



Universidad de las Ciencias Informáticas

Facultad 5

**MÓDULO DE OBTENCIÓN DE DATOS
TOPOLÓGICOS Y MEDIOAMBIENTALES
PARA EL SISTEMA DE PERCEPCIÓN.**

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autor: Raydel Leyva Blanco

Tutor: Ing. Yenifer del Valle Guevara

Co-Tutor: Ing. Alexey Broche Medina

La Habana

“Año del 53 Aniversario del Triunfo de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste, firmamos la presente a los ____ días del mes de _____ del año_____.

Autor: Raydel Leyva Blanco

Tutora: Ing. Yenifer del Valle Guevara

Co-Tutor: Ing. Alexey Broche Medina

DATOS DE CONTACTO

Generales de la Tutora:

Nombre y apellidos: Ing. Yenifer del Valle Guevara

Especialidad: Ingeniería en Ciencias Informáticas

Años de experiencias: 4 años

Correo electrónico: ydelvalle@uci.cu

Teléfono de contacto: 766-7474

Generales del Co-Tutor:

Nombre y apellidos: Ing. Alexey Broche Medina

Especialidad: Ingeniería en Ciencias Informáticas

Años de experiencias: 2 años

Correo electrónico: abroche@uci.cu

Teléfono de contacto: 837 2261

DEDICATORIA

Va dedicada a:

*Mi mamá, por ser no solo la autora de mis días, sino por luchar
para que yo pudiera cumplir este sueño.*

*A mi abuela Mimi y a mis tías Leticia y Leonela por ser tan
maravillosas conmigo.*

A mi papá, a quien espero superar algún día.

AGRADECIMIENTOS

Gracias a:

*Mi novia Judith, por estar a mi lado durante 8 años y vencer
la distancia y otros demonios.*

A toda mi familia por apoyarme y siempre estar conmigo.

Radamesito y Eydin por ser tan buenos hermanos.

*Todos los que conocí durante estos 5 años y en especial a Fabio
y Maybe por darme su amistad.*

Alberto por macharme en el uso de memoria dinámica.

Mis tutores por todo el apoyo y las ganas de siempre avanzar.

RESUMEN

La percepción es uno de los puntos más importantes en la Inteligencia Artificial (IA), todo sistema inteligente debe tener percepción en algún momento. Puede ser tan simple como que el agente pueda ver el entorno que le rodea, oír todos los sonidos a su alrededor o tan complejo como obtener información del medio.

En este trabajo se propone desarrollar un módulo para la obtención de datos topológicos y medioambientales de un Entorno Virtual (EV) para un Sistema de Percepción (SP). Para lograr su desarrollo se realizó un estudio del SP que se está desarrollando, de las principales técnicas para obtener estos datos en un EV, y de las diferentes vías de guardar la información recolectada.

Como resultado de este trabajo se obtuvo un módulo que brinda las funcionalidades necesarias para que los Agentes Inteligentes (AI) puedan obtener información topológica y medioambiental de su entorno. Además se realizó una herramienta para la edición, en un EV, de los datos topológicos y medioambientales que se cargarán en el módulo. Además se provee un sistema de clases que permite cargar los datos en el SP.

Palabras clave: entornos virtuales, Inteligencia artificial, medioambientales, percepción, topológicos.

RESUMEN	VI
ÍNDICE DE FIGURAS.....	X
ÍNDICE DE TABLAS	XI
INTRODUCCIÓN	1
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.....	4
INTRODUCCIÓN.....	4
1.1. PERCEPCIÓN EN AGENTES INTELIGENTES	4
1.2. SISTEMAS DE PERCEPCIÓN	4
1.2.1. <i>Arquitectura del Sistema de Percepción</i>	5
1.3. PERCEPCIÓN BÁSICA	6
1.3.1. <i>Raycasting</i>	6
1.3.2. <i>Scripting</i>	6
1.4. PERCEPCIÓN AVANZADA.	7
1.4.1. <i>Medioambientales (Environment Data)</i>	7
1.4.2. <i>Datos Topológicos (Topological Data)</i>	9
1.5. TIPOS DE ANÁLISIS TÁCTICOS.	9
1.5.1. <i>Emboscada</i>	9
1.5.2. <i>Puntos fuertes (Stronghold)</i>	10
1.5.3. <i>Los lugares ocultos (Hide)</i>	10
1.5.4. <i>Coberturas (Cover)</i>	10
1.5.5. <i>Sniper</i>	11
1.6. TÉCNICAS PARA AL ANÁLISIS TOPOLÓGICO.....	11
1.6.1. <i>Análisis topológico basado en datos manualmente generados</i>	12
1.6.2. <i>Análisis topológico basado en puntos de referencia</i>	12
1.6.3. <i>Fire Zone Defense (FZD)</i>	14
1.7. MODELOS COGNITIVOS.....	14
1.7.1. <i>Rejilla Regular</i>	15
1.7.2. <i>Mallas de navegación:</i>	15
1.7.3. <i>Arboles cuaternarios o Quadtree</i>	16
CONCLUSIONES DEL CAPÍTULO.....	17
CAPÍTULO II: PROPUESTA DE SOLUCIÓN	18
INTRODUCCIÓN.....	18

2.1.	HERRAMIENTAS.....	18
2.1.1.	<i>Visual Studio</i>	18
2.1.2.	<i>Ogre 3D</i>	18
2.1.3.	<i>Qt</i>	18
2.1.4.	<i>Visual Paradigm</i>	19
2.1.5.	<i>Leguaje y notación de modelado</i>	19
2.2.	DESCRIPCIÓN DE SOLUCIÓN.....	19
2.2.1.	<i>Recolector de datos medioambientales</i>	20
2.2.2.	<i>Recolector de datos topológicos</i>	21
2.2.3.	<i>Herramienta TEND Editor</i>	21
2.2.4.	<i>Representación de la información</i>	22
2.3.	REGLAS DEL NEGOCIO.....	23
2.4.	MODELO DE DOMINO.....	23
2.4.1.	<i>Glosario de términos</i>	24
2.5.	REQUERIMIENTOS DEL SISTEMA.....	24
2.5.1.	<i>Requisitos funcionales</i>	24
2.5.2.	<i>Requisitos no funcionales</i>	25
2.6.	MODELO DE CASOS DE USO (CU) DEL SISTEMA.....	26
2.6.1.	<i>Actores del sistema</i>	26
2.6.2.	<i>Diagrama de CU del sistema</i>	27
2.7.	DESCRIPCIÓN DE LOS CU DEL SISTEMA.....	27
2.7.1.	<i>CU Actualizar datos medioambientales</i>	28
2.7.2.	<i>CU Actualizar datos topológicos</i>	28
2.7.3.	<i>CU Cargar escena</i>	29
2.7.4.	<i>CU Gestionar Elementos Medioambientales</i>	30
2.7.5.	<i>CU Gestionar Puntos de Interés Topológicos</i>	32
2.7.6.	<i>CU Administrar fichero DataExport</i>	34
2.7.7.	<i>CU Administrar fichero Malla de navegación</i>	35
	CONCLUSIONES DEL CAPÍTULO.....	37
	CAPÍTULO III: DISEÑO E IMPLEMENTACIÓN.....	38
	INTRODUCCIÓN.....	38
3.1.	MODELO DEL DISEÑO.....	38
3.1.1.	<i>Diagrama de clases del diseño</i>	38

3.1.2. <i>Diagramas de secuencia del diseño</i>	40
3.2. MODELO DE IMPLEMENTACIÓN.....	45
3.2.1. <i>Diagrama de componentes</i>	45
3.2.2. <i>Diagrama de despliegue</i>	47
3.3. RESULTADOS.....	47
CONCLUSIONES DEL CAPÍTULO.....	48
CONCLUSIONES	49
RECOMENDACIONES	50
BIBLIOGRAFÍA	51
GLOSARIO DE TÉRMINOS	53
ANEXOS	54

ÍNDICE DE FIGURAS

FIGURA 1.1 ARQUITECTURA DEL SISTEMA DE PERCEPCIÓN (6).....	5
FIGURA 1.2 DATOS MEDIOAMBIENTALES COMPUESTOS DE LÍNEAS RECTAS (LAS PAREDES) Y CÍRCULOS (LOS OBSTÁCULOS) (6)	8
FIGURA 1.3 REPRESENTACIÓN DE LOS PUNTOS DE EMBOSCADA (12).....	9
FIGURA 1.4 REPRESENTACIÓN DE UN PUNTO DE STRONGHOLD (12)	10
FIGURA 1.5 REPRESENTACIÓN DE UN PUNTO HIDE (12)	10
FIGURA 1.6 REPRESENTACIÓN DE PUNTOS COVER (12)	11
FIGURA 1.7 REPRESENTACIÓN DE PUNTOS DE SNIPER (12).....	11
FIGURA 1.8 LOS WAYPOINT REPRESENTAN, LA FORMA DEL NIVEL (IZQ.) O ESTÁN DENTRO DEL JUEGO (DER.) (10)	13
FIGURA 1.9 WAYPOINT, CARACTERÍSTICAS: PROPIEDADES LOCALES, AGRUPACIÓN, RELACIÓN Y FOCO (10)	13
FIGURA 1.10 REPRESENTACIÓN DE UNA REJILLA BASADA EN CELDAS CUADRADAS Y HEXAGONALES (8)	15
FIGURA 1.11 MALLA DE NAVEGACIÓN (8)	16
FIGURA 1.12 DIVISIÓN EN SECCIONES (18).....	17
FIGURA 2.1 SISTEMA DE PERCEPCIÓN	20
FIGURA 2.2 MODELO DEL DOMINIO	24
FIGURA 2.3 DIAGRAMA DE CU DEL SISTEMA	27
FIGURA 3.1 DIAGRAMA DE CLASES DEL DISEÑO DEL SISTEMA DE PERCEPCIÓN.....	39
FIGURA 3.2 DIAGRAMA DE CLASES DEL DISEÑO DEL TEND EDITOR.	40
FIGURA 3.3 DIAGRAMA DE SECUENCIA ACTUALIZAR LOS DATOS MEDIOAMBIENTALES	41
FIGURA 3.4 DIAGRAMA DE SECUENCIA ACTUALIZAR LOS DATOS TOPOLÓGICOS.....	41
FIGURA 3.5 DIAGRAMA DE SECUENCIA CARGAR ESCENA	42
FIGURA 3.6 DIAGRAMA DE SECUENCIA GESTIONAR DATOS MEDIOAMBIENTALES. SECCIÓN INSERTAR	42
FIGURA 3.7 DIAGRAMA DE SECUENCIA GESTIONAR DATOS MEDIOAMBIENTALES. SECCIÓN MODIFICAR	43
FIGURA 3.8 DIAGRAMA DE SECUENCIA GESTIONAR DATOS MEDIOAMBIENTALES. SECCIÓN ELIMINAR.....	43
FIGURA 3.9 DIAGRAMA DE SECUENCIA GESTIONAR PUNTOS DE INTERÉS TOPOLÓGICOS. SECCIÓN INSERTAR	43
FIGURA 3.10 DIAGRAMA DE SECUENCIA GESTIONAR PUNTOS DE INTERÉS TOPOLÓGICOS. SECCIÓN MODIFICAR	44
FIGURA 3.11 DIAGRAMA DE SECUENCIA GESTIONAR PUNTOS DE INTERÉS TOPOLÓGICOS. SECCIÓN ELIMINAR.....	44
FIGURA 3.12 DIAGRAMA DE SECUENCIA SALVAR FICHEROS. SECCIÓN NAVMESH.....	45
FIGURA 3.13 DIAGRAMA DE SECUENCIA SALVAR FICHEROS. SECCIÓN DATAEXPORT	45
FIGURA 3.14 DIAGRAMA DE COMPONENTES DEL SISTEMA DE PERCEPCIÓN.....	46
FIGURA 3.15 DIAGRAMA DE COMPONENTES DE LAS CLASES QUE PROVEE LA HERRAMIENTA TEND EDITOR	46
FIGURA 3.16 DIAGRAMA DE DESPLIEGUE	47

ÍNDICE DE TABLAS

TABLA 2.1 <i>ACTORES DEL SISTEMA</i>	26
TABLA 2.2 <i>CU ACTUALIZAR DATOS MEDIOAMBIENTALES</i>	28
TABLA 2.3 <i>CU ACTUALIZAR DATOS TOPOLÓGICOS</i>	29
TABLA 2.4 <i>CU CARGAR ESCENA</i>	30
TABLA 2.5 <i>CU GESTIONAR ELEMENTOS MEDIOAMBIENTALES</i>	32
TABLA 2.6 <i>CU GESTIONAR PUNTOS DE INTERÉS TOPOLÓGICO</i>	33
TABLA 2.7 <i>CU SALVAR FICHEROS</i>	35
TABLA 2.8 <i>CU CARGAR FICHEROS</i>	37

INTRODUCCIÓN

En la actualidad, los EV se están convirtiendo en una herramienta cada vez más habitual en diversos ámbitos como el educativo, el diseño industrial, el entrenamiento militar o las visitas virtuales a museos o edificios históricos. Uno de los requisitos que comienzan a ser comunes consiste en dotar a los personajes virtuales de comportamientos similares a los humanos para crear aplicaciones con mayor sensación de realismo.

En numerosos entornos de simulación resulta necesario tomar algún tipo de decisión por parte de los agentes involucrados. Existen diversas técnicas para la toma de decisiones en EV, Máquinas de Estado Finitas, Grafos Motivacionales, Redes Neuronales y otras, pero para alcanzar una mayor humanización del agente se le adiciona la percepción, para que obtenga más información de su entorno.

La percepción es uno de los puntos más importantes en la IA, todo sistema inteligente debe tener percepción en algún momento. Puede ser tan simple como que el agente pueda ver el entorno que le rodea, oír todos los sonidos a su alrededor o tan complejo como obtener información del medio, incluyendo datos topológicos y medioambientales, ejemplo: irregularidades en el terreno para esconderse, zonas de emboscadas, análisis local para localizar las paredes, obstáculos y curvas.

Los videojuegos desarrollados en el Centro de Informática Industrial (CEDIN) no cuentan con un SP genérico. Resuelven el problema de la percepción de forma ineficiente mediante un sensor, preguntando a cada agente que elemento del entorno percibe. De esta manera no se obtiene la ubicación de las paredes y los obstáculos por el agente y hay que predefinir su posición, dando una baja sensación de realismo al no poder agregar obstáculos o paredes nuevas en tiempo de ejecución. Se carece de un recolector de datos en el entorno de los agentes, recargando a los mismos de funcionalidades que lo hacen ineficiente en su toma de decisiones porque debe conocer donde están todos los objetos de la escena y no con los que está interactuando en su entorno local. No existe ningún tipo de análisis de la topología por lo que no se toma en cuenta para la toma de decisiones.

Basado en las motivaciones anteriormente planteadas se tiene como **problema científico**: ¿Cómo adquirir información topológica y medioambiental de un EV?

El **objetivo general** de la investigación es Desarrollar un módulo que permita la obtención de datos topológicos y medioambientales en EV para el SP. Siendo el **objeto de estudio** los sistemas de percepción en EV y el **campo de acción** la obtención de datos topológicos y medioambientales en EV

para sistemas de percepción. Teniendo como **Idea a Defender** con la obtención de datos topológicos y medioambientales en un EV para que sean manejados por el SP, los AI tendrán un mejor sistema de toma de decisiones en tiempo de ejecución.

Las **Tareas Investigativas** son:

- Diseño teórico de la investigación.
- Propuesta de soluciones técnicas según el estudio del estado del arte.
- Definición de las funcionalidades del módulo para la obtención de datos topológicos y medioambientales en EV para el SP
- Diseño de una herramienta para clasificar los datos topológicos y medioambientales en un EV.
- Implementación de la herramienta para clasificar los datos topológicos y medioambientales en un EV.
- Integración de las clases que brindará la herramienta con el módulo de obtención de datos topológicos y medioambientales en el SP.

El contenido de este documento está estructurado en tres capítulos, así como las correspondientes secciones de las recomendaciones, bibliografía, el glosario de términos y los anexos; organizados de la siguiente forma:

Capítulo I: Fundamentación teórica

Se exponen los principales conceptos relacionados con la obtención de datos topológicos y medioambientales, partiendo del estudio del estado del arte del tema tratado; así como la descripción de la arquitectura que sustenta la base de formación del SP.

Capítulo II: Propuesta y descripción de la solución

Se ofrece una descripción de las características que presentará el sistema como solución al problema científico planteado. También serán definidas la metodología de desarrollo de *software*, el lenguaje de modelado y la herramienta que permitirán dirigir y guiar el proceso de desarrollo para poder llegar a los resultados esperados. Se realiza además la modelación del negocio y la captura de los requisitos funcionales y requisitos no funcionales con la generación de sus respectivos artefactos.

Capítulo III: Diseño e implementación

Este capítulo incluye la generación de los artefactos según la metodología seleccionada para los flujos de trabajo diseño e implementación y los resultados obtenidos al desarrollar la aplicación que permitirá cumplir con los objetivos planteados.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

Introducción

La obtención de la información de los objetos que se encuentran en un EV tiene gran importancia para lograr una mejor toma de decisiones de los AI; convirtiéndose en uno de los desafíos a resolver en la realidad virtual. En el presente capítulo se tratan definiciones como: percepción para AI, específicamente en juegos que utilizan IA. Se analizan los diferentes *softwares* elaborados en el mundo que tratan de simular la percepción humana. Se introducen las técnicas básicas que son usadas para lograr la percepción en EV y los diferentes modelos cognitivos donde se almacena la información.

1.1. Percepción en Agentes Inteligentes

La percepción es un proceso nervioso superior que permite al organismo, a través de los sentidos, recibir, elaborar e interpretar la información proveniente de su entorno. (1)

La percepción se considera uno de los temas más importantes dentro de la robótica y los juegos que son creados bajo la base de la IA. Cada juego dirige la percepción de manera diferente, se conoce que la percepción más sofisticada puede imitar las limitaciones del mundo real. La percepción es mucho más que ver y oír, abarca todas las maneras en que un personaje no controlado por el jugador o *Non Player Character* (NPC) obtiene la información sobre el mundo, incluyendo la topología del entorno y los agentes. Provee de información a los AI para que parezcan lo más real posible. (2)

1.2. Sistemas de Percepción

Un SP, será el encargado de otorgarles a los agentes las habilidades de observar, escuchar, conocer las características topológicas y medioambientales del entorno que los rodea y establecer una comunicación con otros agentes. Se han desarrollado varios sistemas de percepción con diferentes objetivos. En (1), se crea un modelo de percepción para AI inteligentes, otro ejemplo de percepción artificial es *ALIVE system*, creado por *Blumberg* (3), donde un perro animado digitalmente fue equipado con sensores para que interactuara con su medioambiente. *Tu* y *Terzopoulos* en (4) introducen un SP visual para peces, otro ejemplo de software sitúa animales artificiales con un SP en un EV (5).

1.2.1. Arquitectura del Sistema de Percepción

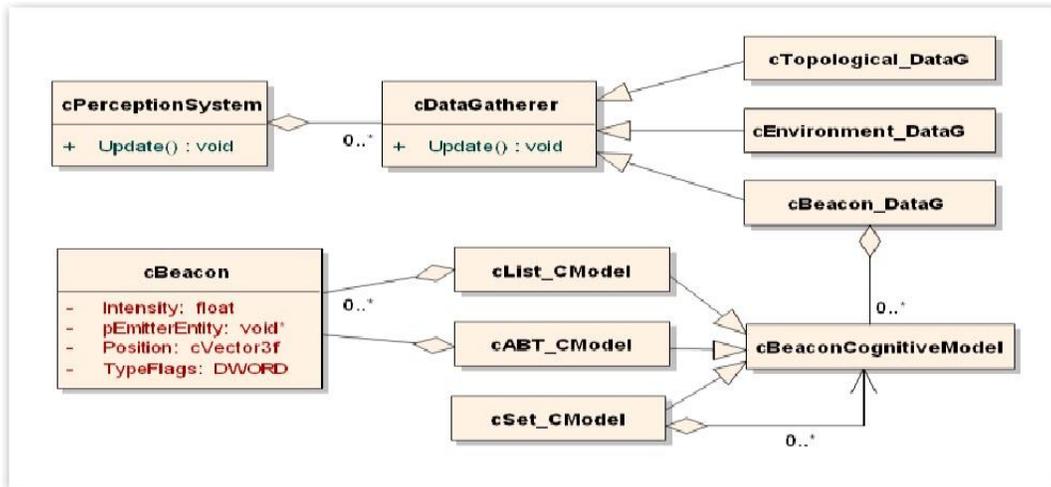


Figura 1.1 Arquitectura del Sistema de Percepción (6)

La arquitectura que presenta el SP con que se está trabajando en el CEDIN y que se muestra en la (figura 1.1) (6) (7) proporciona una serie de elementos a tener en cuenta para su realización:

- Mediante la clase *cPerceptionSystem* se permite la actualización de todas las entidades *cDataGatherer*.
- Mediante la clase *cDataGatherer* se puede almacenar la información de percepción sobre el EV que puede ser detectada por los agentes.
- La información detectada puede ser de cualquier tipo como: datos topológicos, datos ambientales y hasta los acontecimientos que suceden en el mundo mediante las clases *cTopological_DataGatherer*, *cEnvironment_DataGatherer*, *cBeacon_DataGatherer*.
- La clase *cTopological_DataGatherer* es la encargada de buscar la información acerca de las características tácticas del entorno que rodea al agente.
- La clase *cEnvironment_DataGatherer* es la encargada de recolectar una apreciación del entorno que rodea al agente mediante una representación simplificada del mismo basado en formas básicas.
- Por medio de la clase *cBeacon_DataGatherer* los agentes tienen conocimientos de los objetos dinámicos o eventos que son creados y modificados en tiempo de ejecución. Este, a su vez,

está compuesto por uno o más modelos cognitivos que contienen los *beacons* como lista, y malla de navegación, como modelos más especializados.

- En la clase *cBeacon* se pueden almacenar características relacionadas con el evento dinámico y que son de interés para el agente, como pueden ser el identificador del evento dinámico, la entidad emisora, la intensidad de la emisión, entre otras características.

1.3. Percepción básica

Para que los NPC puedan interactuar con su entorno, necesitan conocerlo y para conocerlo necesitan percibirlo. Existen diferentes técnicas para simular la percepción en EV, estas pueden ser bastante simples, empleadas por los desarrolladores para superar las limitaciones de los AI.

1.3.1. Raycasting

Este tipo de percepción es el más utilizado, se utilizan consultas para la percepción mediante el lanzamiento de un rayo para determinar si el agente puede caminar hasta cierto punto, o para localizar objetos específicos en la escena. (8)

Ventajas (9)

- Relativamente fácil de implementar.

Desventajas (9)

- Brinda una pobre percepción del entorno.
- Afecta considerablemente la eficiencia del sistema.
- Se debe encuestar varios rayos para obtener una respuesta a una pregunta como: ¿Dónde me puedo esconder de un enemigo?

1.3.2. Scripting

El *scripting* es otra de las vías que existe para superar la limitante de percepción que presentan los AI. Conduce a lograr una percepción detallada, teniendo en cuenta todas las posibilidades. El *scripting* no es el mejor enfoque para la percepción, debido a que: (9)

- Es tiempo que se consume.
- Interactividad significa que usted no puede prever todo.

- Los escenarios son muy lineales.
- Es complejo hacer guiones para comportamientos que incluyan la coordinación del equipo, búsqueda de posicionamiento.

1.4. Percepción Avanzada.

Con los juegos de próxima generación, estas soluciones para la percepción 3D no funcionarán adecuadamente. El tamaño de los mapas probablemente se multiplicará por cien, así como el número de NPC. (9) Como consecuencia de ello:

- Con más puntos de contacto, el *Raycasting* consume mucho uso del procesador (CPU).
- El tamaño de los mapas probablemente harán la generación manual simplemente imposible.
- Para mapas grandes con muchos puntos de contacto, sólo *Scripting* de alto nivel será aceptable. Guiones muy detallados se convertirán en pesadillas.
- La obtención de datos en grandes mapas necesitará *streaming*¹ para optimizar el consumo de memoria.

Los sistemas modernos de percepción realizan un sofisticado análisis para obtener una percepción precisa de una manera eficiente. En la mayoría de los casos, lo que un agente necesita percibir se puede obtener a partir de dos tipos de percepción:

- Descripción del medio ambiente local.
- Análisis dinámico de la topología para la toma de decisiones.

1.4.1. Medioambientales (Environment Data).

En un juego, se construyen representaciones gráficas y geométricas del ambiente, los NPC deben saber cómo interpretarlas. Los entornos actualmente son complejos y dinámicos, en los cuales se deben considerar las paredes, puertas, salidas, ventanas, esquinas y obstáculos que el NPC puede explorar. (10)

Los videojuegos brindan abundantes situaciones donde la IA debe hacer un reconocimiento del terreno, para:

- Conocer las características claves del terreno.

- Distinguir entre buenas y malas localizaciones.
- Interpretar las localizaciones basadas en el *performance*.
- Incluir el terreno en las decisiones tácticas. (10)

Es posible simplificar estas representaciones automáticamente en formas básicas que ayuden a que el personaje del juego comprenda su ambiente. De esta manera, un mundo 3D complejo puede describirse desde una vista superior (Figura 1.2), con obstáculos delimitados por los círculos y paredes representadas por líneas rectas. (6)

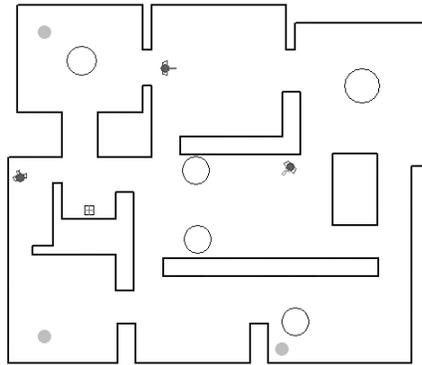


Figura 1.2 Datos Medioambientales compuestos de líneas rectas (las paredes) y círculos (los obstáculos) (6)

Algunos *softwares* como RenderWare A.I² y Kynapse³ proveen funcionalidades para la obtención de los datos del medioambiente local. RenderWare A.I pre-procesa los objetos estáticos del entorno siguiendo dos pasos: (9)

- Extrae los objetos del relevantes del entorno
- Clasifica los objetos extraídos.

Kynapse cuenta con funcionalidades como: (11)

- **3D spatial awareness.** Los NPC pueden determinar las posiciones de los objetos relevantes de su entorno local.

La herramienta *GraphosWorld* fue desarrollada en la UCI en el año 2007, permite la realización de la edición y análisis topográfico de estructuras de navegación. Genera un grafo que se puede emplear en

¹ Este tipo de tecnología permite que se almacenen en un búfer lo que se necesita en ese momento, la mayoría de los contenidos hechos con tecnología "*streaming*" han sido concebidos y diseñados para desechar los datos recién interpretados.

² Es un poderoso IA middleware SDK, usada para el desarrollo de videojuegos.

³ Lidera las soluciones de IA para el desarrollo de videojuegos y simulaciones en tiempo real.

EV, a partir de ahí y a través de algoritmos de IA, se realiza el análisis de los diferentes nodos para obtener datos o valores como el camino más corto a una región o nodo, o el camino más largo. Aunque con este grafo se debería poder encontrar los límites de un terreno, los obstáculos, las puertas etc., la funcionalidad para determinarlas no se desarrolló. (12).

1.4.2. Datos Topológicos (Topological Data).

Para los AI, la percepción es más que simple conocimiento sobre la geometría del mundo. La información sobre las características tácticas del ambiente circundante es a menudo también importante, por ejemplo, lugares de escondite, localización del francotirador, las maneras de entrar en un cuarto, y lugares que pueden emboscarse, entre otros. (6)

A través del análisis topográfico se calculan una serie de datos tácticos que permiten a la computadora actuar y tomar decisiones contra un jugador o un grupo de jugadores. Existen muchos tipos de análisis tácticos, por ejemplo, la emboscada (*ambush*), los puntos fuertes (*stronghold*), los lugares ocultos (*hiden*), los lugares buenos para colocar francotiradores (*sniper*), las regiones que ofrecen defensas naturales o artificiales (*cover*), entre otros. (12)

1.5. Tipos de análisis tácticos.

1.5.1. Emboscada

Los puntos de emboscada son aquellos que presentan una visibilidad amplia de un área determinada y que a su vez brindan defensas naturales o artificiales. Además estos puntos siempre se ubican en lugares por donde es obligatorio pasar para llegar a un objetivo en particular, pero siempre teniendo en cuenta que sean zonas donde no sean descubiertos hasta que se realice la acción planeada.

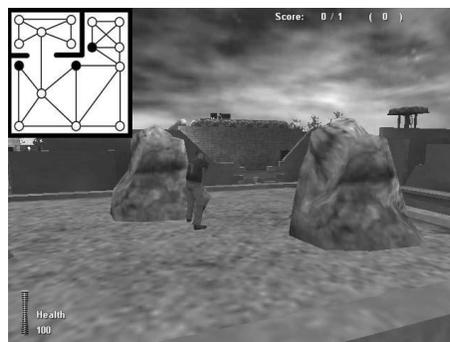


Figura 1.3 Representación de los puntos de emboscada (12)

1.5.2. Puntos fuertes (Stronghold)

Los puntos fuertes son aquellos en donde el jugador se puede resguardar para realizar una defensa óptima de un punto en especial. Estos puntos reúnen las características de ser puntos de difícil acceso, con una visión amplia sobre el área a defender y con los accesos cubiertos. Es una posición defensiva excelente.



Figura 1.4 Representación de un punto de Stronghold (12)

1.5.3. Los lugares ocultos (Hide)

Son los puntos que no son visibles en un área del entorno. Son zonas que ofrecen la posibilidad en el terreno de esconderse del enemigo.



Figura 1.5 Representación de un punto Hide (12)

1.5.4. Coberturas (Cover)

Los puntos de defensas son aquellos lugares que ofrecen protección durante una sorpresa y que puedan ser usados para resguardarse durante un ataque. Son zonas que presentan defensas naturales o artificiales.

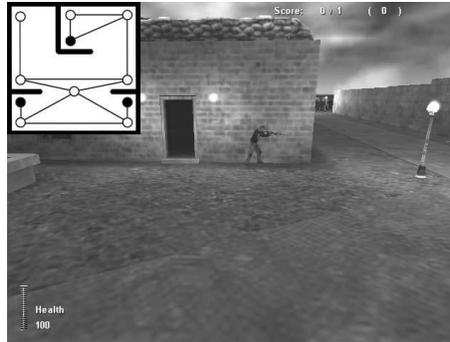


Figura 1.6 Representación de puntos Cover (12)

1.5.5. Sniper

Los puntos más óptimos para colocar francotiradores en el terreno, son aquellos lugares que no se noten mucho, que pasen desapercibidos, difíciles de alcanzar por parte de los objetivos del francotirador, puntos que se encuentren en un lugar que sea poco visible. Lugares donde sea difícil de sorprender al francotirador, que la visibilidad sea la mayor posible de una región determinada, que exista una libertad de movimiento, que exista un lugar donde el francotirador tenga una posición defensiva excelente y que estén lejos de la acción.



Figura 1.7 Representación de puntos de Sniper (12)

1.6. Técnicas para al análisis Topológico.

El análisis topológico en el mundo virtual se basa fundamentalmente en el uso de algoritmos muy sofisticados y eficientes que permiten generar decisiones tácticas, y esto se realiza a través de los grafos de navegación que hay en el mundo.

Como los grafos se usan principalmente para la navegación, contienen información acerca de la relación entre las diferentes posiciones en el entorno. Esta información puede ser explotada para

calcular de forma eficiente el valor estratégico de una posición específica en el terreno. Este tipo de información puede revelar la información estratégica que se tiene de un nodo determinado en la red del grafo, donde se puede, por ejemplo, analizar cuál es el nodo principal que permite el acceso a una región determinada, los nodos que tienen una buena posición defensiva, o para realizar una emboscada.

1.6.1. Análisis topológico basado en datos manualmente generados.

Los datos son generados manualmente por los desarrolladores, que son los encargados de añadir los datos topológicos en el mundo virtual, cuando un NPC desea localizar un Punto de Interés Topológico (PIT) se lo pide al sistema, el cuál ahorra tiempo de cálculo con esta técnica, al tener la posición de todos los PIT. Un ejemplo simple constituye el caso de realizar una emboscada, se entrarán manualmente datos topológicos como: la posición de escondites, la activación de zonas, etc.

Sin embargo no siempre constituyen una solución aceptable porque: (9)

- La generación de datos manual no es exhaustiva.
- La generación de datos manual consume mucho tiempo.
- Los datos generados pueden no ser compatibles con el motor de juego de colisión y los modelos de circulación.

Aunque tienen varias ventajas como:

- Un PIT es fácil de crear y de usar si es creado manualmente.
- Para aplicaciones o juegos donde las escenas no son tan grandes y complicadas es más sencillo insertar un punto manualmente.

1.6.2. Análisis topológico basado en puntos de referencia.

Una forma popular de representar el terreno son los puntos de referencia (*waypoint*). Estos puntos de referencia representan la parte más relevante del terreno: el terreno que sea accesible para los actores del juego. Cada punto de referencia es una muestra, una aproximación, de lo que hay a su alrededor. (10) Los *waypoint* sirven como nodos en el grafo para representar todos los caminos en los cuales un NPC puede navegar en el entorno. Las conexiones entre los nodos del grafo son generadas automáticamente mediante pre-procesamiento o colocadas manualmente por el diseñador. (13)

La característica de los *waypoint* es que describen el terreno accesible a los actores que interactúan con este, el movimiento válido y las propiedades del entorno local como el acceso a lugares dentro de la escena. Las razones por las cuales los *waypoint* son tan atractivos dentro de la IA es que son fáciles

de crear y de usar. Los *waypoint* se usan para hacer la navegación en la escena, para marcar el terreno accesible a la IA y a los jugadores, además para describir y representar las fronteras, expresar los caminos y el tiempo de viaje por estos. (10)

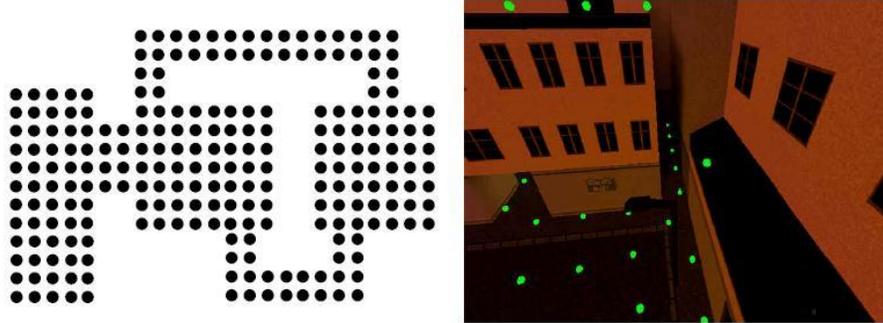


Figura 1.8 Los *waypoint* representan, la forma del nivel (izq.) o están dentro del juego (der.) (10)

Los *waypoint* tienen cuatro características que deben ser tomadas en cuenta. (10)

- **Propiedades locales**, como el nivel de luz, la presencia de una puerta o el tipo de movimiento requerido (agacharse, nadar o usar una escalera).
- **Agrupación**, para representar un terreno más grande (Por lo general un grupo de *waypoint*). Por ejemplo, puede ser parte de una habitación, una calle, o un techo. Tanto las propiedades locales y la agrupación son propiedades no direccionales.
- **Las relaciones con otros *waypoint***, si hay una línea de fuego válida, cuán lejos está un punto del otro.
- La concentración de las relaciones en un sector, **foco**. El jugador o la IA es capaz de concentrarse en una sola dirección y ver toda la actividad visible en esa dirección, sin tener que preocuparse por los ataques de otras direcciones.

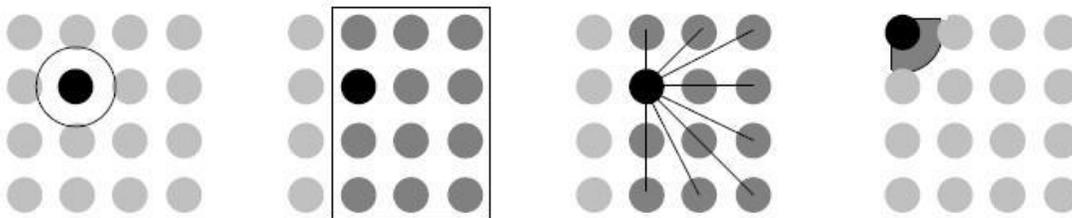


Figura 1.9 *Waypoint*, características: propiedades locales, agrupación, relación y foco (10)

Para calcular una calificación ofensiva o defensiva para un determinado punto de control, se tiene que analizar cada característica táctica aplicable en función de las propiedades del *waypoint*. Por ejemplo, una característica importante de una buena posición de ataque es que prevea un rápido movimiento

en esa posición. El agua no permite un movimiento rápido a nivel local. Un *waypoint* con una propiedad local tipo "agua" debe tener una baja calificación en movimiento rápido. (10)

1.6.3. Fire Zone Defense (FZD)

Se define la zona de fuego (*fire zone*) como el área de terreno que se le asigna a una unidad para defenderse del fuego directo de sus enemigos. La tarea es encontrar un conjunto de lugares para las entidades de la unidad que maximice la zona de fuego de la unidad dentro de campo sin la exposición excesiva al fuego enemigo. (14)

El FZD es un algoritmo creado para resolver el problema de seleccionar la localización de las entidades de una unidad para la defensa efectiva de una posición de combate, considerado la geometría del terreno en términos de cobertura, ocultamiento, y visibilidad. (14)

El algoritmo tiene cuatro pasos: (15)

1. Mediante el análisis de los polígonos de terreno, identificar la ubicación, dentro del área de despliegue de la unidad, que da cobertura y ocultamiento.
2. Del mismo modo, identificar los lugares dentro de la zona de fuego que necesitan visión con el fin de cubrir la zona de fuego.
3. Definir un grafo, con los vértices de los dos conjuntos, y los bordes correspondientes a las líneas de desbloqueo de la vista. El uso de un algoritmo de búsqueda en un grafo, seleccionar un subconjunto de los lugares de cubierta y ocultación de las en
4. Emitir órdenes de movimiento a las entidades de la unidad a los lugares seleccionados.

El algoritmo FZD es muy eficiente para la simulación de entornos de combates, y es utilizado en el entrenamiento militar. Requiere un alto costo computacional por la gran cantidad de variables que toma en cuenta para determinar una posición óptima, además que solo puede ser aplicado en proyectos o juegos FPS⁴.

1.7. Modelos Cognitivos

Los modelos cognitivos se usan en la IA para representar el mundo virtual. En este modelo se incluyen solamente los detalles necesarios como los obstáculos geométricos en el entorno (puertas, ventanas, enemigos). Además almacena el estado de la información como las puertas cerradas, caminos bloqueados o el tipo de enemigo que se está moviendo.

⁴ *First Player Shooter*, juegos de disparar en primera persona.

Constituyen estructuras de datos que se utilizan para almacenar la información. El modelo cognitivo más simple es una Lista, pero existen modelos más especializados como el Octree, y la Rejilla Regular. El uso de diferentes modelos cognitivos en el SP a desarrollar, permite la representación del mundo para manejar la visibilidad y la colisión.

1.7.1. Rejilla Regular

La rejilla regular se utiliza mucho en juegos con sistemas en tiempo real, a veces tiene un complejo ambiente basado en cuadrados o polígonos (triángulos, pentágonos o hexágonos). Esto por consiguiente es racional en el diseño de grafos de navegación. Este particular método reduce nuestra área de búsqueda a una simple matriz bidimensional. Cada elemento de la matriz representa uno de los cuadrados de la rejilla, y su estado se almacena como transitable o intransitable. Las rejillas son más útiles en ambientes 2D, no podrían ser usadas en juegos 3D sin alguna modificación. El valor presente en cada celda representa árboles, murallas, enemigos, el jugador, casas, terrenos entre otras cosas. Por estos valores la IA puede planear donde moverse, donde localizar enemigos o cómo evitar obstáculos. (8)

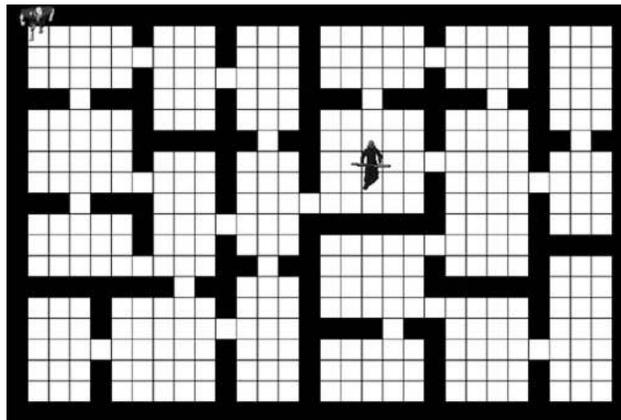


Figura 1.10 Representación de una rejilla basada en celdas cuadradas y hexagonales (8)

1.7.2. Mallas de navegación:

Uno de los modelos más usados para representar mundos 3D son las mallas de navegación, las cuales son conocidas como *NavMesh*. Una malla de navegación es un grupo de polígonos convexos que describen la superficie del ambiente 3D. Este es un simple y altamente intuitivo modelo cognitivo donde los caracteres de la IA realizan un plan que pueden usar para la navegación y la búsqueda de caminos en el mundo. (8)

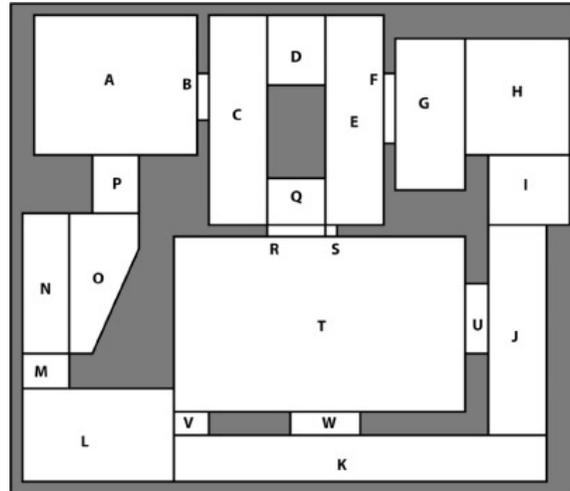


Figura 1.11 Malla de navegación (8)

Esta técnica proporciona una rápida y eficaz alternativa a otras formas de discretización, siendo esta la más utilizada en los últimos años para el desarrollo de EV, donde se hace necesario el desplazamiento de agentes en dicho entorno, ejemplo: Juegos de Estrategia, Simuladores de multitudes, etc. A diferencia de otras técnicas brinda un completo conocimiento sobre las zonas navegables, pudiéndose tomar ventaja sobre esta información para determinar el más corto y natural de los caminos posibles, así los agentes del juego son capaces de evitar obstáculos que bloquean su camino o incluso seleccionar una ruta alternativa, mostrando con esto un comportamiento autónomo. Cada arista común tiene un punto de acceso hacia un polígono adyacente. (16)

En el Centro de Informática Industrial (CEDIN), se desarrolló el software *NavMesh Creator* (17) que permite crear y editar mallas de navegación.

1.7.3. Árboles cuaternarios o Quadtree

Los *quadtrees* son una forma de representar el dominio de manera distinta a otras estructuras de datos. El dominio es rodeado por el cuadrante más pequeño que lo rodea, ver (Figura 1.12). Todo cuadrado se divide en 4 cada vez que es requerido, en forma natural aparece un árbol donde cada nodo padre tiene cuatro nodos hijos. (18)

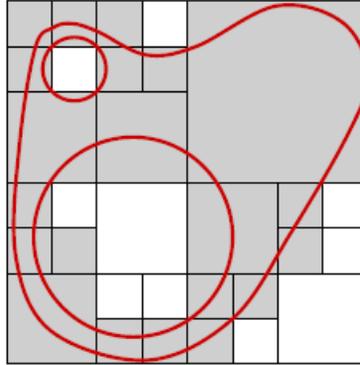


Figura 1.12 División en secciones (18)

Los árboles cuaternarios constituyen una variante jerárquica de la enumeración de ocupación espacial, diseñada para optimizar los requisitos de almacenamiento de este método. Para formar estos árboles cuaternarios se divide sucesivamente un plano bidimensional en ambas direcciones (X, Y) para formar cuadrantes. (18)

Conclusiones del Capítulo.

En este capítulo se conoció que es la percepción para AI, y se hizo un estudio del SP propuesto. Se analizaron las distintas técnicas de percepción, tanto básicas como avanzadas, para determinar cuál adoptar para lograr el objetivo de la investigación. Se hizo un estudio de las técnicas para la obtención de datos topológicos y medioambientales y los puntos tácticos que se pueden incluir en la herramienta a desarrollar. Se tuvieron presente también los distintos modelos cognitivos, en los cuales se puede almacenar la información, para luego ser utilizada en el SP.

CAPÍTULO II: PROPUESTA DE SOLUCIÓN

Introducción

En el presente capítulo se proponen las soluciones técnicas para obtener datos topológicos y medioambientales en EV por el SP. Se abordarán temas relacionados con la forma en que se obtendrán datos del entorno. Serán descritas las características que tendrá la aplicación a desarrollar, las principales funciones con que contará la aplicación, las herramientas y los lenguajes de modelado también formarán parte de los epígrafes de este capítulo.

2.1. Herramientas.

2.1.1. Visual Studio.

Se escogió Visual Studio como Entorno de Desarrollo Integrado (IDE) primeramente por cuestión de ajuste al módulo al cual se acoplará la aplicación. Además soporta el lenguaje de programación Visual C++, proporciona un modelo extensible para añadir nuevos proyectos y la posibilidad de adjuntarle motores gráficos como Ogre. Permite el trabajo con el *framework* QT, así como la selección de este IDE estuvo basada en otras características como son Inteligencia, Depuración, y Organización. (19)

2.1.2. Ogre 3D.

Ogre (*Object-Oriented Graphics Rendering Engine*) es un motor gráfico desarrollado en C++, multiplataforma, libre y diseñado para hacer más fácil e intuitiva la creación de juegos y aplicaciones de realidad virtual, haciendo uso de capacidades 3D presentes dentro del hardware. Presenta una biblioteca de clases basada en *Direct 3D* y *OpenGL* que contiene un conjunto de clases que permiten la generación de mundos virtuales. Además Ogre permite que las escenas se dividan en “páginas” (pedazos de un mapa muy grande se procesan uno a la vez, permitiendo cargar escenas muy grandes). Cada página puede tener sus propios *HeightMaps*. Su renderizado está basado en Octree, BSP (*Binary Space Partition*) y portales. Por lo antes expuesto se selecciona Ogre como motor gráfico a emplear. (20)

2.1.3. Qt.

Se empleará Qt para el diseño de las interfaces de la aplicación ya que es un *framework* de desarrollo multiplataforma con una biblioteca de más de 400 clases, las cuales encapsulan toda la infraestructura necesaria para desarrollar aplicaciones robustas. Además posee una utilidad (QT *Designer*) para la

rápida construcción de capas de interfaz gráfica de usuario con apariencia y aspecto nativos de las plataformas soportadas. Está separado en 13 módulos y la API que brinda incluye funciones para conectividad a bases de datos, programación en red, integración con gráficos 3D, desarrollo multi-hilos y lectura/escritura de ficheros XML. (21)

2.1.4. Visual Paradigm

Se escoge Visual Paradigm como herramienta de modelado ya que es multiplataforma y muy potente. Soporta el ciclo de vida completo del desarrollo de software, ayuda a una rápida construcción de aplicaciones de calidad y a obtener un menor coste en las mismas. Permite dibujar todos los tipos de diagramas de clases, obtener código inverso, generar código desde diagramas así como la documentación que se necesite de cada uno de ellos.

2.1.5. Leguaje y notación de modelado.

Se escoge como lenguaje de modelado al Lenguaje Unificado de Modelado (UML), ya que es utilizado para visualizar, especificar, construir y documentar los artefactos de un sistema obtenidos a partir de una metodología determinada. Con la utilización del lenguaje UML se puede representar todas las fases de un proyecto informático.

El lenguaje de programación escogido para llevar a cabo la realización de esta investigación fue el C++ estándar, porque además de ser un lenguaje multiplataforma es el lenguaje mayormente utilizado para la realización de las aplicaciones de Realidad Virtual, pues en este tipo de aplicaciones se manejan grandes volúmenes de datos y el mismo permite un uso óptimo de la memoria y del CPU de la computadora donde se instalará la aplicación desarrollada.

2.2. Descripción de Solución.

En la arquitectura propuesta del SP a partir de los tipos de recolectores de datos que se usarán, se definen las clases *Topological_DG* y *Environment_DG* como las manejadoras de los datos topológicos y medioambientales (Figura 2.1). Estas clases son las que se desarrollarán para crear el módulo de obtención de datos topológicos y medioambientales. Se le agregarán las funcionalidades necesarias para que el módulo permita a los recolectores de datos obtener la información que necesita el AI y proporcionársela. El AI es quien solicita la información en dependencia del tipo de dato que necesite.

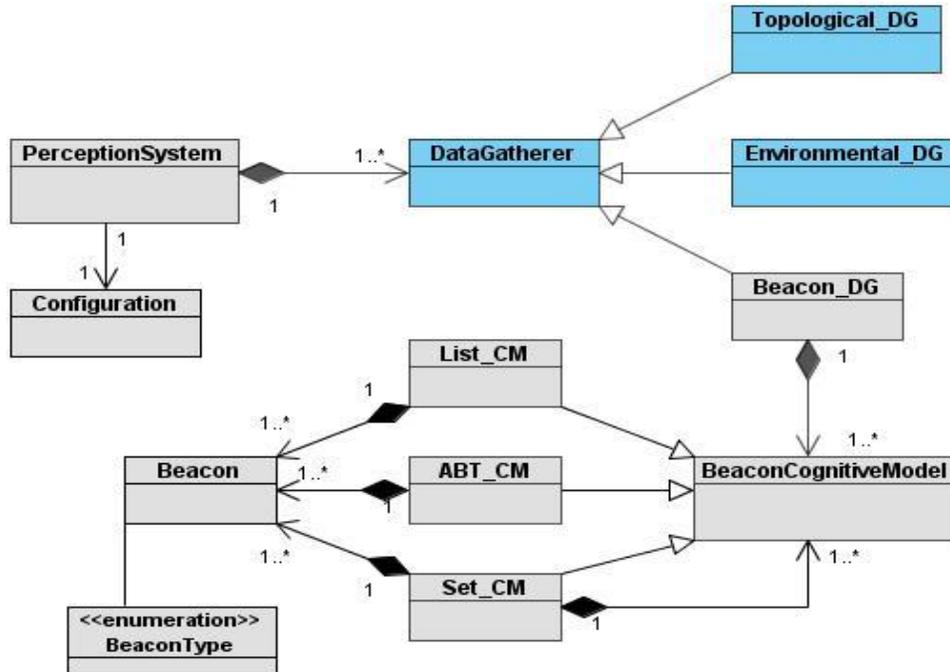


Figura 2.1 Sistema de percepción

2.2.1. Recolector de datos medioambientales.

El recolector de datos medioambientales del módulo permitirá obtener la información de los datos medioambientales de los modelos cognitivos y actualizar la información que es solicitada por el AI mediante los sensores. El recolector obtendrá los siguientes tipos de datos: paredes, obstáculos y puertas, cada uno tendrá sus características, además de las propiedades por defecto se podrá adicionar otras que el desarrollador defina.

- *Wall* (Paredes)
 - *Name*: nombre del objeto en la escena.
 - *From - To*: posición donde comienza y posición donde termina la pared.
 - *High*: altura de la pared.
 - *Sound Propertys*: propiedades de sonido, necesarias para el módulo de percepción sonora.
- *Obstacles* (Obstáculos)
 - *Name*: nombre del objeto en la escena.
 - *Position*: posición del objeto en la escena.

- *Area*: área que ocupa el mismo
- *High*: altura del obstáculo.
- *Door* (puertas o salidas)
 - *Name*: nombre del objeto en la escena.
 - *Position*: posición del objeto en la escena.
 - *isOpen*: para determinar si la puerta está abierta o cerrada.

2.2.2. Recolector de datos topológicos.

El recolector de datos topológicos del módulo permitirá obtener la información de los datos topológicos de los modelos cognitivos y actualizar la información que es solicitada por el AI mediante los sensores. Los tipos de datos topológicos que se tendrán en cuenta en el módulo a desarrollar serán:

- *HidingPos* (Posiciones de escondite).
- *AmbushPos* (Posiciones de emboscadas).
- *StrongholdPos* (Posiciones fuertes para la defensa de una zona).
- *SniperPos* (Posiciones para francotiradores).
- *CornerPos* (Esquinas).

Estos tipos de datos son específicos para juegos FPS, por lo que el módulo también podrá obtener otros tipos de datos más generales para que pueda ser adaptable a otro tipo de aplicaciones que necesite realizar un análisis táctico.

2.2.3. Herramienta *TEnD Editor*.

Para el funcionamiento de los recolectores de datos del módulo debe haber un medio mediante el cual se pueda salvar la información topológica y medioambiental del EV. Luego esta información será obtenida por dichos recolectores. Como estos datos son estáticos es recomendable procesarlos en tiempo de diseño para ahorrar uso del CPU. Para esto se propone el desarrollo de una herramienta la cual brindará al usuario la posibilidad de cargar una escena y determinar los objetos que serán considerados como medioambientales, así como la localización de los puntos de interés topológicos.

Esta herramienta tendrá como base el editor *NavMesh Creator* desarrollado en el CEDIN, el cual permite cargar un entorno 3D y exportar toda la información del entorno en una malla de navegación.

Capítulo II: Propuesta y descripción de la solución.

Para realizar una reutilización de código, al editor *NavMesh Creator* se le diseñará una nueva interfaz visual para permitir un rápido manejo del mismo y se le agregarán las funcionalidades necesarias para posibilitar una vía sencilla de seleccionar e insertar datos, visualizando en todo momento los cambios ocurridos. El resultado final será salvado dando varias opciones de guardado al usuario, y se podrá cargar en el SP para que los recolectores de datos le brinden la información necesaria a los agentes.

Para seleccionar los datos se aplicará la técnica de Datos manualmente generados por las siguientes ventajas:

- Un punto de interés topológico es fácil de crear y de usar si es creado manualmente.
- Si el EV no es muy grande y complejo es fácil señalar manualmente cuales objetos serán datos medioambientales.
- Para aplicaciones o juegos donde las escenas no son tan grandes y complicadas es más sencillo insertar un punto manualmente.
- El desarrollador puede mediante el editor, seleccionar las localizaciones estratégicas y señalarlas manualmente.
- La herramienta brindará la opción de agregar nuevos tipos de puntos topológicos, además de los predefinidos, así como modificar las características de los datos medioambientales.

2.2.4. Representación de la información.

Para almacenar toda la información referente a los datos topológicos y medioambientales, se decidió utilizar dos vías, una mediante una malla de navegación y la otra exportando a un fichero que se denominará *DataExport*.

Malla de navegación.

Se escogió la malla de navegación pues:

- Brindan conocimiento de la geometría del entorno.
- Brindan conocimiento de las zonas transitables.
- Los productos de RV que se están desarrollando en el centro CEDIN utilizan malla de navegación.

Como se utiliza de base el editor *NavMesh Creator*, el cual provee un sistema de clases para la generación de malla de navegación, se le adicionaron las funcionalidades necesarias para que la malla almacenara la información referente a los datos topológicos y medioambientales.

Cada polígono asociado a un nodo de la malla de navegación tiene una lista de vértices, una lista de aristas, una lista de aristas comunes y se le adiciona una lista de puntos de interés topológico y una lista de datos medioambientales. La información de la malla de navegación será posteriormente exportada en un fichero de texto para su empleo en cualquier EV.

Fichero *DataExport*

La información se guardará en un fichero el cual podrá ser cargado en el modelo cognitivo que defina el desarrollador. Tendrá la siguiente estructura:

<Environment>

< Tipo de dato medioambiental>

 <Características>

<Topological>

< Tipo de dato topológico>

 <Características>

2.3. Reglas del negocio.

A continuación se presentan las reglas de negocio (RN) que deben satisfacerse para el correcto funcionamiento del producto “Editor de Datos Topológicos y Medioambientales” (*TEnD Editor*).

- La escena con la que se desea trabajar debe ser .scene.
- La escena tiene que tener las upAxis en el eje “Y”, o sea, mirar hacia arriba en “Y”.

2.4. Modelo de domino.

A partir del entendimiento del problema se obtiene el siguiente modelo de dominio representado a continuación, que no es más que un acercamiento a la solución propuesta, donde se modelan los principales conceptos así como sus relaciones.

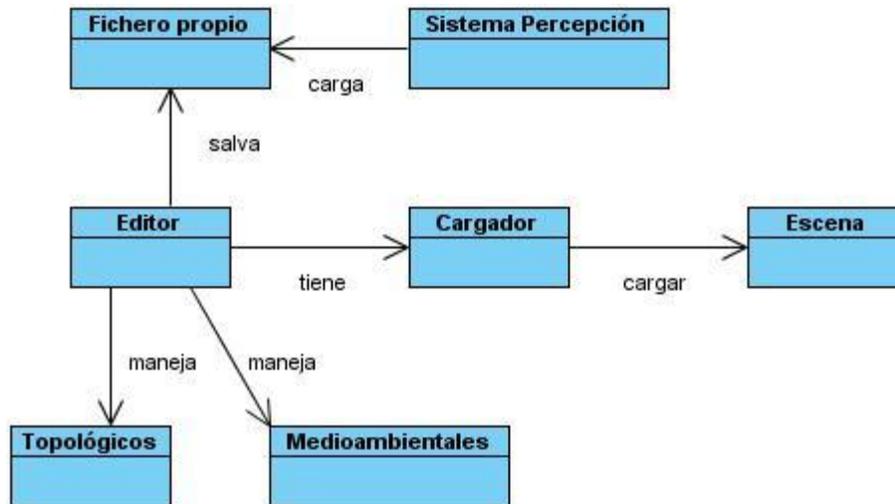


Figura 2.2 Modelo del dominio

2.4.1. Glosario de términos.

Sistema percepción: Carga el fichero que genera el Editor para obtener los datos topológicos y medioambientales.

Editor: Controlador de todo el sistema.

Cargador: Carga un fichero .scene, el cual contiene un modelo de la geometría del mundo.

Fichero: Fichero .scene, el cual contiene un modelo de la geometría del mundo.

Fichero propio: Fichero donde se salva toda la información

Topológicos: Controlador de la información topológica en la escena.

Medioambientales: Controlador de la información medioambiental en la escena.

2.5. Requerimientos del sistema.

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir y los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Teniendo en cuenta las necesidades del cliente se definen los siguientes requisitos funcionales y no funcionales.

2.5.1. Requisitos funcionales.

RF1 Iniciar la aplicación.

RF1.1 Inicializar OGRE.

- RF1.2 Inicializar Qt.
- RF1.3 Crear escenario de trabajo (Interfaz).
- RF2 Destruir recursos.
- RF3 Crear un nuevo proyecto.
- RF4 Cargar fichero DataExport.
- RF5 Cargar Malla de navegación.
- RF6 Cargar escena.
- RF7 Crear datos medioambientales.
- RF8 Modificar datos medioambientales.
- RF9 Eliminarlos datos medioambientales.
- RF10 Insertar puntos de interés topológicos.
- RF11 Eliminar puntos de interés topológicos.
- RF12 Editar puntos de interés.
- RF13 Salvar en malla de navegación.
- RF14 Salvar en fichero DataExport.
- RF15 Salir de la aplicación.
- RF15.1 Verificar que se hayan salvado los cambios efectuados.
- RF16 Obtener datos medioambientales en el módulo.
- RF17 Actualizar datos medioambientales en el módulo.
- RF18 Obtener datos topológicos en el módulo.
- RF19 Actualizar datos topológicos en el módulo.

2.5.2. Requisitos no funcionales

Usabilidad

La aplicación está concebida para ser reutilizada por aplicaciones finales, además los usuarios de la misma deberán tener conocimiento sobre el tema en cuestión.

Diseño e implementación

Se hará uso del lenguaje de programación C++ y se regirá por la filosofía de Programación Orientada a Objeto (POO).

Se empleará el *framework* de desarrollo QT SDK by Nokia v2009 para el diseño de las interfaces gráficas.

Apariencia e interface de usuario

Se debe proporcionar una interface gráfica con botones y ventanas amigables de manera que facilite la interacción de los usuarios con la misma.

Hardware

PC Pentium 4, 2.4 GHz, 512 MB de memoria RAM.

Requerimientos de soporte

El sistema que se desarrollará, en su finalidad, deberá tener compatibilidad con el sistema operativo Windows XP/Seven.

2.6. Modelo de Casos de Uso (CU) del sistema.

2.6.1. Actores del sistema.

Los actores de un sistema son agentes externos o roles que las personas o usuarios desempeñan cuando interactúan con el software que se realizará.

Actores	Descripción
Diseñador	Es quien hará uso de la herramienta <i>TEnD Editor</i> para la selección de los datos topológicos y medioambientales en un EV que serán guardados en un fichero que será cargado en el SP.
Aplicación	Es la que inicializa el SP para que se ejecuten las funcionalidades para obtener los datos topológicos y medioambientales del fichero que se salvó con la herramienta.

Tabla 2.1 Actores del sistema

2.6.2. Diagrama de CU del sistema

El diagrama de CU del sistema define las relaciones entre los actores y los diferentes CU existentes. Para la realización de esta investigación fueron definidas un conjunto de acciones que deben ser ejecutadas por el actor del sistema y que desencadenan un conjunto de operaciones. Las interrelaciones entre las acciones y el actor del sistema son agrupadas en el diagrama de CU del sistema que se muestra a continuación.

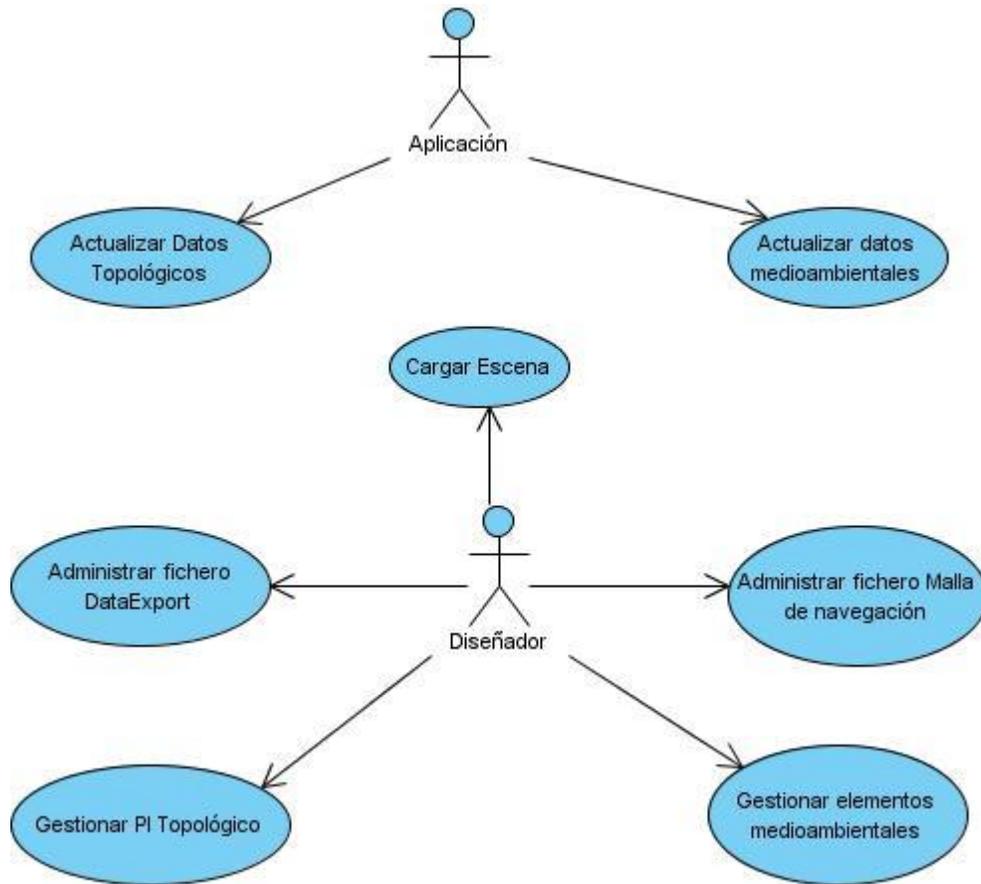


Figura 2.3 Diagrama de CU del sistema

2.7. Descripción de los CU del sistema

Los CU son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario, estableciendo un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema.

CU del Módulo de obtención de datos topológicos y medioambientales en el Sistema de percepción.

2.7.1. CU Actualizar datos medioambientales

CU	Actualizar datos medioambientales	
Actores	Aplicación	
Resumen	Se actualizan los datos medioambientales de un agente dada su posición.	
Precondiciones	Se debe entrar la posición del agente.	
Referencias	RF16,RF17	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1. La aplicación pasa la posición del agente que solicita los datos.	1.1 El sistema hace una búsqueda dada la posición. 1.2 Devuelve una lista de los datos medioambientales. 1.3 Actualiza los datos del agente.	
Poscondiciones		

Tabla 2.2 *CU Actualizar datos medioambientales*

2.7.2. CU Actualizar datos topológicos

CU	Actualizar datos topológicos
-----------	------------------------------

Capítulo II: Propuesta y descripción de la solución.

Actores	Aplicación
Resumen	Se actualizan los datos topológicos de un agente dada su posición.
Precondiciones	Se debe entrar la posición del agente.
Referencias	RF18,RF19
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1. La aplicación pasa la posición del agente que solicita los datos.	1.1 El sistema hace una búsqueda dada la posición. 1.2 Devuelve una lista de los datos topológicos. 1.3 Actualiza los datos del agente.
Poscondiciones	

Tabla 2.3 CU Actualizar datos topológicos

CU Herramienta *TEnD Editor*.

2.7.3. CU Cargar escena

CU	Cargar escena
Actores	Diseñador
Resumen	Se cargar un fichero .scene
Precondiciones	

Capítulo II: Propuesta y descripción de la solución.

Referencias	RF6
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1. Selecciona opción "Load Scene". 2. Selecciona el fichero .scene deseado y pulsa botón "Open".	1.1 Muestra cuadro de diálogo con filtro para la selección de un fichero .scene. 2.1 Muestra la escena en el viewports.
Poscondiciones	

Tabla 2.4 CU Cargar escena

2.7.4. CU Gestionar Elementos Medioambientales

CU	Gestionar Elementos Medioambientales (EM).
Actores	Diseñador
Resumen	El caso de uso comienza cuando el diseñador escoge la opción de crear, modificar o eliminar algún EM.
Precondiciones	Tanto para los procesos de modificar y eliminar el EM debe existir dentro del sistema.
Referencias	RF6,RF7,RF8, RF9
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio

Capítulo II: Propuesta y descripción de la solución.

<p>a) Si el actor desea crear un EM ir a la sección “crear”.</p> <p>b) Si el actor desea modificar un EM ir a la sección “modificar”.</p> <p>d) Si el actor desea eliminar un EM ir a la sección “eliminar”.</p>	
Sección “ Crear”	
Acción del Actor	Respuesta del Negocio
<p>1. Selecciona “Terrain Analysis”, escogiendo la opción “Build Type”.</p> <p>2. Selecciona el elemento que desea convertir a EM.</p>	<p>1.1. El sistema muestra una ventana con todos los elementos de la escena.</p> <p>2.1. El sistema adiciona los elementos a la lista de EM.</p>
Sección “ Modificar”	
Acción del Actor	Respuesta del Negocio
<p>1. Selecciona el elemento en la escena.</p> <p>2. El diseñador modifica las propiedades del elemento seleccionado.</p>	<p>1.1. El sistema muestra en el inspector de propiedades las propiedades del elemento.</p> <p>2.1. El sistema actualiza las propiedades modificadas del EM.</p>
Sección “Eliminar ”	
Acción del Actor	Respuesta del Negocio
<p>1. Selecciona el elemento en la escena.</p>	<p>1.1. El sistema muestra en el inspector de propiedades las propiedades del elemento.</p>

Capítulo II: Propuesta y descripción de la solución.

2. El diseñador modifica la propiedad “Type” del elemento seleccionado colocándola en “None”.	2.1. El sistema elimina el elemento de la lista de EM.
Poscondiciones	

Tabla 2.5 *CU Gestionar elementos medioambientales*

2.7.5. CU Gestionar Puntos de Interés Topológicos

CU	Gestionar Puntos de Interés Topológicos (PIT).	
Actores	Diseñador	
Resumen	El caso de uso comienza cuando el diseñador escoge la opción de crear, modificar o eliminar algún PIT.	
Precondiciones	Tanto para los procesos de modificar y eliminar el PIT debe existir dentro del sistema.	
Referencias	RF6,RF10,RF11, RF12	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
a) Si el actor desea crear un PIT ir a la sección “crear”. b) Si el actor desea modificar un PIT ir a la sección “modificar”. d) Si el actor desea eliminar un PIT ir a la sección “eliminar”.		
Sección “ Crear”		

Capítulo II: Propuesta y descripción de la solución.

Acción del Actor	Respuesta del Negocio
<ol style="list-style-type: none"> 1. Selecciona opción “Insert” y presiona “Interesting Point”. 2. Ubica en la escena un punto de interés. 3. Ingresa la información solicitada. 	<ol style="list-style-type: none"> 2.1 Muestra ventana para ingresar la información del punto. 3.1 Verifica si el punto es único. En caso negativo muestra un cartel de error. 3.2 Muestra los datos ingresados en el inspector de propiedades.
Sección “ Modificar”	
Acción del Actor	Respuesta del Negocio
<ol style="list-style-type: none"> 1. Selecciona el PIT. 2. El diseñador modifica las propiedades del PIT seleccionado. 	<ol style="list-style-type: none"> 1.1 El sistema muestra en el inspector de propiedades las propiedades del PIT. 2.1 El sistema actualiza las propiedades modificadas.
Sección “Eliminar ”	
Acción del Actor	Respuesta del Negocio
<ol style="list-style-type: none"> 1. Selecciona el PIT. 2. Con el PIT seleccionado, el usuario da clic en la opción “delete interesting point”. 	<ol style="list-style-type: none"> 2.1. El sistema elimina el PIT.
Poscondiciones	

Tabla 2.6 *CU Gestionar Puntos de interés topológico*

2.7.6. CU Administrar fichero DataExport

CU	Administrar fichero DataExport.	
Actores	Diseñador	
Resumen	Se salva un fichero con la extensión txt.	
Precondiciones	Estén creados datos MA o PIT para el proceso Salvar. Para el proceso de Cargar la escena debe estar cargada	
Referencias	RF4, RF6,RF7,RF8, RF9,RF10,RF11,RF12, RF14	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
<p>a) Si el actor desea salvar en el fichero DataExport ir a la sección “Salvar”.</p> <p>b) Si el actor desea cargar el fichero DataExport ir a la sección “Cargar”.</p>		
Sección “Salvar”		
Acción del Actor	Respuesta del Negocio	
1. Selecciona la opción “Save as DataExport”.	<p>1.1 Verifica si existían salvas anteriores. En caso positivo salta al paso 1.3.</p> <p>1.2 Muestra cuadro de dialogo para especificar destino.</p> <p>1.3 Salva los cambios realizados.</p>	

Capítulo II: Propuesta y descripción de la solución.

	1.4 Exporta un fichero txt y lo almacena en la dirección destino seleccionada.
Sección “Cargar”	
Acción del Actor	Respuesta del Negocio
1. Selecciona la opción “Load DataExport”.	1.1 Se comprueba la estructura del fichero, si es un nombre válido y coincide con la escena cargada. En caso negativo se muestra un mensaje de error 1.2 Se carga un fichero DataExport que fue previamente guardado.
Poscondiciones	

Tabla 2.7 CU Salvar ficheros

2.7.7. CU Administrar fichero Malla de navegación

CU	Administrar fichero Malla de navegación.
Actores	Diseñador
Resumen	Se carga un fichero con la extensión txt.
Precondiciones	Estén creados datos MA o PIT para el proceso Salvar
Referencias	RF5, RF6,RF7,RF8, RF9,RF10,RF11,RF12, RF13
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio

Capítulo II: Propuesta y descripción de la solución.

<p>a) Si el actor desea salvar en una malla de navegación ir a la sección “malla navegación”.</p> <p>b) Si el actor desea cargar una malla de navegación ir a la sección “malla navegación”.</p>	
Sección “Salvar”	
Acción del Actor	Respuesta del Negocio
1. Selecciona la opción “Save As NavMesh”.	<p>1.1 Verifica si existían salvadas anteriores. En caso positivo salta al paso 1.3.</p> <p>1.2 Muestra cuadro de dialogo para especificar destino.</p> <p>1.3 Salva los cambios realizados.</p> <p>1.4 Exporta un fichero txt y lo almacena en la dirección destino seleccionada.</p>
Sección “Cargar”	
Acción del Actor	Respuesta del Negocio
1. Selecciona la opción “Load”.	<p>1.1 Se comprueba la estructura del fichero y si es un nombre válido. En caso negativo se muestra un mensaje de error</p> <p>1.2 Se carga una malla de navegación que fue previamente guardada.</p> <p>1.3 Se visualiza la malla y se crea el escenario de trabajo</p>
Poscondiciones	

Tabla 2.8 *CU Cargar ficheros*

Conclusiones del capítulo

En este capítulo se presentó la tecnología seleccionada para el desarrollo de la solución. Se realizó la propuesta del módulo de obtención de datos topológicos y medioambientales y de la herramienta *TEnD Editor*. Al finalizar este capítulo se obtuvo la especificación de los requerimientos del cliente, esto fue empleado para definir los CU del sistema cuya descripción serán empleadas en el diseño del sistema.

CAPÍTULO III: DISEÑO E IMPLEMENTACIÓN

Introducción

En este capítulo se abordan los temas del diseño y de la implementación del sistema, basado en todo el trabajo acumulado a lo largo de los capítulos anteriores y mostrando la realización de los distintos artefactos correspondientes a cada uno de los flujos de trabajo analizados. Además se muestran los resultados del desarrollo del módulo.

3.1. Modelo del diseño

El modelo del diseño describe la realización física de los CU centrándose tanto en los requisitos funcionales como en los requisitos no funcionales. En el diseño se modela el sistema y se confecciona su estructura (arquitectura), que sirve de soporte para la realización de todos los requisitos. La realización de los CU del diseño contiene una descripción de los flujos de eventos textuales, diagramas de clases y diagramas de interacción. Los diagramas de clases son los más utilizados en el modelado de sistemas orientado a objetos.

3.1.1. Diagrama de clases del diseño

Un diagrama de clases muestra un conjunto de clases, interfaces y colaboraciones, así como las relaciones existentes entre las mismas. Una clase del diseño es una abstracción sin costuras con una clase o construcción similar en la implementación del sistema donde se le atribuyen visibilidad a sus atributos y operaciones.

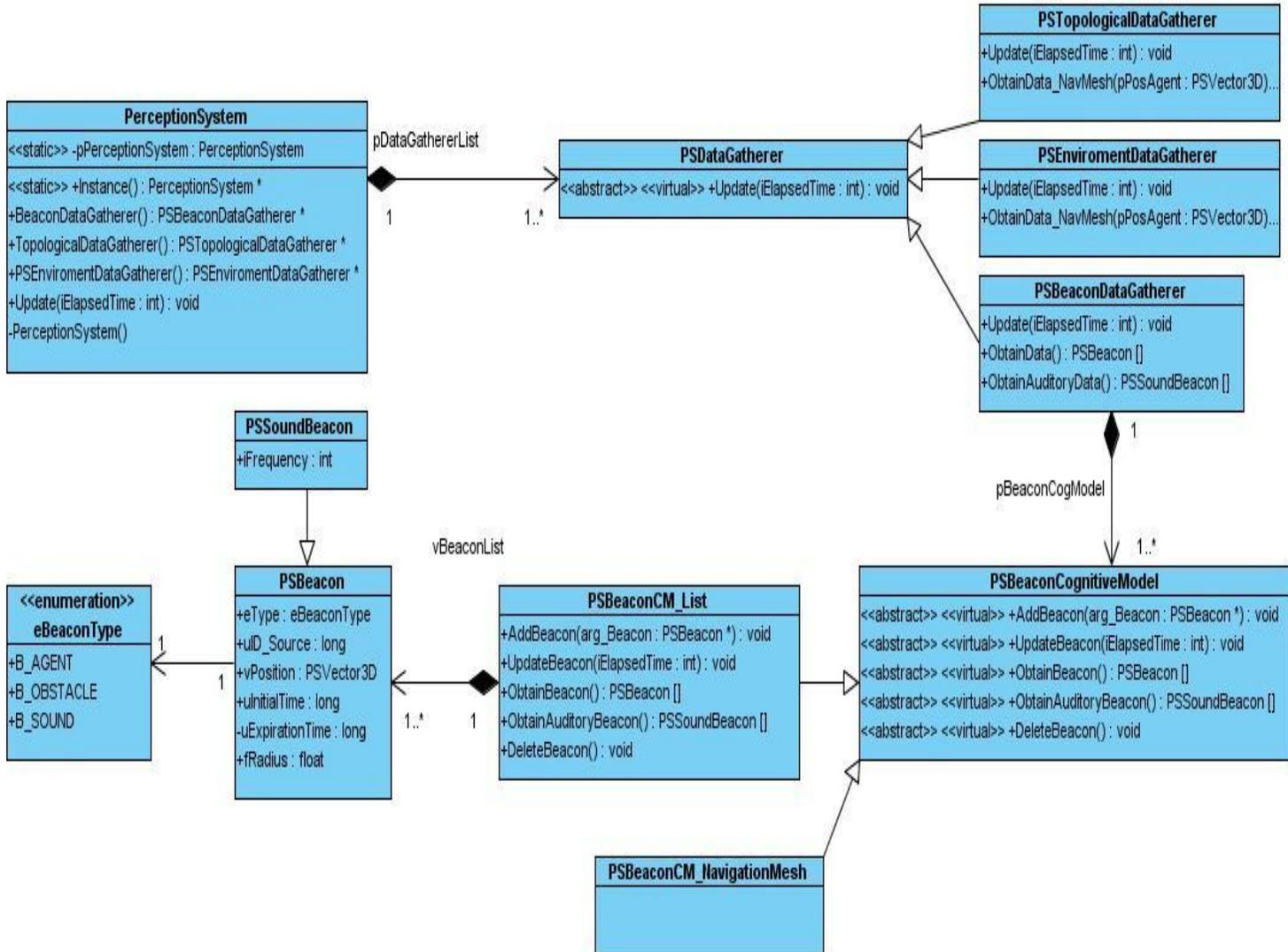


Figura 3.1 Diagrama de clases del diseño del Sistema de percepción

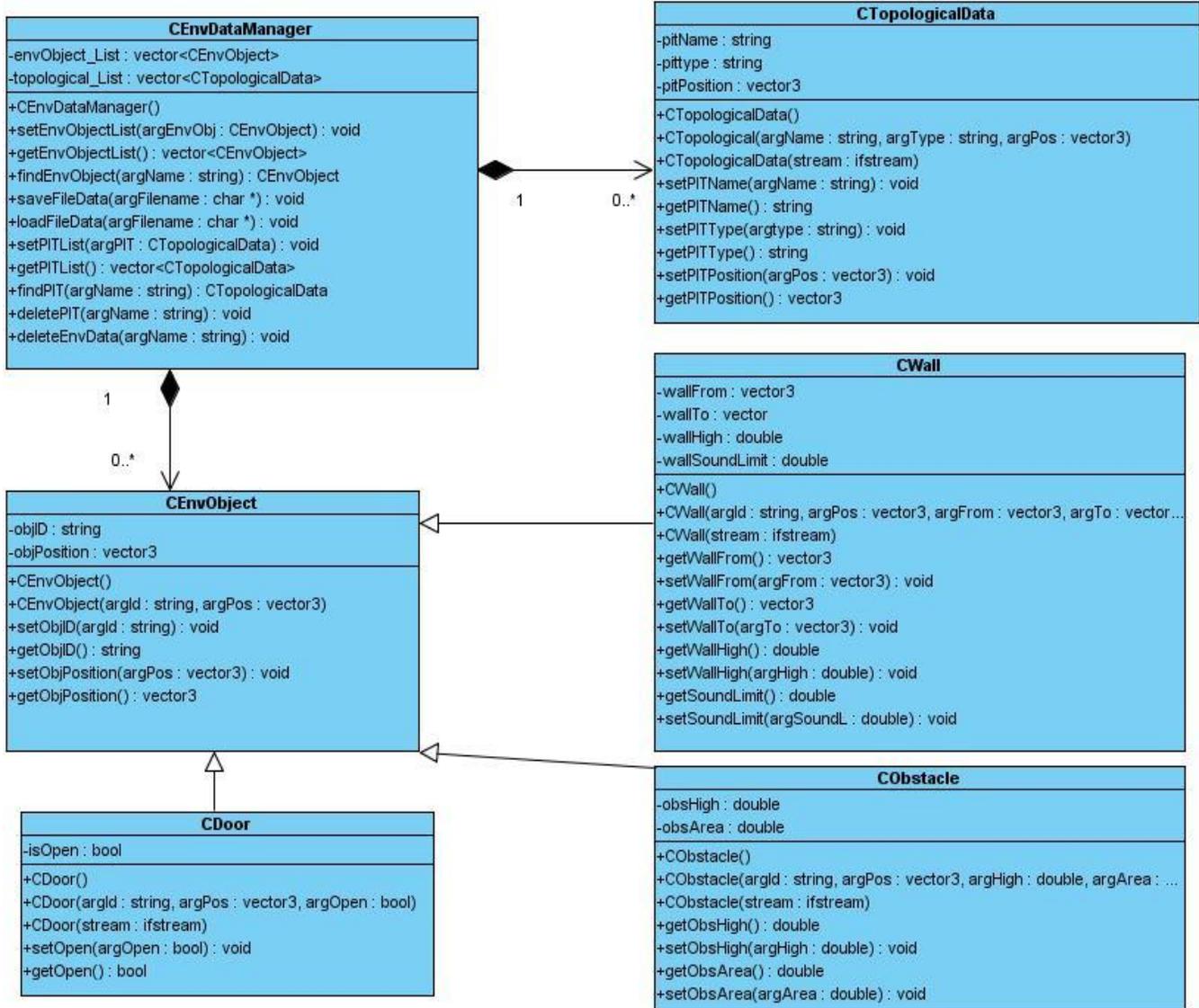


Figura 3.2 Diagrama de clases del diseño del TEnD Editor.

3.1.2. Diagramas de secuencia del diseño

A continuación se encuentran los diagramas de secuencia del diseño para cada uno de los distintos CU y clases existentes en dicho sistema; de forma tal que se facilite la comprensión de las relaciones entre los mismos.

3.1.2.1. CU Actualizar los datos medioambientales

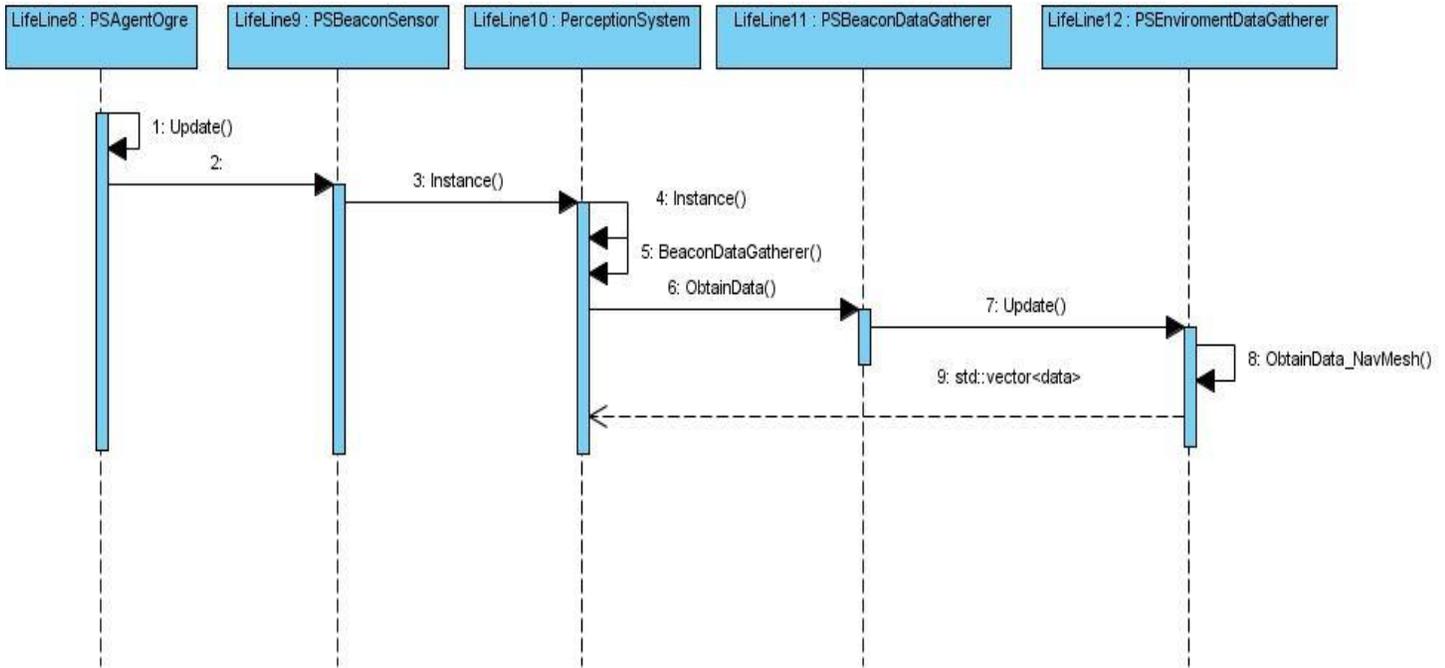


Figura 3.3 Diagrama de secuencia Actualizar los datos medioambientales

3.1.2.2. CU Actualizar los datos topológicos

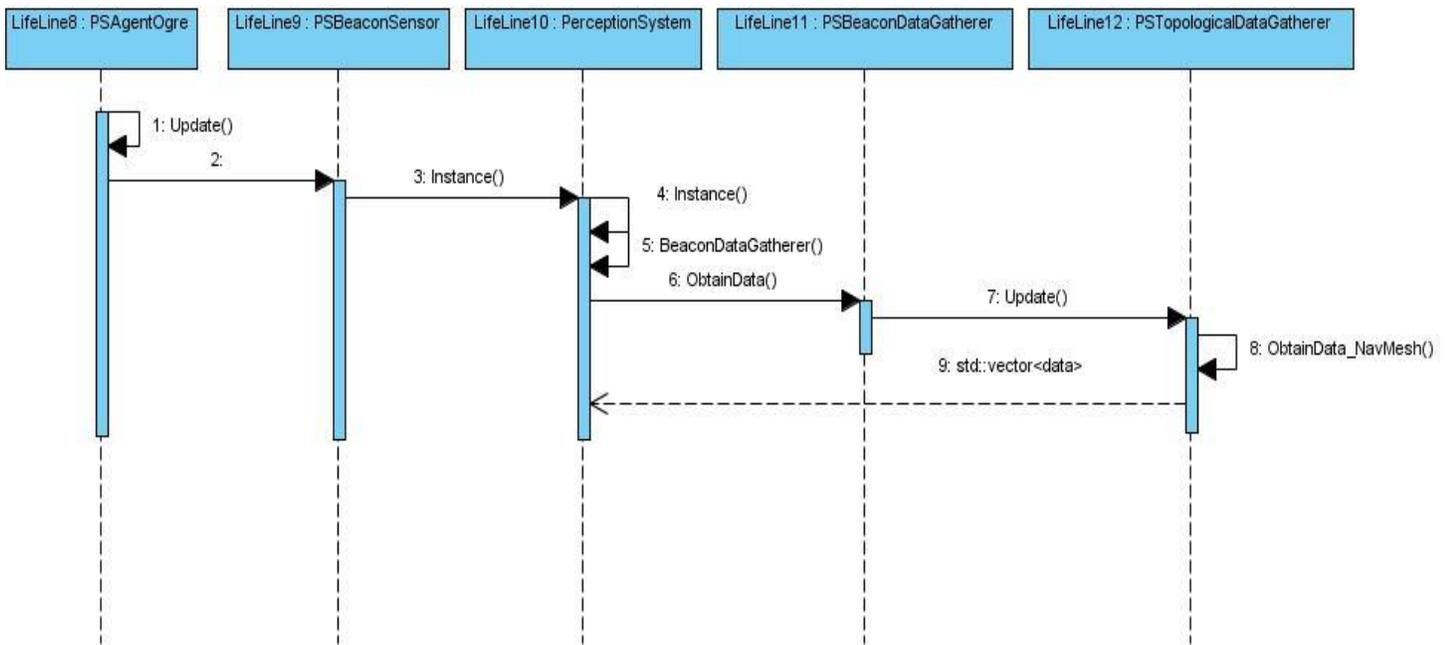


Figura 3.4 Diagrama de secuencia Actualizar los datos topológicos

3.1.2.3. CU Cargar escena

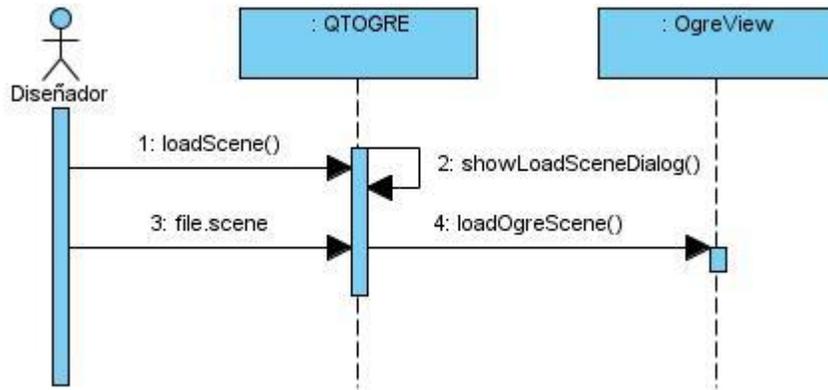


Figura 3.5 Diagrama de secuencia Cargar escena

3.1.2.4. CU Gestionar datos Medioambientales. Sección Insertar

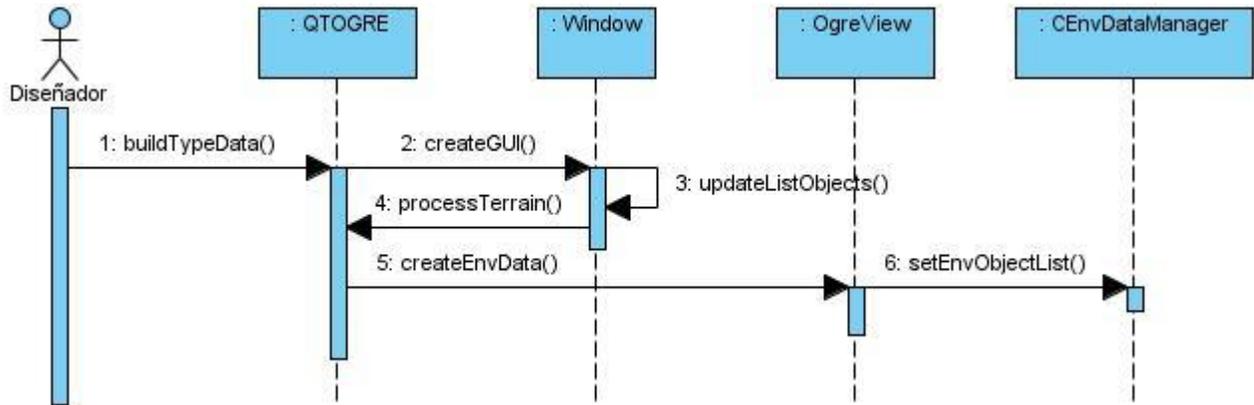


Figura 3.6 Diagrama de secuencia Gestionar datos Medioambientales. Sección Insertar

3.1.2.5. CU Gestionar datos Medioambientales. Sección Modificar

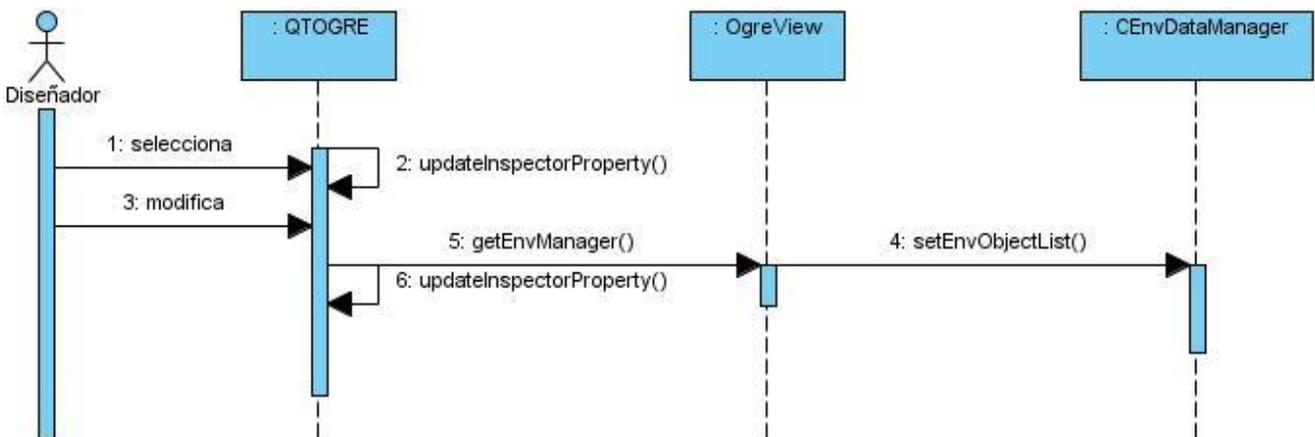


Figura 3.7 Diagrama de secuencia Gestionar datos Medioambientales. Sección Modificar

3.1.2.6. CU Gestionar datos Medioambientales. Sección Eliminar

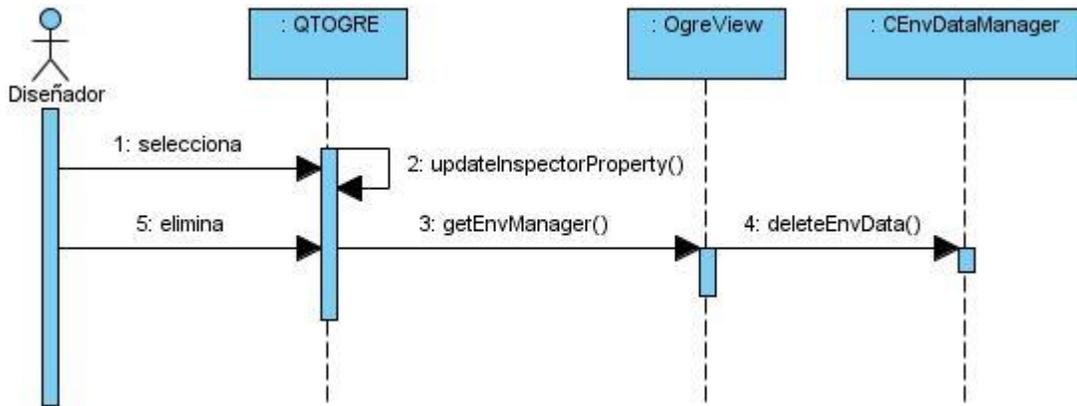


Figura 3.8 Diagrama de secuencia Gestionar datos Medioambientales. Sección Eliminar

3.1.2.7. CU Gestionar Puntos de Interés Topológicos. Sección Insertar

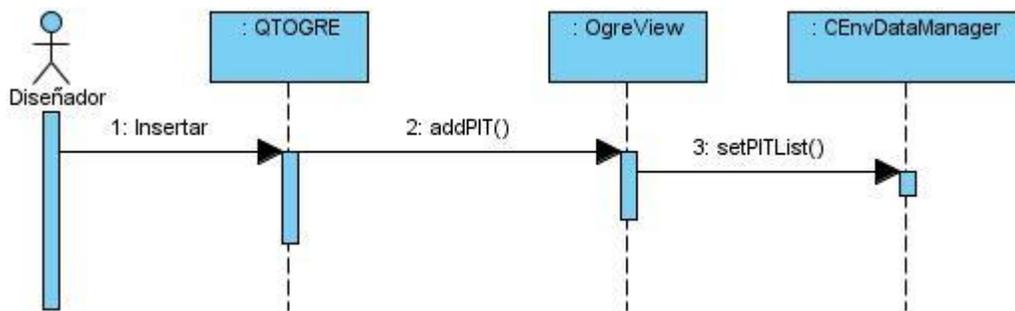


Figura 3.9 Diagrama de secuencia Gestionar Puntos de Interés Topológicos. Sección Insertar

3.1.2.7. CU Gestionar Puntos de Interés Topológicos. Sección Modificar

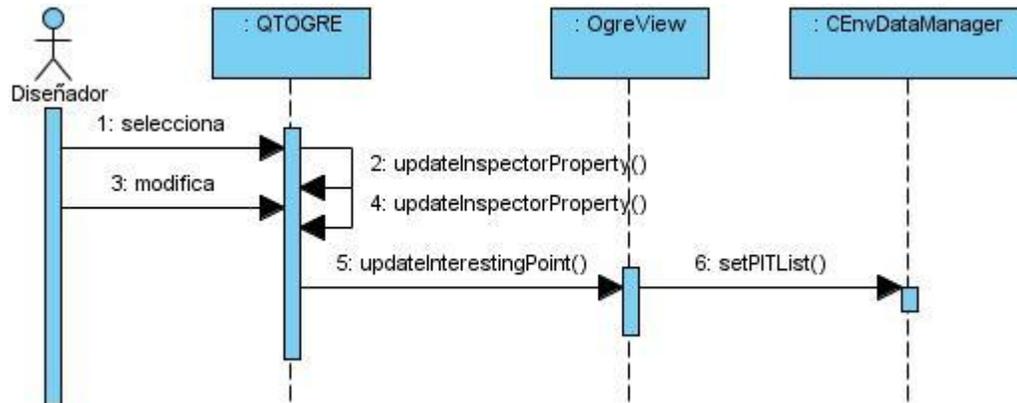


Figura 3.10 Diagrama de secuencia Gestionar Puntos de Interés Topológicos. Sección Modificar

3.1.2.7. CU Gestionar Puntos de Interés Topológicos. Sección Eliminar

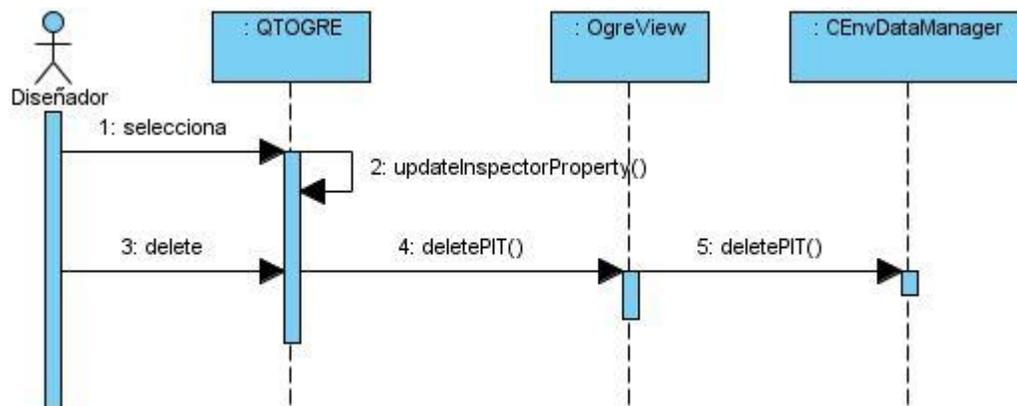


Figura 3.11 Diagrama de secuencia Gestionar Puntos de Interés Topológicos. Sección Eliminar

3.1.2.8. CU Salvar ficheros. Sección NavMesh

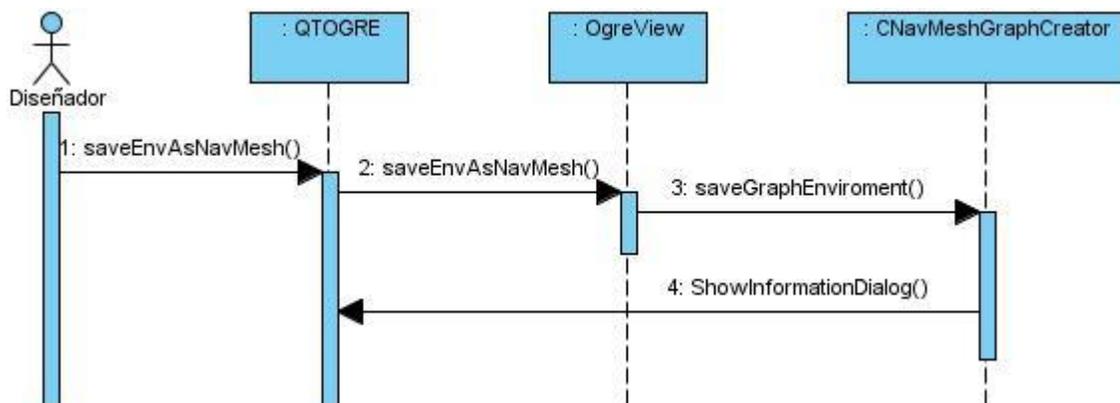


Figura 3.12 Diagrama de secuencia Salvar ficheros. Sección NavMesh

3.1.2.9. CU Salvar ficheros. Sección DataExport

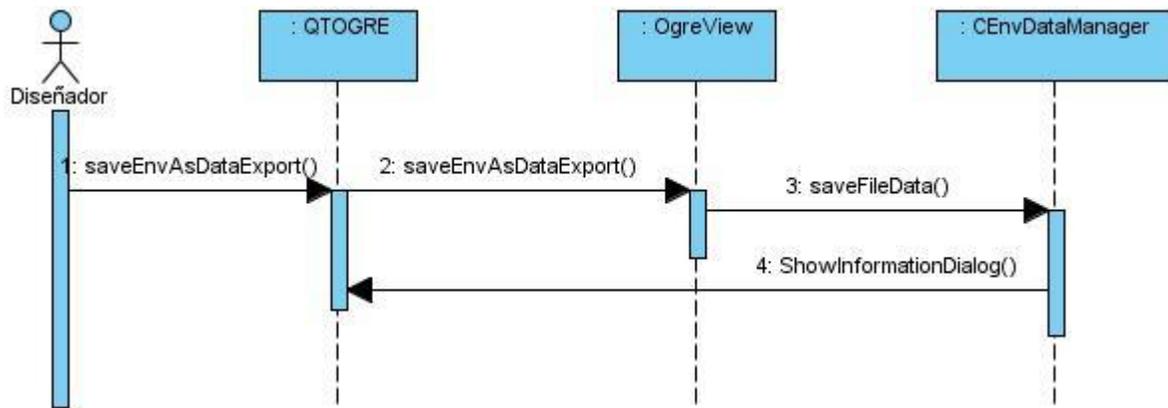


Figura 3.13 Diagrama de secuencia Salvar ficheros. Sección DataExport

3.2. Modelo de implementación

El flujo de trabajo de implementación describe cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue.

Los diagramas de componentes y de despliegue, que son artefactos generados en este flujo de trabajo, conforman lo que se conoce como un modelo de implementación, al describir los componentes a construir y su organización; así como su dependencia entre nodos físicos en los que funcionará la aplicación, respectivamente.

3.2.1. Diagrama de componentes

Las clases resultantes del análisis y del diseño se hacen físicas mediante componentes. A continuación se muestra cómo se agruparon dichas clases en los componentes según las relaciones que tienen entre sí y una breve descripción de las mismas, entre los componentes que más se utilizan para el desarrollo de esta aplicación.

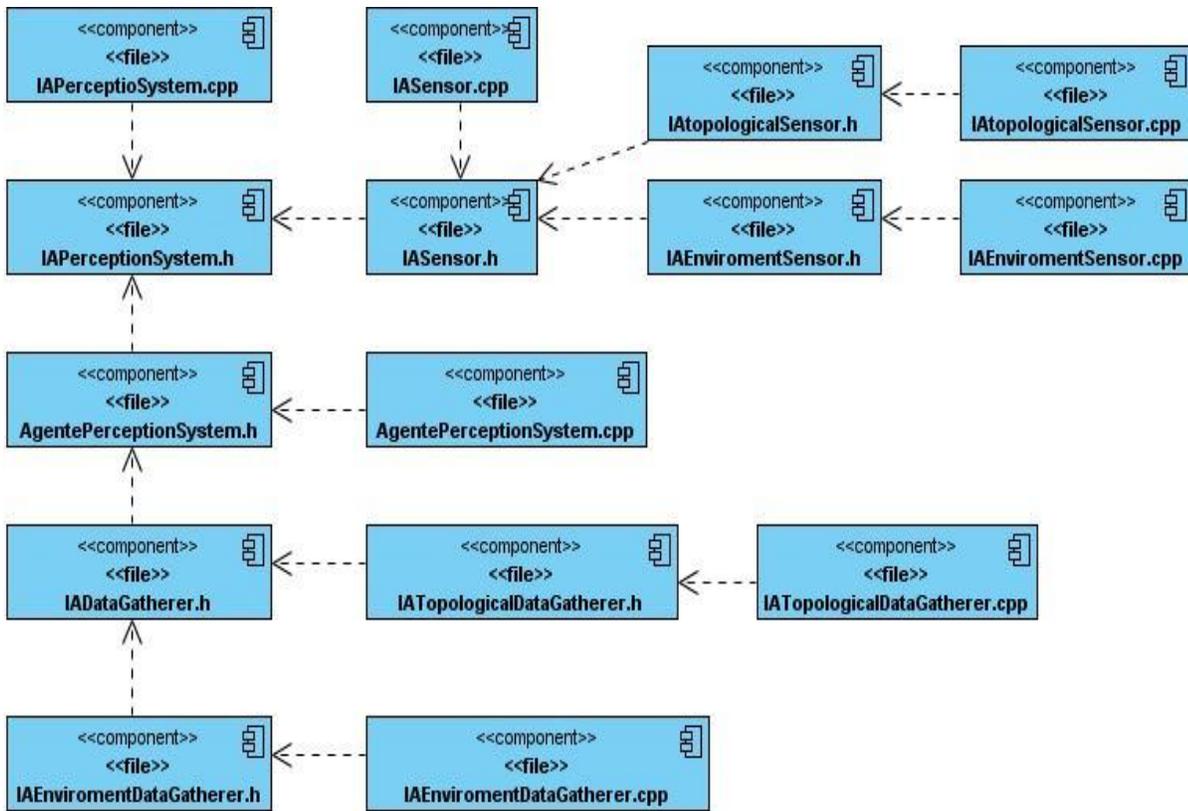


Figura 3.14 Diagrama de componentes del Sistema de percepción

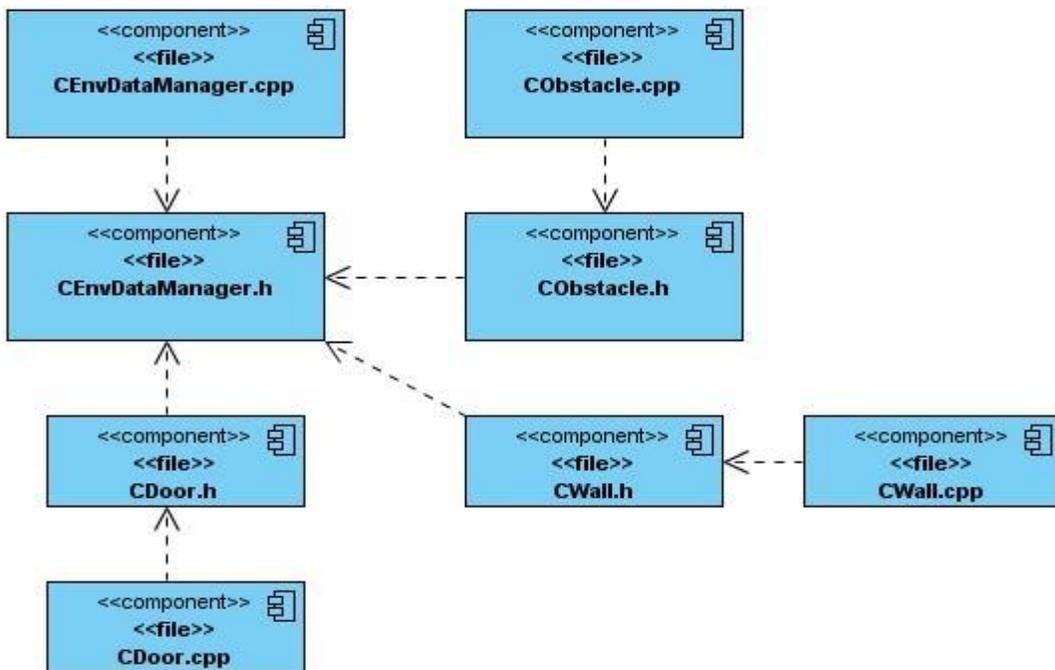


Figura 3.15 Diagrama de componentes de las clases que provee la herramienta TEnD Editor

3.2.2. Diagrama de despliegue

En el diagrama de despliegue se indica la situación física de los componentes lógicos desarrollados. Es decir, se sitúa el software en el hardware que lo contiene. Cada hardware se representa como un nodo. Un nodo se representa como un cubo y es un elemento donde se ejecutan los componentes. Atendiendo a las características del sistema, antes planteadas, se realizó el diagrama de despliegue que se muestra a continuación, el cuál satisface las necesidades y exigencias de la aplicación.

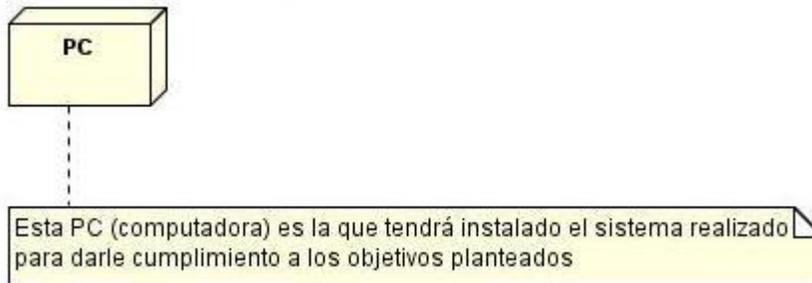


Figura 3.16 Diagrama de despliegue

3.3. Resultados

Como resultado del desarrollo de la aplicación se obtuvo un módulo de obtención de datos topológicos y medioambientales, el cual fue acoplado al SP. El módulo le permite a los AI actualizar sus datos topológicos y medioambientales en tiempo de ejecución, posibilitando conocer solo los datos de su entorno local. Además, le permite tomar en cuenta la topología y el medioambiente local para que puedan determinar cómo moverse a un lugar designado mientras se evitan las colisiones o dónde esconderse de un enemigo usando los PIT más cercanos. Para demostrar los resultados mencionados el módulo fue probado en el Entrenador Aduanero con gran aceptación. Ver Anexos 1 y 2.

Como valor agregado para el módulo se desarrolló la herramienta *TEnD Editor* (Anexo 3) que permite cargar un EV, posibilitando una vía sencilla de seleccionar los objetos que serán tomados como datos medioambientales (Anexo 4) e insertar los PIT que serán los datos topológicos (Anexo 5). La herramienta permite mediante un editor de propiedades modificar las características editables de los objetos seleccionados. El resultado final será salvado dando varias opciones de guardado al usuario, el cual podrá salvar en una malla de navegación o en un fichero propio del sistema para luego ser cargado en el SP. También permite cargarlos para una posible modificación.

Conclusiones del Capítulo

En el capítulo que concluye se realizaron los flujos de trabajo correspondientes al diseño y a la implementación, generando cada uno de los artefactos proporcionados por la metodología seleccionada y por las características principales de la propuesta realizada. Se realiza un análisis de los resultados obtenidos por el módulo y la herramienta desarrollada.

CONCLUSIONES

Al concluir la investigación realizada y después de haber valorado cada uno de los resultados obtenidos en los capítulos anteriores, se puede arribar a las siguientes conclusiones:

- Se obtuvo un módulo de obtención de datos topológicos y medioambientales, el cual fue incorporado al SP que se está desarrollando, posibilitando una mayor interacción del agente con su entorno y mejorando su toma de decisiones.
- Se desarrolló una herramienta que brinda la posibilidad de determinar los objetos que serán considerados como medioambientales, así como la localización de los puntos de interés topológico. Además propone un sistema de clases que permite cargar la información obtenida del EV.
- El SP obtenido, al hacer uso de la nueva arquitectura definida como solución al problema planteado, se aplicó al proyecto productivo Aduana. Y estará disponible para ser usado por cualquier entidad.

RECOMENDACIONES

- Ampliar los formatos de la herramienta para cargar una escena.
- Desarrollar de forma dinámica la obtención de los datos medioambientales para que se pueda interactuar con EV donde se agreguen y destruyan los objetos del entorno.
- Implementar nuevos modelos cognitivos para acoplarlos a la solución.

BIBLIOGRAFÍA

1. **Herrero Martín, María del Pilar.** *Modelo de percepción para agentes virtuales inteligentes.* Tesis doctoral. Madrid: Universidad politécnica de Madrid. Departamento de informática : s.n., 2003.
2. **Hernández López, Luis Daniel y Peña Escobar, Jorge Luis.** *Sistema de percepción para agentes virtuales.* Habana,Cuba : UCI, 2008.
3. **B., Blumberg.** *Go whit the flow: Synthetic Vision for autonomuos animated creatures.* California : s.n., 1997.
4. **D., Terzopoulos, X., Tu y R., Grzeszczuk.** *Artificial fishes: Autonomous locomotion, preception, behAlior and learning in a simulated physical world.* Cambridge : s.n., 1995.
5. **Terzopoulos, D. y Rabie, T. F.** *Animal vision: Active vision in artificial animals.* 1997.
6. **Placeres, Frank Puig.** Generic Perception System. [aut. libro] Steve Rabin. *Ai Game Programming Wisdom 4.* Estados Unidos : Charles River Media, 2006.
7. **Del Valle Guevara, Yenifer.** *Sistema de percepción genérico para agentes autónomos en videojuegos.* Universidad de las Ciencias Informáticas : s.n., 2010.
8. **Placeres, Frank Puig.** *DEA Topics. Desarrollo de Elementos Inteligentes.* La Habana : s.n., 2008.
9. **Pontevia, Pierre.** *Open the eyes of your Non Player Characters.*
10. **Sterren, William van der.** *Terrain Reasoning for 3D Action Game.* *GameDevelopers Conference.* San Jose Convention Center : s.n., 2001.
11. Autodesk- 3D Design. [En línea] [Citado el: 25 de 03 de 2011.]
<http://usa.autodesk.com/adsk/servlet/pc/index?id=11390544&siteID=123112>.
12. **Iparraguirre, Andrés y Castro-Palomino, Miguel.** *Herramienta de edición y análisis de topográficos de estructuras de nAlegació.* Habana : UCI, 2007.
13. **Lidén, L.** *Strategic and Tactical Reasoning with Waypoints.* 2002.
14. **Petty, Maikel D., Franceschini, Robert W. y Mukherjee, Amar.** *A terrain reasoning algorithm for defending a fire zone.* 1999.

15. **Petty, Mikel D.** *Computational geometry Techniques for Terrain Reasoning and Data Distribution Problems in Distributed Battlefield Simulation*. University of Central Florida : s.n., 1997.
16. **Smed, Jouni y Hakomen, Harri.** *Algorithms and Networking for Computer Games*.
17. **Massip, Reinier Melian y Martinez, Lianet Campos.** *Herramienta para la Creacion de Mallas de NAlegacion en Entornos Virtuales*. Habana : s.n., 2010.
18. **Fernández, Marcos.** Ampliación de Informática Gráfica. [En línea] 2006.
http://informatica.uv.es/iiguia/AIG/web_teoría/Tema1_cn.pdf.
19. [En línea] [Citado el: 12 de 04 de 2011.]
<http://oreilly.com/pub/a/windows/2005/08/22/whatisVisualStudio.html>.O'Reilly..
20. [En línea] [Citado el: 12 de 04 de 2011.] <http://www.ogre3d.org/about>.
21. [En línea] [Citado el: 12 de 04 de 2011.] <http://qt.nokia.com/products/qt>.

GLOSARIO DE TÉRMINOS

Entorno virtual: Es el encargado de interactuar con el agente, comunicar independencia, colaborar y elegir, dándose la oportunidad de convertirse en un nuevo modelo comunicativo, un poco más interactivo, participativo, divertido y colaborativo.

Inteligencia artificial: Se denomina a la ciencia que intenta la creación de programas para máquinas que imiten el comportamiento y la comprensión humana.

Interfaz de usuario: Es la parte del programa informático que permite el flujo de información entre varias aplicaciones o entre el propio programa y el usuario.

Modelo cognitivo: Representación del mundo virtual, que ofrece información a los agentes del entorno para que puedan interactuar con el mismo de manera eficiente, haciendo uso de diferentes estructuras de datos: Lista, ABT, BSP y las rejillas, como modelos más especializados.

Percepción: Proceso nervioso superior que permite al organismo, a través de los sentidos, recibir, elaborar e interpretar la información proveniente de su entorno.

Realidad Virtual: Es una representación de las cosas a través de medios electrónicos, que nos da la sensación de estar en una situación real en la que podemos interactuar con lo que nos rodea.

Topología: Rama de las matemáticas que estudia las propiedades de las figuras con independencia de su tamaño o forma.

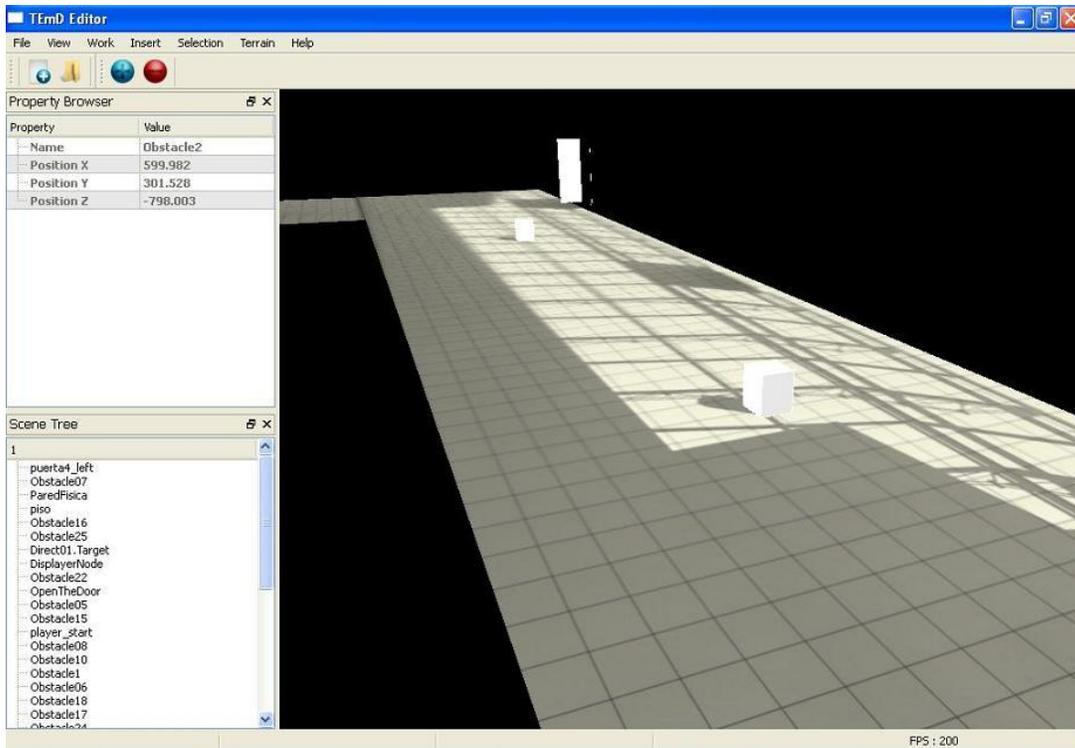
ANEXOS



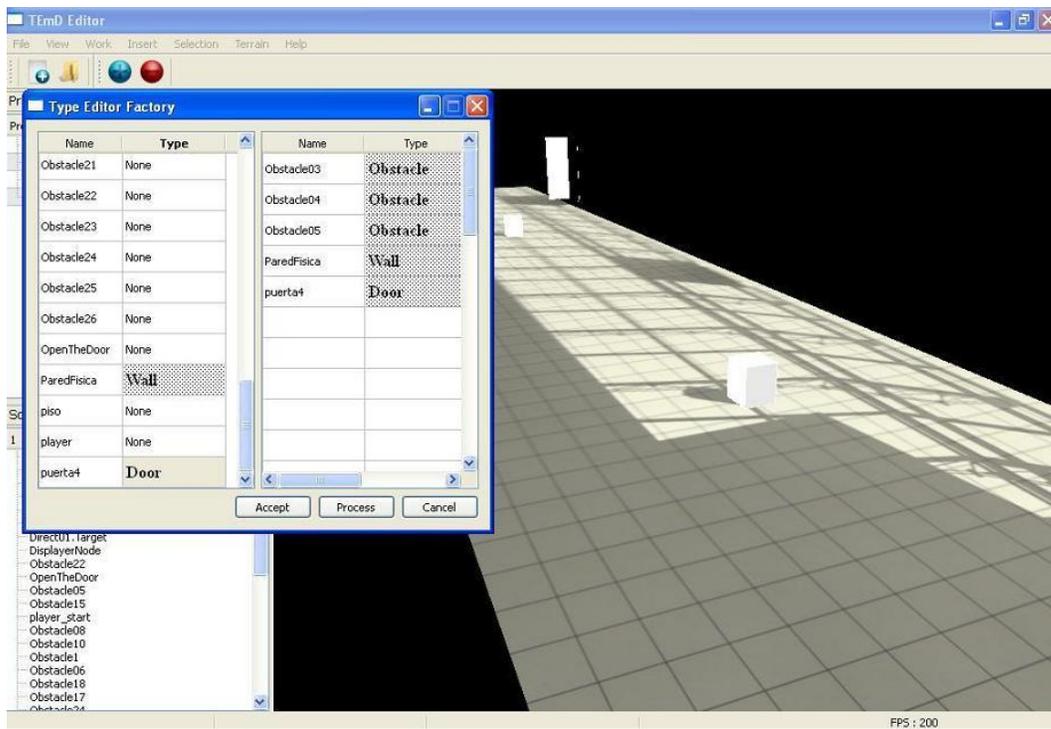
Anexo 1. Pasajeros percibiendo una puerta



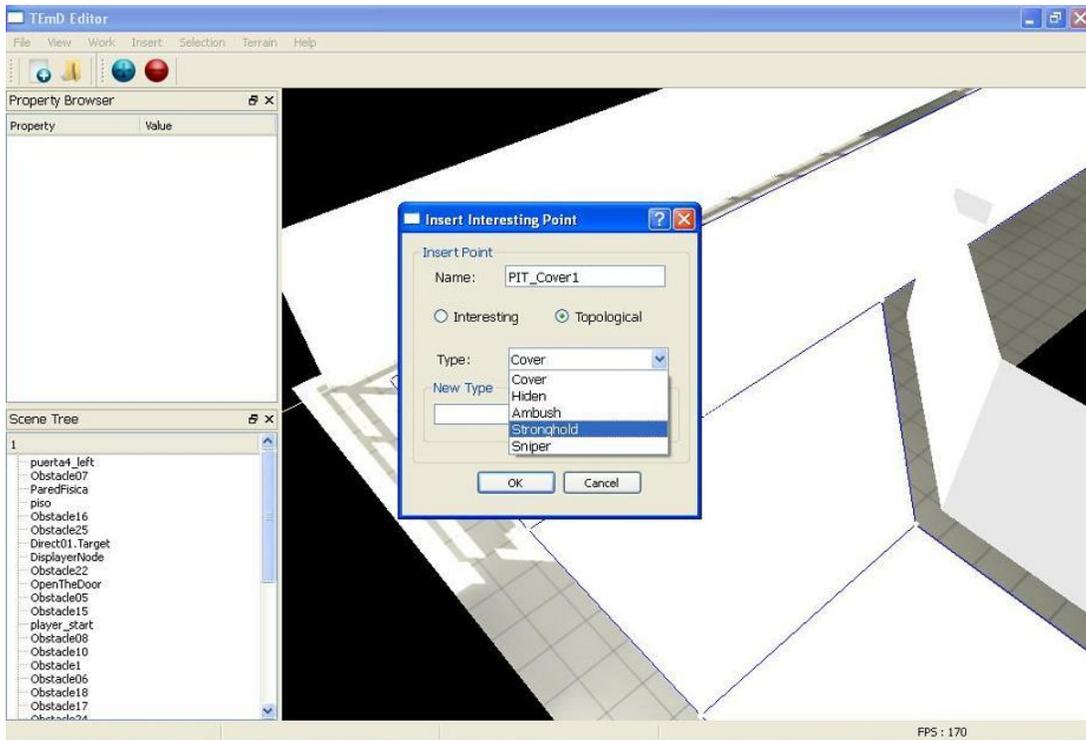
Anexo 2. Pasajeros percibiendo obstáculos



Anexo 3. Interfaz de la herramienta TmD Editor



Anexo 4. Selección de los objetos medioambientales



Anexo 5. Inserción de PIT