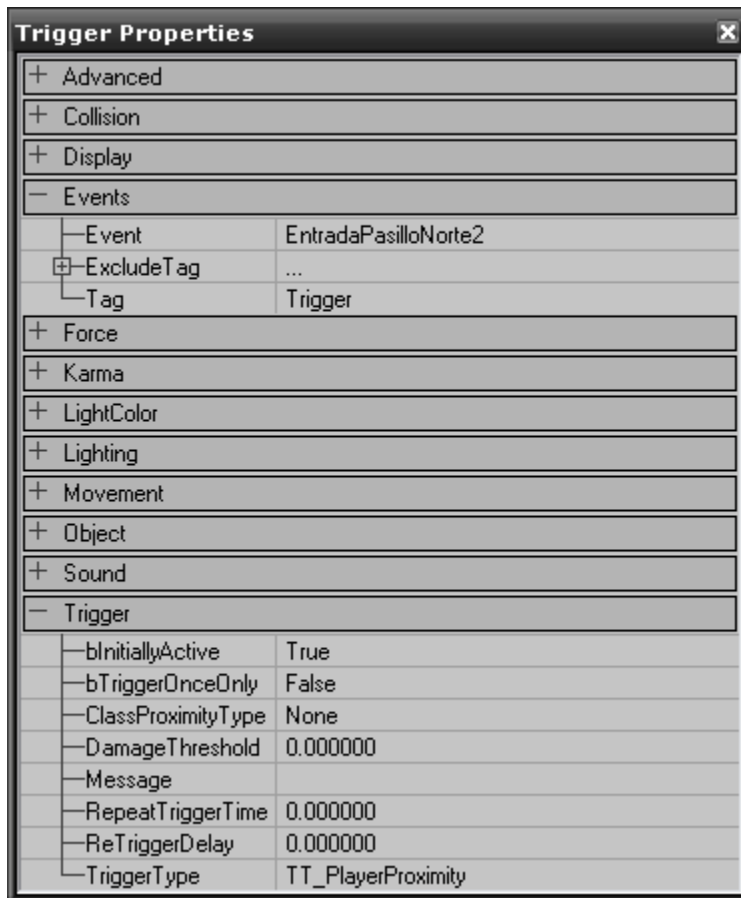
11.1.3 Patrón de Definición de Sectores

Lo que se muestra a continuación corresponde a elementos ubicados en el sector definido como “Pasillo Norte” en los escenarios relacionados con la estación de Metro Universidad de Chile.

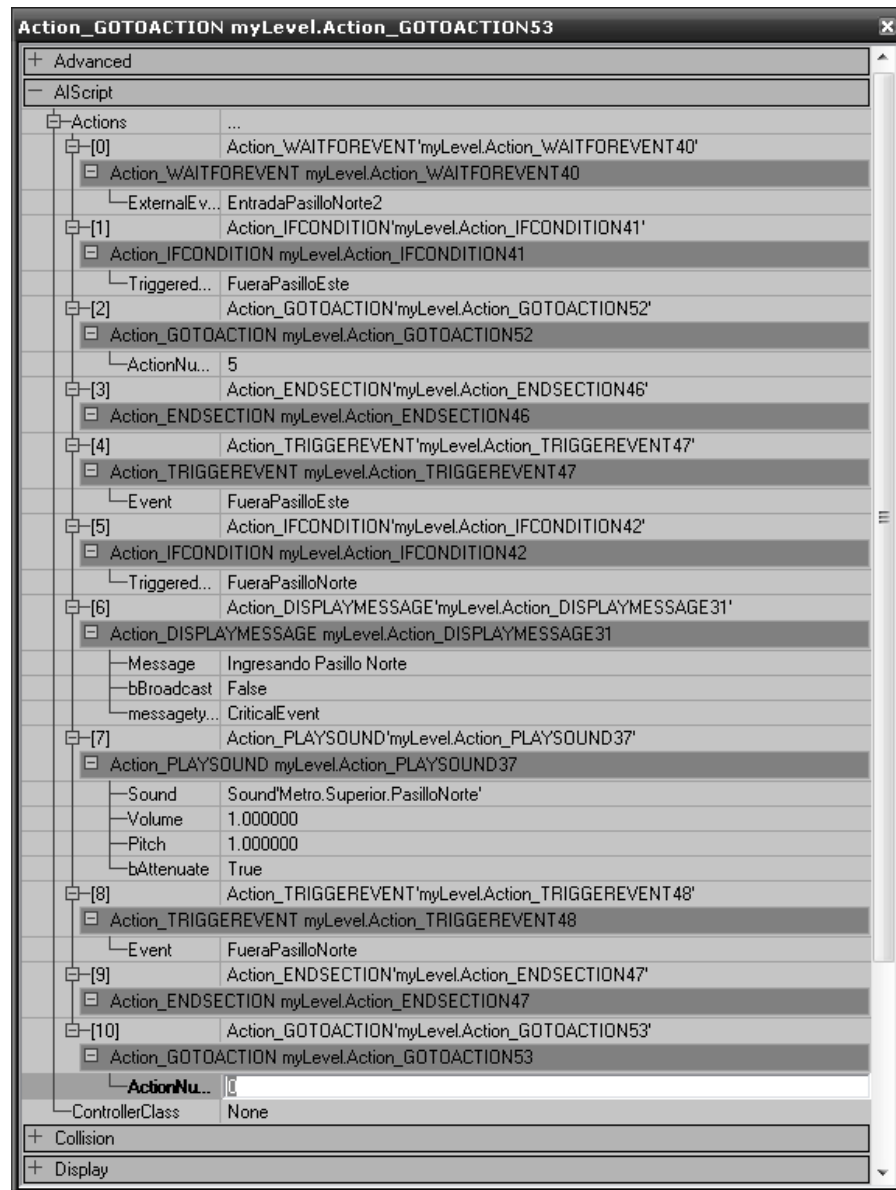
A continuación se muestran las propiedades más importantes para definir un elemento *TriggeredCondition* que permita almacenar un valor que indicará si el usuario se encuentra o no dentro del sector denominado “Pasillo Norte”:



A continuación se muestran las propiedades del gatillador que se encuentra en la entrada del Pasillo Norte y que se utilizará para activar la condición:



A continuación se despliega el contenido del *ScripterTrigger* que se encarga de activar la condición que indica que el usuario ha salido del sector del cual proviene, de desactivar la condición que indica que el usuario se encuentra fuera del “Pasillo Norte” (pues acaba de ingresar a él) y de generar el sonido que así lo indica:



11.1.4 Patrón de Bloqueo de Sectores

A continuación se despliega el contenido del *ScriptedTrigger* utilizado para gatillar el evento que provocará que los *Movers* que se encuentran bloqueando los accesos que simbolizan las entradas a los andenes en el escenario del metro Universidad de Chile se desbloqueen:

