

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 5



**“Módulo de
Seguimiento de Terrenos
Para Sistemas de
Realidad Virtual”**

Trabajo para optar por el Título de Ingeniería en
Ciencias Informáticas

Autores: Daily Ibarreche Delgado

Manuel Macías Martínez

Tutor: Ing. Yanoski Rogelio Camacho Román

Asesor: MSc. Pedro Carlos Pérez Martinto

Ciudad de la Habana

Junio 2007

“caminante no hay camino, se hace camino al andar...”

Antonio Machado.

Declaración de Autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos al Proyecto “Herramientas de Desarrollo para Sistemas de Realidad Virtual”, de la Facultad 5 de la Universidad de las Ciencias Informáticas, a hacer uso del mismo en su beneficio.

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

Autores:

Daily Ibarreche Delgado

Manuel Macías Martínez

Tutor:

Ing. Yanoski Camacho Román

Datos de Contacto

Nombre y Apellidos: Yanoski Rogelio Camacho Román

Edad: 26 años

Ciudadanía: cubano

Institución: Universidad de las Ciencias Informáticas (UCI)

Título: Ingeniero en Informática

Categoría Docente: Profesor Instructor

E-mail: rcamacho@uci.cu

Graduado de la CUJAE, con tres años de experiencia en el tema de la Gráfica Computacional y líder de un proyecto de Realidad Virtual en la Universidad de las Ciencias Informáticas.

Dedicatoria

A mi abuelita linda Aida, que estaría orgullosa de mis 22 años.

A mis padres Daimy y Roberto, por todo su amor incondicional y su guía continua, siempre alentándome e inculcándome la superación profesional.

A mi novio Juan Carlos, por todo el amor, la fuerza y el apoyo que vierte en mí.

A mi hermanita linda Deiny, con el añoro que un día pueda leer la dedicatoria de ella.

A mi familia, por toda su preocupación y apoyo.

A mi tía Marta que ha sido como mi abuela.

A mi amiga Yailín (Lope), que aun estando lejos significó impulso y cariño.

A Sulaimi y a Gretel.

A mi juventud, por ser este el fruto de tantos años de esfuerzo.

A todas las maestras (os) y profesores (soros) que dejaron su huella en mí desde mi primera clase.

A mis amistades antiguas, por formar parte de los años que hicieron posible disfrutar este sueño hecho realidad, a mis amistades actuales por brindarme su apoyo y ocupar un lugar en mi corazón, que aunque pequeño tiene un lugar inmenso para todos.

Daily

A mis padres y hermanos, pues a ellos me debo.

Manuel (Noly)

Agradecimientos

de Daily

En especial a mi tutor Yanoski, por estar presente en todo momento que lo necesité.

A mi novio Juan Carlos por toda su ayuda, amor y comprensión inagotable.

A Walter, por todo su aporte, a Millet por prestarme su tiempo. A Pedro Carlos Pérez Martinto por ser tan atento, a Lenna por su tiempo libre, a Lien, Yaima, Sachel, Osley por estar siempre dispuestos.

A mis padres por su preocupación, comprensión y amor eterno.

A mi papito por su experiencia, su conocimiento, su dedicación a la familia. Gracias pipo.

A mi mamita por toda su dulzura y cariño, por todo su esfuerzo y voluntad ante el trabajo, ante sus niña ante la vida, por sus consejos únicos y sabios.

A Nely y Dayami por toda su ayuda y sus consejos. A Lisbe por escucharme siempre.

A todos los que me dieron su apoyo y a los que se ofrecieron en algún momento brindando su ayuda.

A la UCI por darnos la oportunidad de formarnos en sus aulas y hacernos los ingenieros de la tropa de Futuro.

de Manuel (Noly)

A Yanosky R. Camacho, Profesor a usted le debemos la realización de este trabajo, gracias por darnos el tiempo y su propio esfuerzo junto con increíble paciencia para con nosotros.

A mis padres Manuel Macías Fernández y Olga Martínez Ortega, por el respeto, responsabilidad, dedicación y tenacidad que me inculcaron, por su apoyo incondicional, por su desvelo constante por mi superación, por estar ahí cuando mas lo necesite, por siempre darme el consejo apropiado en el momento oportuno, por ser mis mayores inspiradores para con la vida.

A mi padre, por su inteligencia, por su apoyo, por sus valores, por su ejemplo, y sobre todo por confiar en mí.

A mi madre, por sembrar en mí la semilla de la sensibilidad y la comprensión; por su ternura, por su atención, por su cariño y amor. Gracias mami.

A mis hermanos Laura Macías Martínez y Pavel González Martínez, por todos los juegos, conversaciones y momentos vividos, aún en la distancia. Por su cariño incondicional, por la confianza que han depositado en mi.

A Eileen Núñez Huerta, por su apoyo, amor, respeto y colaboración, durante todo este tiempo. Por compartir las alegrías, las penas y por siempre tener palabras de aliento.

A mis abuelas, por sus ganas de vivir y ver en mí satisfacción al realizarme en mis estudios.

A Daily Ibarreche Delgado, por dejarme ser tu compañero de Trabajo, de Tesis y mi amiga, por darme apoyo y consuelo en el momento apropiado.

A Carlos Alberto, por ser mi amigo y hermano; por siempre estar presente, por su cariño; además de "darme siempre una manito" cuando la he necesitado.

A Raul Perrez-Alejo, por todas las penas y alegrías vividas juntas, gracias por tu amistad.

A mis amigos del apartamento, por permitirme conocerlos y ser parte de sus vidas, por ayudarme y estar conmigo durante este año y aun después.

A Yaself Machado Tugores, por su desinteresada forma de ayudar.

A Mayra Duran, por su particular amistad, por sus enseñanzas.

A Annia Mestre, por su peculiar apoyo.

A Francisca A. Mazorra, por el apoyo brindado.

A Pedro Carlos Pérez Martinto, por su disposición de ayudar en todo momento.

A todos los profesores que me han transmitido sus conocimientos durante mi vida estudiantil.

A la Revolución, que me ha permitido ir convirtiendo sueños en realidades.

A los viejos amigos, que a pesar de todo, están ahí.

A mi familia, quienes hicieron posible que haya llegado hasta aquí y que con su motivación y paciencia hicieron posible que pueda estar aquí hoy.

A mis amigos, especialmente aquellos que involuntariamente no haya mencionado.

Gracias, eternamente gracias, a todos.

Resumen

Los Sistemas de Realidad Virtual han tenido una elevada connotación económica y social en la vida del hombre durante las últimas dos décadas. Dado el realismo que intenta simular representando escenas virtuales, se hace evidente la utilización de los mismos para solucionar problemas en importantes esferas como la medicina, la educación, la defensa y el entretenimiento.

En los entornos virtuales existe un tema polémico y necesario para lograr una modelación de efectos reales eficientes, específicamente relacionado con el desplazamiento de los objetos por los terrenos del mundo virtual, conocido como seguimiento de terrenos.

Este trabajo abarca el diseño y la implementación de un módulo de seguimiento de terrenos para sistemas de realidad virtual. Para esto se llevó a cabo un estudio investigativo donde se profundizó en distintos métodos y algoritmos que dan solución a este problema, teniéndose en cuenta la cantidad de puntos de apoyo de cada objeto sobre el terreno. Se desarrolló el seguimiento de terrenos para 1, 3 y 4 puntos de apoyo. Para la implementación de 4 puntos, fue necesario el manejo del método matemático de aproximación llamado Mínimos cuadrados, ubicando cada punto de apoyo en un mismo plano. El módulo resultante permite hacer dicho seguimiento aportando mayor realismo a los juegos y simuladores que lo utilicen, según lo requiera el sistema usuario.

Contenido

INTRODUCCIÓN	12
CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA	16
INTRODUCCIÓN	16
1.1 SISTEMAS DE REALIDAD VIRTUAL	17
1.2 MODELADO GEOMÉTRICO.....	20
1.2.1 Modelos digitales del terreno (MDT)	23
1.2.2 Red Irregular de Triángulos.....	26
1.2.3 Modelos de barrido	29
1.2.4 Modelado CSG (Geometría Sólido Constructiva)	29
1.3 REPRESENTACIÓN DE LA INFORMACIÓN. PARTICIÓN ESPACIAL	30
1.3.1 Elementos 2D, árboles cuaternarios o quadtree	33
1.3.2 Particiones Jerárquicas 3D, árboles octarios u octrees.	35
1.4 ALGORITMOS PARA EL SEGUIMIENTO DE TERRENOS.....	40
1.4.1 CLOD	40
1.4.2 QOTA	41
1.5 MÉTODO MATEMÁTICO PARA EL AJUSTE DE SUPERFICIES.....	43
1.5.1 Método de mínimos cuadrados	43
1.6 LENGUAJE DE PROGRAMACIÓN	45
1.7 CONCLUSIONES	46
CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.....	47
INTRODUCCIÓN	47
2.1 SOLUCIONES TÉCNICAS.....	48
2.2 REGLAS DEL NEGOCIO.....	50
2.3 MODELO DEL DOMINIO	51
2.3.1 Glosario de términos del modelo del dominio.	52
2.4 CAPTURA DE REQUISITOS	54
2.4.1 Requisitos funcionales	54
2.4.2 Requisitos no funcionales	59
2.5 MODELO DE CASOS DE USOS DEL SISTEMA.....	61
2.5.1 Actores del sistema.....	61
2.5.2 Descripciones de casos de uso.....	62
2.5.3 Modelo de casos de uso	64
2.5.4 Especificación de los casos de uso en formato expandido.....	66
CONCLUSIONES	79
CAPÍTULO 3 DISEÑO DEL SISTEMA.....	80
INTRODUCCIÓN	80
3.1 DIAGRAMAS DE CLASES DEL DISEÑO.....	81
3.1.1 Diagrama de clase del paquete Clases de la Herramienta	82
3.1.2 Diagrama de clases de paquete Clases Terreno.	83
3.1.3 Descripción de las clases del diseño	84
3.2 DIAGRAMAS DE SECUENCIA	86
CONCLUSIONES	93
CAPÍTULO 4 IMPLEMENTACIÓN DEL SISTEMA.....	94

INTRODUCCIÓN	94
4.1 ESTÁNDARES DE CODIFICACIÓN	95
4.2 DIAGRAMA DE DESPLIEGUE	102
4.3 DIAGRAMA DE COMPONENTES	102
CONCLUSIONES	104
CONCLUSIONES	105
RECOMENDACIONES.....	106
REFERENCIAS BIBLIOGRÁFICAS	107
BIBLIOGRAFÍA CONSULTADA	110
APÉNDICES	111
GLOSARIO DE ABREVIATURAS	111
GLOSARIO DE TÉRMINOS	112

Introducción

Las constantes transformaciones sociales y materiales que experimenta el mundo en la era postmoderna, han venido acelerando la expansión de las nuevas tecnologías de la información, destacándose la Informática Gráfica. Esta tecnología, con las avanzadas técnicas de Realidad Virtual desarrolladas, consigue un efecto de interactividad y realismo, logrando cambios en la forma de percibir el tiempo, el espacio e incluso la propia identidad.

El término Realidad Virtual se asocia a ambientes tridimensionales e interactivos orientados, entre otros, a la visualización de objetos 3D generados por ordenadores, recreando situaciones y mundos virtuales.

En la actualidad, la Realidad Virtual se plasma en una multiplicidad de sistemas, un ejemplo notorio es la empresa norteamericana VPL Research (Visual Programming Language), quien ha venido desarrollando competentemente, en conjunto con la NASA, una arquitectura básica para la implementación de una variedad casi ilimitada de laboratorios virtuales. [9]

Profesionales de otros campos como la medicina, economía, exploración espacial, educación y adiestramiento, turismo, arquitectura y telecomunicaciones, aplican esta tecnología, utilizando los laboratorios virtuales para una gran variedad de funciones. Sin dejar atrás también el área de defensa y de la investigación espacial o nuclear y el ámbito científico, donde se han producido los avances más espectaculares. [5]

En la actualidad mundial, la modelación de terrenos y la proporción del fácil desplazamiento sobre él, son temas que han ido ampliando su aplicación a sistemas de simulación, aportando mayor realismo al proceso y haciendo más predecible cada respuesta del sistema, conociéndose dentro del entorno virtual como seguimiento de terrenos. [5]

El seguimiento de terrenos es más que la mera representación de un paisaje, debido a que el comportamiento físico del objeto que se desplaza, requiere una modelación matemática para cumplir con las leyes físicas que rigen el movimiento. De ahí que, una

importante necesidad de la Realidad Virtual, sea simular el seguimiento de terrenos no uniformes en tiempo real, por el aporte de aparente realismo que dota al sistema.

El desarrollo científico alcanzado por Cuba y la necesidad de avanzar aceleradamente en el uso de las nuevas tecnologías de la información y las comunicaciones, ha provocado una evolución cuantitativa y cualitativa en la rama de la Informática Gráfica. Ejemplo de esto es la concepción y el desarrollo de espacios virtuales en Internet para la ciencia, encaminados a promover el trabajo colaborativo entre comunidades científicas y académicas, tomando como base la infraestructura técnica y organizativa existente en el país y su proyección para los próximos años.

La Universidad de las Ciencias Informáticas (UCI), se ha integrado a este escalonado avance de la Informática Gráfica colocando entre sus líneas de investigación y desarrollo la Realidad Virtual. Con este fin se han estado desarrollando varios proyectos enfocados a dicho tema. El proyecto titulado “Herramientas de Desarrollo para Sistemas de Realidad Virtual”, forma parte de esta primera avanzada, el cual se encarga de la investigación y desarrollo de las herramientas necesarias para la creación de otros proyectos virtuales.

En dicho proyecto se tiene desarrollada una biblioteca básica (SceneToolKit), en la cual se hallan algunas funcionalidades para el seguimiento de terrenos, pero estas no responden completamente al realismo requerido por los nuevos simuladores y juegos que utilizan dicha herramienta. Debido a que estos entornos virtuales requieren terrenos cada vez más complejos y solo se tiene creada la posibilidad de hacer seguimiento de terrenos con 3 puntos de apoyo, los cuales carecen de realismo, constituye una importante necesidad la implementación de un módulo que permita hacer seguimiento con otras cantidades de puntos, que simule de forma lógica y real el desplazamiento por los terrenos.

De ahí que surja la siguiente interrogante, constituyendo el **problema científico** de este trabajo. ¿Cómo implementar un módulo de seguimiento de terrenos que permita hacer seguimientos más reales y complejos, para el proyecto “Herramientas de Desarrollo para Sistemas de Realidad Virtual” en la Universidad de las Ciencias Informáticas? Siendo el **objeto de estudio** el proceso de establecimiento de realismo a los desplazamientos de los objetos en el entorno virtual, mientras que el **campo de acción**

se enmarca, en el proceso de seguimiento de terrenos virtuales con otras cantidades de puntos de apoyo sobre el terreno.

El **objetivo** de este trabajo es desarrollar un módulo para el seguimiento de terrenos propiciando mayor realismo en los sistemas de Realidad Virtual y su integración a la biblioteca básica creada en el proyecto “Herramientas de Desarrollo para Sistemas de Realidad Virtual”

De esta forma se defiende la siguiente idea: el módulo de seguimiento de terrenos y su integración a la herramienta donde se desarrolla el proyecto “Herramientas de Desarrollo para Sistemas de Realidad Virtual”, posibilitará un mayor realismo en el seguimiento de terrenos de las escenas virtuales en tiempo real.

Se plantean a continuación un grupo de **tareas investigativas** que permitirán dar solución a los objetivos planteados:

- Análisis de las tendencias y algoritmos para el seguimiento de terrenos a nivel mundial.
- Estudio de las características de la herramienta básica existente en el proyecto.
- Diseño de un módulo de seguimiento de terrenos virtuales.
- Implementación del módulo de seguimiento de terrenos.
- Integración del nuevo módulo a la herramienta básica del proyecto.

Como resultados de este trabajo se pretende obtener un módulo que permita hacer el seguimiento de terrenos, con lo que se brindará mayor realismo a los juegos y simuladores. Este será incorporado a la herramienta desarrollada por el proyecto, por lo que será actualizable, flexible y de fácil uso por parte de los usuarios de la misma.

El presente documento está estructurado de la siguiente manera: resumen, introducción, cuatro capítulos de contenido, conclusiones, recomendaciones, referencias bibliográficas, bibliografía y los apéndices, glosario de abreviaturas y glosario de términos.

En el *Capítulo 1: Fundamentación Teórica*, se hace un estudio donde se investigan las características de los Sistemas de Realidad Virtual (SRV), las técnicas, algoritmos y tendencias actuales para el desarrollo de seguimiento de terrenos virtuales.

En el *Capítulo 2: Descripción de la solución propuesta*, se plantean las soluciones técnicas, así como la descripción de las mejoras propuestas para lograr un mayor realismo en el desplazamiento por los terrenos de los simuladores y juegos de realidad virtual. También se crea el modelo del dominio, se definen los requerimientos funcionales y no funcionales con que debe contar el sistema y se especifican los casos de uso del sistema y sus descripciones en formato expandido.

En el *Capítulo 3: Diseño del sistema*, se presenta la relación entre las clases de seguimiento de terreno y las clases ya creadas en la herramienta del proyecto, agrupadas en paquetes para un mejor entendimiento. Además se presentan los diagramas de secuencia divididos por secciones.

En el *Capítulo 4: Implementación del sistema*, se muestra el diagrama de componentes, los estándares de codificación por los que se regirá la implementación del módulo de seguimiento de terrenos virtuales. Finalmente se proporciona un apéndice compuesto por un glosario de abreviaturas y un glosario de términos, para apoyar la comprensión del lenguaje técnico utilizado en la investigación y el desarrollo del trabajo realizado.

Capítulo 1 Fundamentación Teórica

Introducción

Los Sistemas de Realidad Virtual enfocan sus principales esfuerzos en elevar las funcionalidades de prestación de la calidad de las imágenes proyectadas, la velocidad de respuesta, la calidad con que se logran los efectos de inmersión, el número de sentidos que se emplean en el mundo virtual, así como la calidad con que se simulan, en esencia, alcanzan una manipulación real del ambiente virtual.

Para contribuir a estas funcionalidades existen características de estos sistemas que no se pueden obviar, como es el caso del desarrollo de un seguimiento de terrenos de forma lo mas real posible, hecho que dota al sistema de credibilidad y realismo aparente.

En este capítulo que comienza se lleva a cabo un estudio de las principales características de los Sistemas de Realidad Virtual, los conceptos fundamentales y las principales técnicas y tecnologías, así como los algoritmos más usados en la actualidad. También se estudian las características de la Herramienta ya desarrolladas en el proyecto, para lograr un mejor entendimiento entre lo que existe y lo que se requiere implementar.

1.1 Sistemas de Realidad virtual

Sistemas Realidad Virtual (SRV)

La Realidad Virtual (RV) constituye una tecnología de la Informática Gráfica en plena evolución, por lo que cualquier definición actual de su concepto debe considerarse solo con carácter transitorio.

El uso original de la palabra, se remonta a cuando la realidad virtual era tan sólo una imagen mental, referida casi en su totalidad a la inmersión completa de sistemas. Esto significa un complejo sistema para proyectar espacios visuales en 3D y un circuito electrónico para enviar y recibir señales con respecto a la posición de su propio cuerpo.

Su meta consiste en crear una experiencia que haga sentir al usuario en la mitad de un mundo virtual, separado del mundo real, moviéndose dentro del mundo virtual, en vez de controlar figuras generadas por una computadora alrededor de él en el mismo mundo.

Examinando con mayor detalle sus características, se definen dos tipos de Realidad Virtual, la **inmersiva** y la **no immersiva**. [22]

Los métodos **inmersivos** de realidad virtual con frecuencia se ligan a un ambiente tridimensional creado por computadora, el cual se manipula a través de cascos, guantes u otros dispositivos que capturan la posición y rotación de diferentes partes del cuerpo humano. [22]

La realidad virtual **no immersiva** utiliza medios como el que actualmente ofrece Internet, en el cual se puede interactuar en tiempo real con diferentes personas en espacios y ambientes que en realidad no existen, sin la necesidad de dispositivos adicionales a la computadora. [22]

Especificar un concepto único y completo de realidad virtual es un tema controversial en estos momentos. Por lo que a continuación se citarán varias interpretaciones por autores diversos.

«La RV es la simulación de medios ambientes y de los mecanismos sensoriales del hombre por computadora, de tal manera que se busca proporcionar al usuario la sensación de inmersión y la capacidad de interacción con medios ambientes artificiales». (Dr. Homero Ríos Figueroa).

"Realidad virtual: un sistema de computación usado para crear un mundo artificial donde el usuario tiene la impresión de estar en ese mundo y la habilidad de navegar y manipular objetos en él". (Manetta C. y R. Blade (1995)).

"La realidad virtual te permite explorar un mundo generado por computadoras a través de tu presencia en él". (Hodder y Stoughton(s/a)).

“Un Sistema de Realidad Virtual es aquel que da al usuario la experiencia de estar inmerso en un entorno sintético”. (H.Fuchs). [7]

“La realidad virtual se caracteriza por ofrecer la ilusión de participación en un entorno sintético en vez de una observación externa del mismo. La realidad virtual es una experiencia inmersiva y multisensorial”. (M. A. Gigante). [7]

“Un entorno virtual se caracteriza por ser interactivo, con un procesamiento especial de imagen, sonido y tacto, para convencer al usuario que se encuentra inmerso en un espacio sintético”. (S.R.Ellis). [7]

Teniendo en cuenta las definiciones anteriores y estableciéndose un criterio común y abarcador, que resuma de forma concisa las principales características de este término controversial, se plantea la siguiente definición:

La realidad virtual constituye la posibilidad de incursionar en un mundo artificial de la forma más cercana posible a la realidad, caracterizado por ser interactivo, inmersivo y multisensorial.

Aplicaciones de tiempo real

El concepto de **tiempo real** viene del procesamiento digital de señales. Un sistema de tiempo real es aquel capaz de procesar una muestra de señal antes de que ingrese al sistema la siguiente muestra. [25]

Los primeros sistemas de tiempo real se desarrollaron para la telefonía digital, utilizando hardware de propósito específico. Con el paso del tiempo y la mejora constante en la velocidad de los microprocesadores de propósito general, la implementación de sistemas de tiempo real se ha convertido en un campo de la informática donde se han presentado condiciones y problemas enfocados al disco duro de una computadora. [25]

Este concepto se puede aplicar a distintas ramas de la Informática y proviene de la ingeniería de control. En general la caracterización como 'tiempo real' se refiere a la existencia de determinadas restricciones sobre el comportamiento temporal de nuestro sistema. Dependiendo del tipo de aplicación, esas limitaciones serán de una clase u otra. El concepto de tiempo real, por tanto, puede aplicarse de forma más o menos estricta según lo duras que sean las restricciones impuestas. [23]

Un ejemplo de ejecución en tiempo real es la Realidad Virtual, esto expresa que lo que sucede, se calcula exactamente en ese momento. Permitiendo una libertad y realismo, que no puede dar la animación. Significa que la computadora hace todos los cálculos necesarios y presenta una imagen nueva cada 30 segundos como mínimo, haciendo que los requerimientos computacionales sean grandes. [20]

1.2 Modelado Geométrico

Existen diferentes tipos de modelos para la representación de cada objeto puesto en escena, los cuales se clasifican en modelos alámbricos, de superficies de contorno (planas y curvas) y sólidos.

Los **modelos alámbricos** son aquellos cuerpos que se representan a partir de las aristas que los componen. No tienen superficies, son los más simples, fáciles de construir, con pocas necesidades de memoria y almacenamiento, y se visualizan rápidamente, aunque tienen representación ambigua y falta de coherencia visual. [\[8.1\]](#)

Los **modelos de superficies planas (poligonales)** “son un conjunto de líneas rectas (arcos) que no se cruzan y que unen un conjunto coplanar de puntos (vértices) definiendo un área simple (habitualmente convexa y sin agujeros)”. La representación poligonal a través de triángulos mejora mucho las prestaciones del HW. Las tarjetas gráficas visualizan más triángulos/segundo de los que pueden enviarse por el bus. [\[8.1\]](#)

Los **modelos de superficies curvas** se definen por ecuaciones cuadráticas (descritos mediante ecuaciones de segundo grado) y **splines** (representación por interpolación (la curva o superficie pasa por los puntos de control) o por aproximación (la curva se ajusta a los puntos de control)); existen otros tipos de ecuaciones para expresar el modelo a través de curvas y superficies por sólo citar algunas, **Hermite**, **Bezier**, y **B-Splines**. [\[8.1\]](#)

Los **modelos sólidos** tienen en cuenta además de las características geométricas del cuerpo, los comportamientos físico-mecánicos de los mismos, por tanto son modelos más completos que los anteriores. Los modelos más simples son caracterizados a través de las ecuaciones cuadráticas y constituyen los cuerpos parametrizados tridimensionales que se pueden observar en cualquier *software* dedicado a la gráfica. A los cuerpos se les atribuye propiedades tales como masa, textura, color y se permite la interacción entre los mismos.

Existen además, modelos no convencionales muy útiles en dependencia de cuánto realismo se le quiera dar al mundo virtual, como son: fractales, gramáticas, blobs (burbujas o gotas) y modelado basado en propiedades físicas, por citar algunos.

Una técnica importante en la optimización del modelado de los cuerpos en la escena, es el modelado por niveles de detalle (LOD). Esta técnica consiste en que, según las distancias a las que se pueden encontrar los objetos respecto a la cámara de la escena, se optimizan los cuerpos conformándolos con la menor cantidad posible de polígonos, cantidad satisfactoria a su vez para la distancia a la que se encuentran. [\[8.2\]](#)

Modelado de terrenos

Una malla de un terreno no es más que una cuadrícula o red de triángulos, pero con una altura por cada vértice, la cual representará cómo se modela una transición suave de montañas a valles, por citar un ejemplo, simulando terrenos naturales, esto por supuesto con una textura adecuada.

Para el manejo de los terrenos se utilizan los llamados (***Heightmaps***) o mapas de alturas, que no son más que arreglos donde cada elemento especifica la altura de un vértice en particular de la cuadrícula del terreno. Una variante del mapa de alturas es a través de un mapa de tonalidades de grises (***Grayscale map***), donde los valores más oscuros representan los puntos bajos del terreno y los más claros, los puntos más altos. Esto brinda la posibilidad de almacenar un mayor rango de diferencias entre alturas. [1]

1.2.1 Modelos digitales del terreno (MDT)

Los Modelos digitales del terreno son una estructura numérica de datos que representa la distribución espacial de una variable cuantitativa y continua. Por consiguiente, los MDT son modelos simbólicos, ya que las relaciones de correspondencia que se establecen con el objeto real tienen la forma de algoritmos o formalismos matemáticos. [\[21\]](#)

A partir de la definición anterior, se pueden enunciar las propiedades básicas de los MDT:

- Forman una estructura de datos, lo que significa que no son sólo una acumulación o lista de cifras, sino que guardan relaciones entre ellos. (Son el producto de una modelización de datos.) [\[21\]](#)
- Representan la distribución espacial de una variable, lo que acota claramente su ámbito de actuación en la modelización. [\[21\]](#)
- Sus variables son cuantitativas y continuas. [\[21\]](#)

El papel equivalente en los MDT lo desempeñan los Modelos Digitales de Elevaciones (MDE), que describen la altimetría de una zona mediante un conjunto de cotas.

El objetivo original de los MDE fue el tratamiento de problemas tecnológicos, científicos y militares, como el diseño de carreteras mediante el tratamiento digital de datos del terreno. En los últimos años han surgido ya multitud de aplicaciones informáticas capaces de manejar eficazmente los MDE, en distintas áreas como la industria, el comercio y otros. [\[3.1\]](#)

Definición y estructura del MDE

Un MDE es una estructura numérica de datos que representa la distribución espacial de la altitud de la superficie del terreno. Un terreno real puede describirse de forma genérica como una función bivariable continua $z=z(x, y)$ donde z representa la altitud del terreno en el punto de coordenadas (x, y) y z es una función que relaciona la variable con su localización geográfica, ver (Fig. 1). En un modelo digital de elevaciones

se aplica la función anterior sobre un dominio espacial concreto, D . En consecuencia, un MDE puede describirse genéricamente como $MDE = (D, z)$.

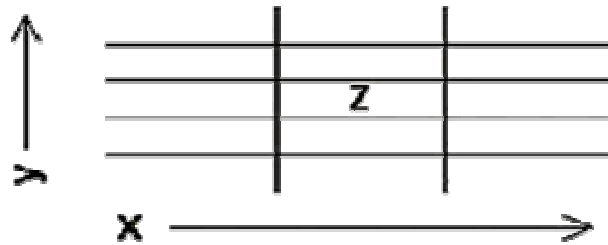


FIG. 1 SISTEMA DE REFERENCIA ESPACIAL

En la práctica, la función no es continua, sino que se resuelve a intervalos discretos, por lo que el MDE está compuesto por un conjunto finito y explícito de elementos. Los valores de x e y suelen corresponder con las abscisas y ordenadas de un sistema de coordenadas plano, habitualmente un sistema de proyección cartográfica. La generalización inherente a la discretización del modelo implica una pérdida de información que incrementa el error del MDE y en consecuencia, se propaga a los modelos derivados.

Por este motivo, se han ensayado numerosas opciones en la búsqueda de una forma de representar y almacenar la altitud, donde se equilibre la pérdida de información y algunos efectos secundarios indeseables, como son el excesivo tamaño de los archivos o la dificultad de manejo.

Estructura de datos en el MDE

De forma general, la unidad básica de información en un MDE es un punto acotado, definido como una terna compuesta por un valor de altitud z , al que acompañan los valores correspondientes de x e y . Las variantes aparecen cuando estos datos elementales se organizan en estructuras que representan las relaciones espaciales y topológicas. Mientras que los mapas impresos usan casi exclusivamente una única convención, las curvas de nivel para la representación de la superficie del terreno, en los MDE han utilizado alternativas algo más variadas.

Históricamente, las estructuras de datos en los sistemas de información geográfica y por extensión, en los Modelos Digitales del Terreno, se han dividido en dos grupos en

función de la concepción básica de la representación de los datos: vectorial y raster.

[3.1]

- **El modelo de datos vectorial** está basado en **entidades u objetos** geométricos definidos por las coordenadas de sus nodos y vértices. [3.2]
- **El modelo de datos raster** está basado en **localizaciones espaciales**, a cada una de las cuales se les asigna el valor de la variable para la unidad elemental de superficie. [3.2]

Cada modelo de datos puede expresarse mediante diferentes estructuras de datos, dentro de los dos modelos básicos, la práctica y el tiempo han reducido las potenciales variantes de estructuración a unas pocas. Las más representativas son dos estructuras vectoriales: la basada en **Isohipsas** o **Líneas de Contornos** y la **Red Irregular de Triángulos** (Triangulated Irregular Network) (TIN), y dos estructuras raster: las matrices de mallas regulares **URG**, (Uniform Regular Grids) y las matrices jerárquicas (**Quadtree**), matrices imbricadas en una estructura jerárquica. [21] y [3.1]

- **Estructuras vectoriales, basadas en entidades/objetos**
 - Contornos: polilíneas de altitud constante.
 - TIN: red de triángulos irregulares adosados.
- **Estructuras raster, basadas en localizaciones**
 - Matrices regulares: malla de celda cuadrada.
 - Quadtrees: matrices imbricadas en una estructura jerárquica. [21] y [3.1]

El papel del modelo de contornos ha quedado reducido a ser una etapa intermedia en la captura de información, por ser poco útil para el manejo por medios informáticos. Mientras que por otro lado, a pesar de que el mayor interés en la utilización de las matrices jerárquicas está en que permiten solucionar el principal problema de las matrices regulares: (su resolución espacial constante), esta estructura denominada quadtree, se ha utilizado ocasionalmente en el tratamiento de variables nominales con el fin de reducir el tamaño de almacenamiento.

Por lo que las estructuras de datos más utilizadas para el tratamiento de aplicaciones de MDT son la estructura vectorial: **TIN**, y la estructura raster: **URG**. [3.1] y [14]

1.2.2 Red Irregular de Triángulos

Esta estructura de datos se compone de un conjunto de triángulos irregulares adosados al terreno, cada uno de los cuales se adapta a una zona con características de pendiente similares, representando eficazmente un relieve determinado. Los triángulos se construyen ajustando un plano a tres puntos cercanos no colineales y se adosan sobre el terreno formando un mosaico que puede adaptarse a la superficie con diferente grado de detalle, en función de la complejidad del relieve. Se trata de una estructura en la que el terreno queda representado por el conjunto de superficies planas que se ajustan a un conjunto previo de puntos.

La creación de un TIN se compone de dos partes diferentes: la triangulación propiamente dicha y la selección de los puntos que deben usarse como vértices.

En este caso el método de triangulación más habitual es el conocido como Triangulación de Delaunay. Significa que se cubre toda la superficie a representar con triángulos lo más equiláteros posible, de tal manera que en terrenos más abruptos los triángulos son más pequeños y en terrenos de pendientes más uniformes y constantes los triángulos resultan mayores, adaptándose de esta forma a la superficie con diferente grado de detalle, en función de la complejidad del relieve, ver (fig. 2).

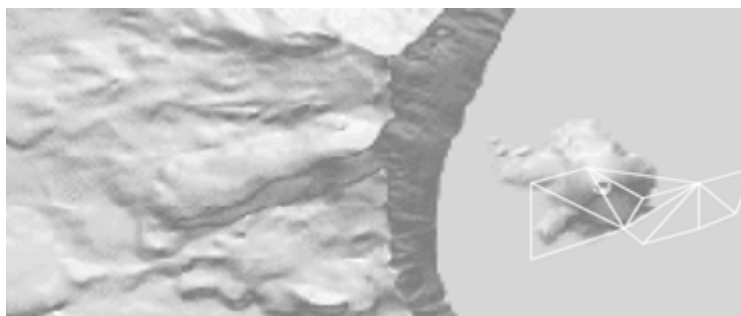


FIG. 2 RED IRREGULAR DE TRIÁNGULOS

Es una fórmula muy ajustada para definir una superficie en el espacio, pero tiene el inconveniente de que el almacenamiento que se necesita es muy grande, al tener que guardar las variables (X, Y, Z) de cada punto del triángulo. Un TIN se compone de triángulos, nodos y arcos. Esta estructura permite incorporar datos auxiliares como líneas de inflexión, red hidrológica o zonas de altitud constante.

El método de triangulación ha sido estudiado desde hace más de un siglo y en la actualidad es un tópico bien conocido, utilizado en numerosas aplicaciones como la robótica, análisis de elementos finitos, visión artificial y síntesis de génesis.

Particularmente la Triangulación de Delaunay consigue que la malla de triángulos que se genera sea lo más regular posible, cumpliendo la propiedad del círculo vacío que consiste en definir triángulos cuyas circunferencias circunscritas no contienen a ningún otro vértice de la red. [3.1]

Los vértices se forman mediante la unión de los puntos originales a partir de la determinación de los vecinos más cercanos. Más tarde, se traza una malla regular sobre la red de triángulos, estableciéndose un complejo proceso de interpolación que le asigna un valor a cada celda de la malla a partir de cálculos que incorporan el estudio de la localización del centroide de cada triángulo, su área, su pendiente, entre otros aspectos. Finalmente se obtiene una superficie suavizada que posee “influencia reducida” de los triángulos formados en la primera fase, ver (Fig. 3). [14]

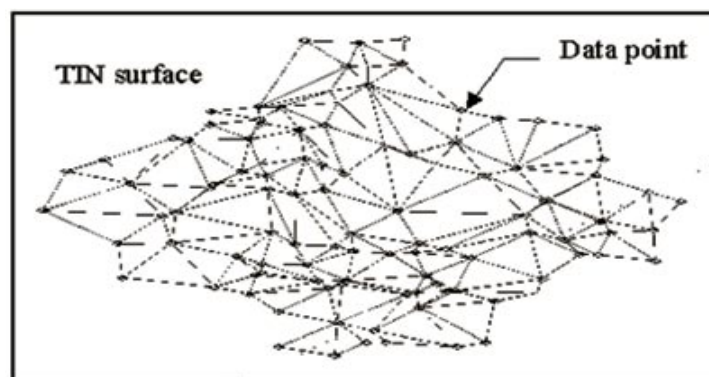


FIG. 3 EJEMPLO DE TRIANGULACIÓN.

Esta metodología presenta las siguientes ventajas: la triangulación realizada es la más regular posible y la red es independiente del número de puntos procesados. Sin embargo también presenta una serie de problemas cuando está definida para un conjunto de puntos de diferente naturaleza, también cuando tres puntos son colineales, cuando cuatro o más puntos son concíclicos y cuando existen puntos cercanos planimétricos y diferencias de cota dispares, por ejemplo en zonas de acantilados, fallas, cuevas y otras. [14]

Modelo raster: matriz regular

En el modelo raster, los datos se interpretan como el valor medio de unidades elementales de superficie no nula que teselan el terreno con una distribución regular, sin solapamiento y con recubrimiento total del área representada. Estas unidades se llaman celdas o teselas y si se admite la analogía con los términos usados en proceso de imágenes, píxeles.

La estructura matricial es el resultado de superponer una retícula sobre el terreno y extraer la altitud media de cada celda. La retícula adopta normalmente la forma de una **red regular de malla cuadrada (GRID)**. En esta estructura, la localización espacial de cada dato está determinada de forma implícita por su situación en la matriz, una vez definidos el origen y el valor del intervalo entre filas y columnas. [\[3.2\]](#)

El sistema de almacenamiento es mínimo, pues se guarda la cota de los nodos de una malla regular, espaciada una misma cantidad (paso de malla) tanto en el eje X como en el eje Y y en un sistema de coordenadas específicos. El resultado es una matriz de puntos altimétricos, donde se conocen las coordenadas del primer punto y las del último, con lo que queda descrita toda la matriz.

La matriz regular es la estructura más utilizada para construir los MDE. La razón es que se trata de una estructura de fácil manejo informático y simple de representar mediante estructuras lógicas como matrices de dos dimensiones. [\[3.2\]](#)

Existen otros tipos de representaciones de Sólidos, los cuales se clasifican en Modelos de Barrido y Modelado basado en Geometría sólida Constructiva (CSG).

1.2.3 Modelos de barrido

Varios son los tipos de barridos que conforman los **Modelos de Barrido**, ellos son: el Barrido Traslacional, Barrido Rotacional, Barrido Circular y el Barrido General, dado por la extrusión curva y la extrusión divergente. El mecanismo de barrido se suele emplear para construir los objetos, pero a la hora de visualizarlos lo habitual es pasar a otro tipo de representación (paramétrica o poligonal). Sin embargo, también es posible hacer un trazado de rayos con bastante facilidad.

1.2.4 Modelado CSG (Geometría Sólido Constructiva)

El Modelado CSG es uno de los paradigmas dominantes en el modelado de sólidos. Se trata de instanciar objetos primitivos y combinarlos mediante operaciones booleanas (unión, intersección, resta y otros). La representación de un objeto complejo es un árbol en el que figuran las operaciones y objetos primitivos que definen la forma del objeto. De esta forma tan sencilla se representa un objeto en la memoria o en una base de datos. En este caso el objeto se define a partir de un árbol CSG, que suele recorrerse en profundidad, primero Nodos Primitiva Sólida, luego, Nodos Transformación y por último, Nodos Operación.

Como otros métodos de representación del volumen, el sistema CSG resulta también adecuado para añadir a los objetos propiedades físicas, ya que se está representando directamente su volumen y permite realizar la comprobación de pertenencia al objeto de los puntos en el espacio. El modelo CSG resulta muy adecuado cuando los objetos a representar se forman por combinación de formas básicas sencillas (paralelepípedos, esferas, cilindros y otros), pero puede resultar muy ineficiente para otro tipo de objetos.

La geometría constructiva de sólidos es un método de modelado no ambiguo, con un dominio amplio (condicionado por el repertorio de primitivas disponibles) y válido. La principal debilidad del método es la no unicidad de las representaciones, ya que hay infinitud de formas de representar cualquier sólido. [12]

1.3 Representación de la información. Partición espacial

La Partición Espacial es una forma de almacenar la información de los objetos que llega a tener grandes utilidades, como son la clasificación de puntos, información particular de cada zona del espacio, la clasificación de objetos en una escena y la optimización de la visualización. [13]

Se clasifica en Particiones no Jerárquicas y Particiones Jerárquicas.

Las **Particiones no Jerárquicas** están constituidas por Descomposición en celdas y Enumeración de ocupación espacial, donde la primera se utiliza cuando se tiene ese tipo de estructura o se pretende añadir propiedades para análisis mecánicos de elementos finitos, mientras que la segunda, consiste en la descomposición en voxels (el equivalente de un píxel en tres dimensiones), donde se indican los que pertenecen al objeto y cuáles no.

Es un caso especial de descomposición por celdas en el que éstas son todas idénticas y se organizan en un entramado regular o rejilla. Es frecuente que las celdas sean pequeños cubos alineados en una trama ortogonal en cada uno de los ejes de coordenadas. Se trata de una representación muy utilizada para almacenar datos de dispositivos tipo escáner.

Las **Particiones Jerárquicas** se caracterizan por irse realizando por niveles sucesivos. La estructura de datos que se emplea es un árbol, lo que disminuye el espacio de almacenamiento requerido y hace posible definir a la vez varios niveles de detalle. Podremos acceder a la información o efectuar la visualización recorriendo la estructura solamente hasta el nivel que nos interese.

Esta representación se divide en **Particiones Jerárquicas no ortogonales** (árboles binarios) y **Particiones Jerárquicas ortogonales** (según los ejes del espacio). Para las Particiones Jerárquicas no ortogonales se puede efectuar una división sucesiva del espacio en semiespacios utilizando planos de orientación arbitraria.

Estos planos se organizan en un *árbol de particiones espaciales binarias* o *BSPtree* (*Binary Space Partition tree*), en el que cada nodo representa un plano y sus dos hijos son los dos semiespacios que define. En los nodos terminales del árbol se indica si cada región definida por los sucesivos cortes cae dentro o fuera del objeto ver (Fig. 4).

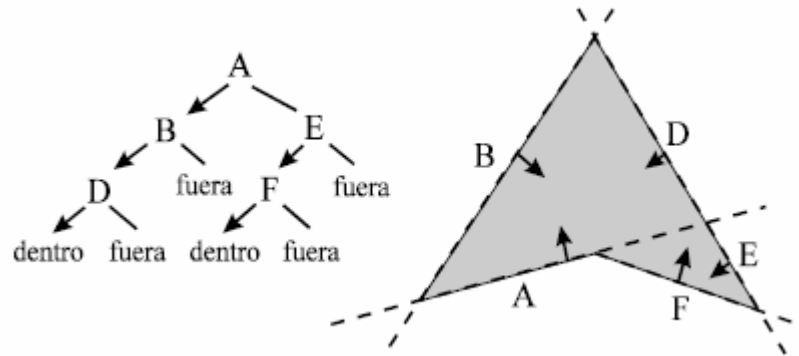


FIG. 4 DESCOMPOSICIÓN DE UNA FIGURA BIDIMENSIONAL EN UN BSP-TREE.

Como los demás métodos de partición espacial, estos árboles binarios permiten efectuar una clasificación de la pertenencia de cualquier punto del espacio al objeto definido. Para ello hay que comenzar por el plano que está en el nodo raíz del árbol. Calculando el signo de la distancia del punto al plano, se sabe si hay que pasar a la rama derecha o a la izquierda del árbol. Se sigue este proceso recursivamente hasta que se llega a un nodo terminal, que indicará si el punto está dentro o fuera.

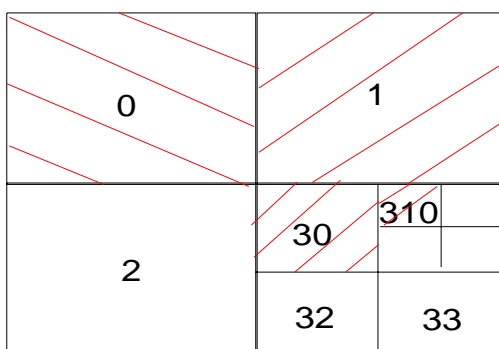
Además de las utilidades comunes a todos los métodos de partición, los árboles binarios han sido frecuentemente utilizados para ordenar los polígonos de una escena según la distancia al observador y poder de esa manera, aplicar el algoritmo del pintor (dibujar los polígonos de mayor a menor distancia), para calcular la ocultación de superficies en una escena 3D. Para visualizar este tipo de objetos podemos emplear el trazado de rayos o bien convertirlos primero en polígonos.

Por sus grandes utilidades, los árboles binarios de particiones resultan especialmente eficientes para representar objetos formados por facetas planas, en los que la representación resulta además, exacta.

En el caso de las **Particiones Jerárquicas ortogonales**, normalmente se realizan utilizando planos en la dirección de los ejes principales. Los espacios producidos

pueden estar completamente llenos, completamente vacíos o tener ocupación parcial. Las sucesivas particiones se realizan según planos que están orientados siguiendo los ejes ortogonales del espacio.

El espacio suele, en este caso, delimitarse mediante una caja ortogonal que contiene completamente al objeto o escena. Si cada partición divide al espacio en dos semiespacios iguales, entonces se estaría de nuevo ante un árbol binario. Si cada partición divide la caja en cuatro partes iguales de forma recursiva, entonces se tiene un *árbol de cuadrantes* o *quadtree*, que se suele usar para figuras planas, ver (Fig. 5).



$$\text{Objeto} = 0+1+30+310$$

FIG. 5 EJEMPLO DE QUADTREE.

En el espacio resulta conveniente la división de cada caja en ocho cajas de tamaño mitad, formándose un *árbol de octantes* u *octree*.

Las particiones también pueden ser **no homogéneas**, cuando las partes resultantes no son iguales.

Partición jerárquica ortogonal no homogénea.

En este caso la descomposición no se realiza por mitades iguales sino desplazando los planos de corte al punto donde se considere óptimo. De esta forma se puede aproximar más rápidamente la forma del objeto, aunque la estructura de datos se complica ligeramente y resulta un poco más difícil de manejar.

Los métodos jerárquicos ortogonales tienen la ventaja de presentar una estructura de datos muy compacta, que se presta a la utilización de algoritmos recursivos muy sencillos. Al ser jerárquicas, permiten una representación multirresolución del objeto y resultan mucho más compactas que la enumeración espacial (voxels).

Como desventaja, resulta difícil llegar a representar a un objeto de forma exacta, ya que éstos raramente tienen todas sus facetas orientadas ortogonalmente. La visualización de un objeto representado por un octree también es difícil, puesto que la transformación en polígonos no es inmediata y el trazado de rayos tampoco es fácil si quiere calcular el valor del vector normal en la superficie del objeto para efectuar la iluminación. [13]

1.3.1 Elementos 2D, árboles cuaternarios o quadtree

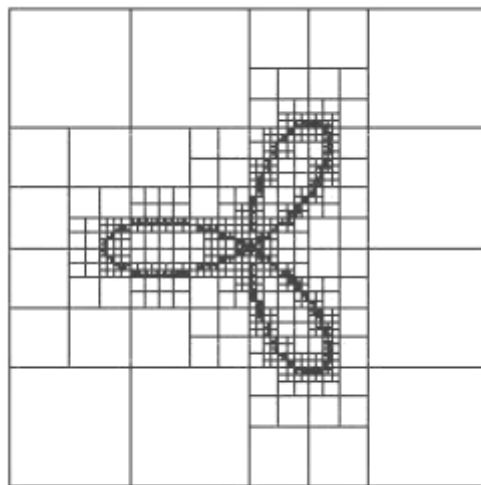


FIG. 6 PARTICIÓN QUADTREE.

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 (Fig. 6) y (Fig. 7). 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. [13]

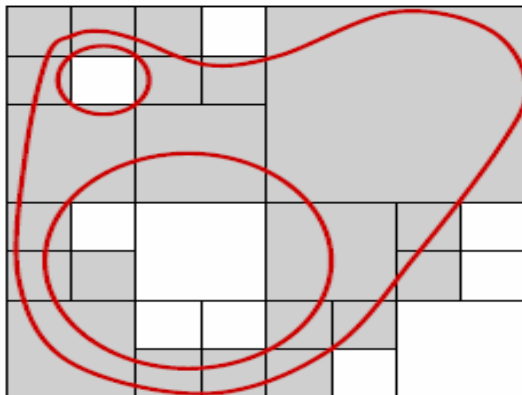


FIG. 7 DIVISIÓN EN SECCIONES.

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. [13]

Cada cuadrante puede estar lleno, parcialmente lleno o vacío. Un cuadrante parcialmente lleno se subdivide recursivamente en subcuadrantes. Este proceso de división continúa hasta que todos los cuadrantes sean homogéneos o hasta alcanzar una profundidad límite previamente determinada, ver (Fig.8). [13]

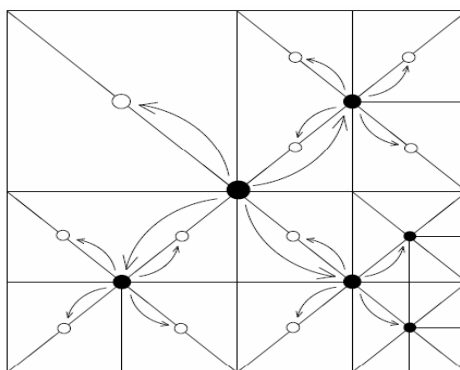


FIG. 8 REPRESENTACIÓN DE CUADRANTES.

Cuando 4 cuadrantes hermanos son homogéneos, se eliminan y su padre se reemplaza por un nodo totalmente lleno o vacío. Cualquier nodo parcialmente lleno en la profundidad límite se clasifica como lleno, con lo que tampoco en este caso existe el concepto de ocupación parcial. [13]

En la siguiente figura se muestra la representación con árbol de cuadrantes parcialmente llenos en los nodos internos y llenos o vacíos en las hojas. Relajando el criterio para representar un nodo como homogéneo, la representación será más compacta aunque menos precisa. [11]

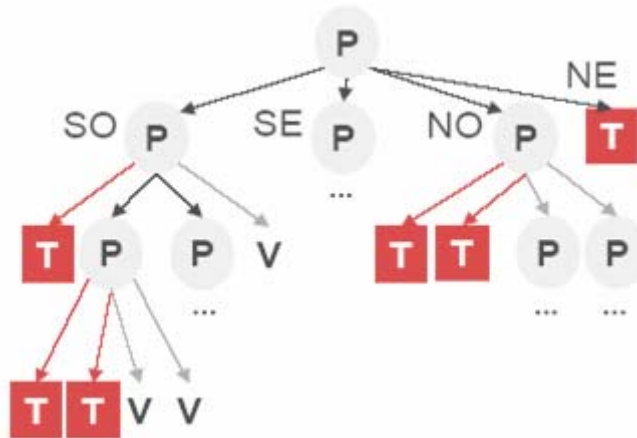


FIG. 9 CUADRANTES PARCIALMENTE LLENOS.

1.3.2 Particiones Jerárquicas 3D, árboles octario u octrees.

Partición jerárquica ortogonal homogénea

Si queremos describir la forma de un objeto con un octree o árbol de octantes tenemos que crear una estructura arbórea en la que, el volúmen ocupado por el objeto se va aproximando recursivamente por medio de cubos o paralelepípedos en general. Se trata de combinar celdas ortogonales cuya longitud de arista se va dividiendo sucesivamente por 2, intentando rellenar lo más aproximadamente posible el espacio del objeto deseado ver (Fig. 10). Los nodos que no caen totalmente dentro o fuera del objeto se seguirán subdividiendo de forma recursiva. [18]

En cada nodo del árbol se puede indicar únicamente si la celda correspondiente cae dentro o fuera del objeto o se puede considerar algún tipo de gradación (parcialmente dentro, 25, 50 y 75 % dentro y otros). La recursión deberá detenerse según algún criterio (por ejemplo, las celdas no pueden tener una arista menor de un cierto valor) se limita directamente la profundidad del árbol a un valor máximo. [18]

Existen métodos para asociar un código a cada posible celda, de manera que exista una notación para enumerar las que componen un cierto objeto.

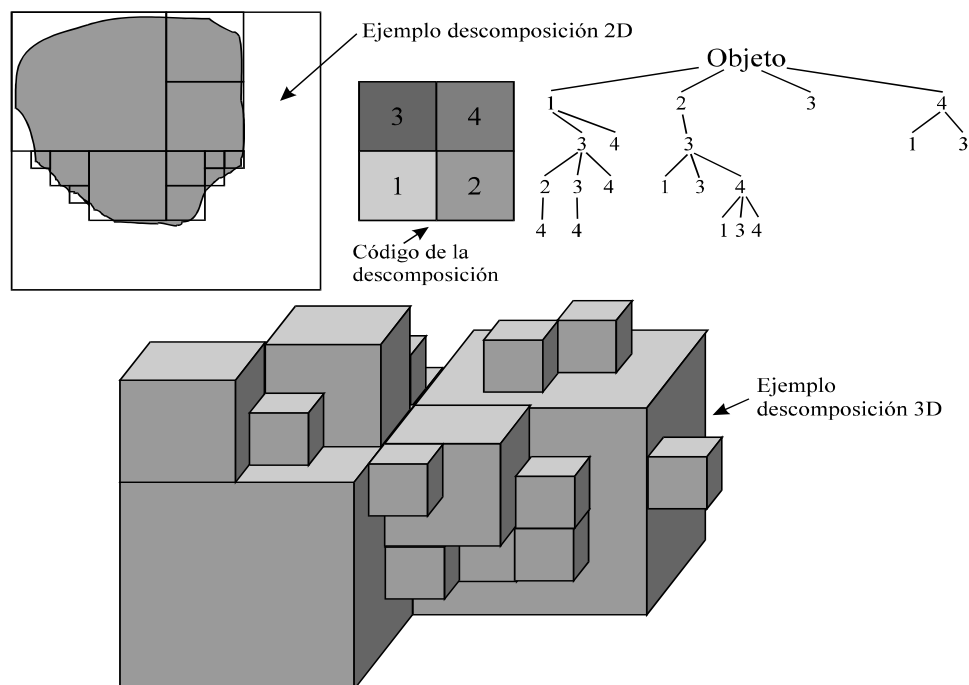


FIG. 10 DESCOMPOSICIÓN DE SUPERFICIES EN QUADTREES Y SÓLIDOS EN OCTREES.

Estos métodos en Particiones Jerárquicas y Áreas de Definición de la Representación suelen emplearse en espacios abiertos, visualización de terrenos, donde las oclusiones no siguen un patrón regular y se deben a la existencia de montañas, casas y otros. Siendo usual utilizar como base la descomposición general del terreno en árboles de tipo *quadtree*, asociado a los niveles del *quadtree* determinados objetos en función de la visibilidad. [13]

Estos árboles son una variante jerárquica de la enumeración espacial, como resultado de un derivado de los quadtrees. Los octrees utilizan la división recursiva del espacio en ocho octantes, (por ello el nombre de octree), que son colocados en una estructura lógica de árbol octario (de ocho hojas), es decir, se basa en la formación de manera recursiva de un árbol de 8 nodos, ver (Fig. 11).

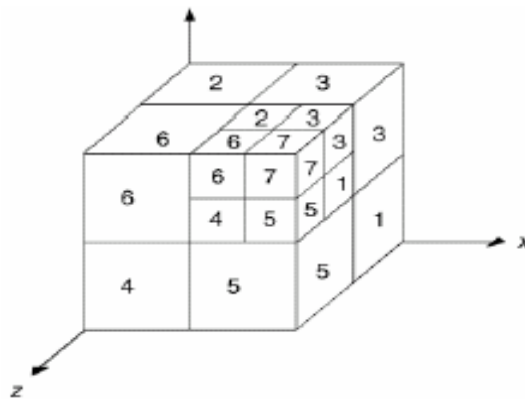


FIG. 11 EJEMPLO DE PARTICIÓN DE UN ÁRBOL OCTREE.

El límite de crecimiento del árbol es hasta el nivel 5 y el máximo de polígonos definido para cada nodo es 1000. Cada triángulo en este procesamiento previo, se ubica en el nodo que le corresponde según las coordenadas de sus vértices, ver ejemplos a continuación de una pirámide representada con un octree, donde los distintos colores representan distintos niveles del árbol. [13]

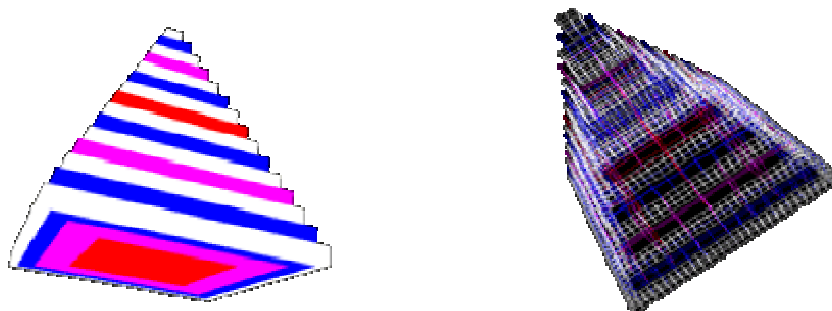


FIG. 12 Y 13 REPRESENTACIÓN DE PIRÁMIDE CON UN OCTREE.

El objetivo de este algoritmo es reducir la cantidad de triángulos que se van a dibujar a partir de una división del terreno en varias regiones. Llegado el momento, se dibujan solo aquellas regiones que se encuentran en la zona de visibilidad.

La idea fundamental es un algoritmo divide y vencerás. Así como en el quadtree el plano se divide en cuadrantes. Si se usa completo se escribe como "full", sino como "empty", si es parcialmente usado se divide, hasta una profundidad determinada. Un octree es semejante con tres dimensiones, subdividido en octantes de manera recursiva. El número de nodos es proporcional al perímetro en un quadtree o superficie del objeto en un octree, en ambos casos es proporcional a la medida de la frontera.

Con esta técnica se minimiza la carga del microprocesador, ya que no se utilizan las sucesivas llamadas a funciones para la definición de los datos de los vértices del tipo glVertex(). La figura a continuación, muestra la ventaja de evitar la carga que producen sobre el procesador las diferentes formas de invocar el dibujo de triángulos. Esta recursividad va por todos los nodos finales asignándoles una identificación de Lista. De esta forma se llama a esta identificación cuando se dibuja, incrementando considerablemente la velocidad en cuadros/segundo.

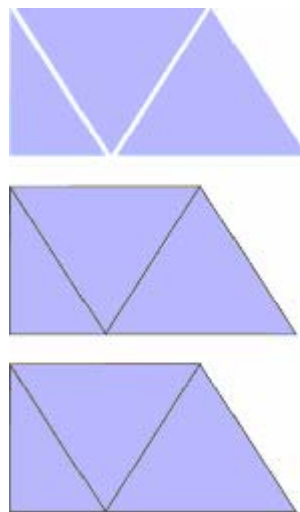


FIG. 14 PROPIEDADES DE LOS OCTREES.

Los octrees presentan un modelo de representación poderoso, son modelos de representación aproximada y pueden ser exactos para ciertos objetos, no requieren de

una conectividad especial, todos los octrees son representaciones válidas de algún sólido. No presentan ambigüedad y unicidad, hasta los límites de resolución, todos los octrees no ambiguos definen un sólido.

En cuanto a los lenguajes de descripción, los octrees son formados mediante una conversión de otras representaciones, como las constructivas y el procesamiento de imágenes. Los octrees y los quadrees son formados directamente de imágenes digitales de datos, por medio de un proceso de "raster". Presentan alta consistencia, en general el número de nodos que un octree representa es proporcional al área del objeto.

En promedio, un octree fácilmente puede medir más de un millón de bytes de memoria, soporta algoritmos cerrados para los problemas de traslación, rotación y operaciones booleanas y son sencillos computacionalmente, muchos algoritmos de los octrees toman la forma de transversal, donde las operaciones son relativamente fáciles para cada nodo del árbol.

1.4 Algoritmos para el seguimiento de terrenos

En la industria de los juegos para computadoras, varias técnicas han sido utilizadas para hacer seguimiento de terrenos, entre ellas se destacan el CLOD (Continuous Level Of Detail) y QOTA (Quick Oriented Terrain Algorithm).

1.4.1 CLOD

Los algoritmos de nivel continuo de detalle *CLOD* (*Continuous Level of Detail*), representan una malla dinámica de polígonos y proporcionan triángulos extra en las zonas que requieren un mayor nivel de detalle. [15]

Los algoritmos *CLOD* requieren más investigación, son más complejos de implementar y exigen más ciclos de *CPU* que los algoritmos de fuerza bruta. [15]

Existen varios algoritmos *CLOD*, entre los cuales destacan los tres siguientes:

- Algoritmo *quadtree* de Stefan Roettger.
- Algoritmo *ROAM* (*Real-time Optimally Adapting Meshes*) de Mark Duchaineau.
- Algoritmo *geomipmapping* de Willem H. Boer.

Los avances en los algoritmos de nivel continuo de detalle han traído esta tecnología de gran alcance a la vanguardia de la investigación de la optimización de los gráficos 3D. [17]

1.4.2 QOTA

Para mantener objetos móviles en la tierra de ambientes virtuales, es necesario encontrar los puntos donde estos objetos deben entrar en contacto con el terreno. Esto se hace a menudo usando la detección de la colisión, sin embargo, esto es ineficaz, porque la detección general de la colisión soluciona un problema que es intrínsecamente más complejo que la simple determinación de puntos de contacto del terreno. [10]

Para dar solución a esta problemática se crea el algoritmo de rápida orientación del terreno (QOTA), el cual se centra solamente en el problema de intersecar líneas de una orientación predeterminada con un modelo del terreno, proporciona la ayuda rápida para el seguimiento del terreno. [10]

QOTA fue desarrollado como parte de La Plataforma de Scalable para los Ambientes Networked interactivos grandes (Spline), para permitir el uso de un terreno complejo sin incurrir en costos de cómputo desrazonables. [10]

Este algoritmo utiliza un paso del proceso previo que clasifique los polígonos del terreno en un quadtree y agregue las cajas de limitación y los parámetros de la ecuación del borde del polígono para acelerar la comprobación de la contención del polígono. Además del seguimiento de terreno, QOTA es útil para detectar ciertas clases limitadas de detección de la colisión y determinar la contención. [10]

Se puede concluir que QOTA se utiliza para determinar no sólo si hay un polígono del piso debajo de un punto de prueba dado, sino también si hay un polígono del techo sobre el punto de prueba. Si existen ambos polígonos, entonces el punto de prueba está dentro del volumen 3D especificado. [10]

Existe una confianza fundamental en coherencia espacial y temporal para el uso de este algoritmo. La razón dominante por la que QOTA es más rápido que otros algoritmos es que ataca un problema más simple que se puede reducir casi enteramente a operaciones 2D en vez de resolver operaciones 3D. [10]

Otros usos de QOTA

Aunque está diseñado sobre todo para el seguimiento de terrenos, QOTA se puede utilizar para una variedad de tareas adicionales. Por ejemplo en Spline, se utiliza para una clase estricta de detección de la colisión y de la determinación rápida de la contención 3D. [\[24\]](#)

QOTA se puede utilizar para detectar colisiones simples con obstáculos fijos creando terrenos con los agujeros en ellos. Por ejemplo, si un objeto procura moverse en uno de estos agujeros, QOTA puede determinar rápidamente que el caerá en este y puede reaccionar rechazando la entrada del objeto en el agujero. [\[24\]](#)

1.5 Método matemático para el ajuste de superficies

Para la aproximación de superficies se impone como mejor opción, la utilización del método matemático denominado desde siglos pasados: Método de Mínimos cuadrados. [19]

1.5.1 Método de mínimos cuadrados

Método de cálculo ideado por Carl Friedrich Gauss, publicado en 1809 que consiste en imponer la condición de mínimo a la suma de los cuadrados de las diferencias de una medición sin sesgo. [19]

Es la técnica de optimización matemática mas utilizada mundialmente para ajustar superficies, pues brinda soluciones eficientes y elegantes.

Consiste en una serie de pasos utilizados para aproximar un conjunto de puntos a un modelo determinado, el cual puede ser lineal, cuadrático, exponencial u otros. A continuación se explica de manera detallada cada paso lógico para alcanzar la solución requerida. [19]

Pasos a seguir:

- Para el caso de un modelo lineal:

$$g(x) = C_1g_1(x) + C_2g_2(x) + \dots + C_n g_n(x)$$

- Se calcula la desviación cuadrática de la siguiente forma:

$$D = \sum_{j=1}^m [C_1g_1(x_j) + C_2g_2(x_j) + \dots + C_n g_n(x_j) - y_j]^2$$

Como D es una función cuadrática de las variables C_1, C_2, \dots, C_n , posee un punto mínimo que se obtiene igualando a cero las derivadas de D respecto a C_1, C_2, \dots, C_n .

- Derivando parcialmente con respecto a C_i ($i= 1, 2, \dots, n$) e igualando a cero:

$$\frac{\partial D}{\partial C_i} = \sum_{j=1}^m 2[C_1 g_1(x_j) + C_2 g_2(x_j) + \dots + C_n g_n(x_j) - y_j] g_i(x_j) = 0$$

Pudiéndose escribir de esta forma, separando en sumas y extrayendo factores comunes:

$$C_1 \sum_{j=1}^m g_1(x_j) g_i(x_j) + C_2 \sum_{j=1}^m g_2(x_j) g_i(x_j) + \dots + C_n \sum_{j=1}^m g_n(x_j) g_i(x_j) = \sum_{j=1}^m y_j g_i(x_j)$$

- Dando a i los valores $1, 2, \dots, n$, se obtiene el siguiente sistema lineal de ecuaciones:

$$\begin{aligned} \left[\sum_{j=1}^m g_1^2(x_j) \right] C_1 + \left[\sum_{j=1}^m g_1(x_j) g_2(x_j) \right] C_2 + \dots + \left[\sum_{j=1}^m g_1(x_j) g_n(x_j) \right] C_n &= \sum_{j=1}^m g_1(x_j) y_j \\ \left[\sum_{j=1}^m g_2(x_j) g_1(x_j) \right] C_1 + \left[\sum_{j=1}^m g_2^2(x_j) \right] C_2 + \dots + \left[\sum_{j=1}^m g_2(x_j) g_n(x_j) \right] C_n &= \sum_{j=1}^m g_2(x_j) y_j \\ &\vdots \\ \left[\sum_{j=1}^m g_n(x_j) g_1(x_j) \right] C_1 + \left[\sum_{j=1}^m g_n(x_j) g_2(x_j) \right] C_2 + \dots + \left[\sum_{j=1}^m g_n^2(x_j) \right] C_n &= \sum_{j=1}^m g_n(x_j) y_j \end{aligned}$$

[2]

1.6 Lenguaje de programación

Existen tres lenguajes de programación principales para desarrollar aplicaciones gráficas profesionales en 3D y estos son el Lenguaje Ensamblador, el Lenguaje C y el Lenguaje C plus plus (C++), donde este último ha ido evolucionando y en la actualidad existen muchas herramientas ya probadas, que hacen dicho lenguaje mucho más potente que otros. A éstos se han unido recientemente los lenguajes de programación Java y C#, como una opción para el desarrollo de este tipo de aplicaciones. [4] y [6]

Fue seleccionado el lenguaje de programación C++ para la implementación del módulo de seguimiento de terrenos, pues la herramienta está programada en este lenguaje, obteniéndose magníficos resultados.

C++ es un potente lenguaje de programación que apareció en 1980 ampliando las ventajas, flexibilidad y eficacia del C. Es la evolución de C adaptada a la programación orientada a objetos, que ha pulido el control del manejo de tipos de datos y otras características que ayudan a la programación libre de errores. [16]

Además, es el lenguaje en el que se tiene mayor experiencia por parte del equipo que ha trabajado en el desarrollo de las funcionalidades de la herramienta, pues están familiarizados con códigos fuentes estandarizados, código eficiente y estimaciones más realistas en la planificación de la codificación.

Conclusiones

En este capítulo que concluye se definieron los principales conceptos que ayudarán a comprender el desarrollo del módulo. Se fundamentaron las principales técnicas, tecnologías y tendencias más utilizadas en la actualidad mundial en cuanto a estos temas, propiciando un mayor conocimiento para determinar cuáles son mejores opciones para emplear en el diseño e implementación del módulo de seguimiento de terrenos.

Se comprendió el funcionamiento de las propiedades que tiene desarrollada la biblioteca básica del proyecto para hacer seguimiento de terrenos, para de esta forma darle continuidad y aportar el realismo requerido por los simuladores y juegos que utilizan dicha herramienta.

Capítulo 2 Descripción de la Solución Propuesta

Introducción

En el presente capítulo se proponen soluciones técnicas para el desarrollo del módulo de Seguimiento de Terrenos, así como la descripción de las mejoras propuestas para lograr un mayor realismo en los sistemas de RV. Ya se comienza a tener una perspectiva de lo que se implementará, iniciándose sobre la base de las dificultades, necesidades y características organizacionales del cliente. Se destacan las reglas establecidas en el negocio, se identifican el o los actores y trabajadores del mismo. Se plantean los requerimientos funcionales y no funcionales con que debe contar el sistema.

Se representan los casos de uso del sistema y su relación, mediante el diagrama de casos de uso del sistema, utilizándose para su modelado el Lenguaje Unificado de Modelación (UML), además de la especificación de los mismos en formato expandido.

2.1 Soluciones Técnicas

La herramienta a la cual se acoplará el módulo de seguimiento de terrenos tiene actualmente implementada la representación de los objetos puestos en escena usando los **Modelos de superficies planas** (*poligonales*), debido a que este tipo de representación poligonal mediante triángulos propicia un óptimo aprovechamiento del HW, por lo que no será modificado.

Las mallas de los terrenos se encuentran almacenadas en estructuras **Quadtree**, donde cada nodo raíz se ramifica en cuatro nodos ramas, optimizando los requisitos de almacenamiento además del procesamiento para cálculos geométricos, debido a su estructura de árbol, por lo que tampoco sufrirá modificación.

Para encontrar los puntos donde deben entrar en contacto con el terreno los objetos que se moverán en el Mundo, la herramienta tiene implementado el algoritmo **QOTA**. Este algoritmo proporciona eficaz solución, pues resuelve el problema rápidamente y más simple que otros algoritmos, reduciéndose casi enteramente a operaciones 2D, por lo que tampoco sufrirá cambios.

Adicionalmente al seguimiento de terrenos con 3 puntos de apoyo, ya implementado en la herramienta, se brindará la posibilidad de hacer seguimiento con 1 y con 4 puntos de apoyo sobre el terreno. Agrupando estas variantes en el **módulo de seguimientos de terrenos** que será acoplado a la biblioteca del proyecto.

Se implementará el seguimiento de terrenos con **1 punto de apoyo**, ya que es muy sencillo y en determinadas situaciones donde no es importante la orientación del objeto, sino solamente su posición (altura sobre el terreno), puede ser muy útil, por ejemplo, si se aplica optimización por niveles de detalles, quizás no sea importante aplicar seguimiento con 3 o 4 puntos para los objetos alejados del observador.

Se implementará el seguimiento con **4 puntos de apoyo**, porque de esta forma aporta más realismo para juegos y simuladores que utilizan la herramienta, debido a que en su generalidad los objetos que hacen seguimiento son autos y de esta forma, se buscan los puntos de apoyo en sus 4 gomas, proporcionando movimientos reales.

Quedando conformado un módulo de seguimiento de terrenos, el cual permite al usuario escoger la cantidad de puntos con los que desea hacer el seguimiento, según lo real que desea se visualice la escena (1, 3 o 4 puntos de apoyo sobre el terreno).

En el caso de 4 puntos de apoyo, es necesario ajustar estos a un mismo plano, para que luego se puedan hacer los cálculos pertinentes para llevar a cabo el seguimiento de terrenos, para esto se empleará el método matemático de los **Mínimos cuadrados**. Este método facilita un ajuste eficiente y práctico de dichos puntos en un plano 3D, proporcionando una solución elegante para nuestro problema.

El diseño e implementación se corresponderá con la filosofía de Programación Orientada a Objetos, a través de funciones programadas puramente en el **lenguaje de desarrollo C++**.

2.2 Reglas del negocio

Las reglas del negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio. De ahí la importancia de saber identificarlas, evaluarlas teniendo en cuenta su relevancia dentro del campo de acción que se está modelando e implementarlas en la solución final.

A continuación quedan definidas las reglas para este dominio de trabajo, orientadas a los diseñadores gráficos que construyan los terrenos de los entornos.

- Un mismo terreno no puede cruzarse sobre sí mismo, es decir, el terreno por extendido que sea, no debe hacer un elevado que cruce por encima del propio terreno.
- Los terrenos tienen que ser regulares, es decir, deben ser lo menos abruptos posible.

2.3 Modelo del dominio

El diagrama que a continuación se muestra representa el modelo de dominio del módulo de seguimiento de terrenos. Este modelo ayudará a los programadores y otros usuarios similares en la comprensión y el análisis de la relación conceptual a diseñar.

Este módulo no se ajusta al concepto de negocio en toda su extensión, sino que responde al concepto de modelo de dominio, donde se comporta como tal, conociendo qué información es sensible o relevante en el dominio y cómo es el ambiente en el que será acoplado.

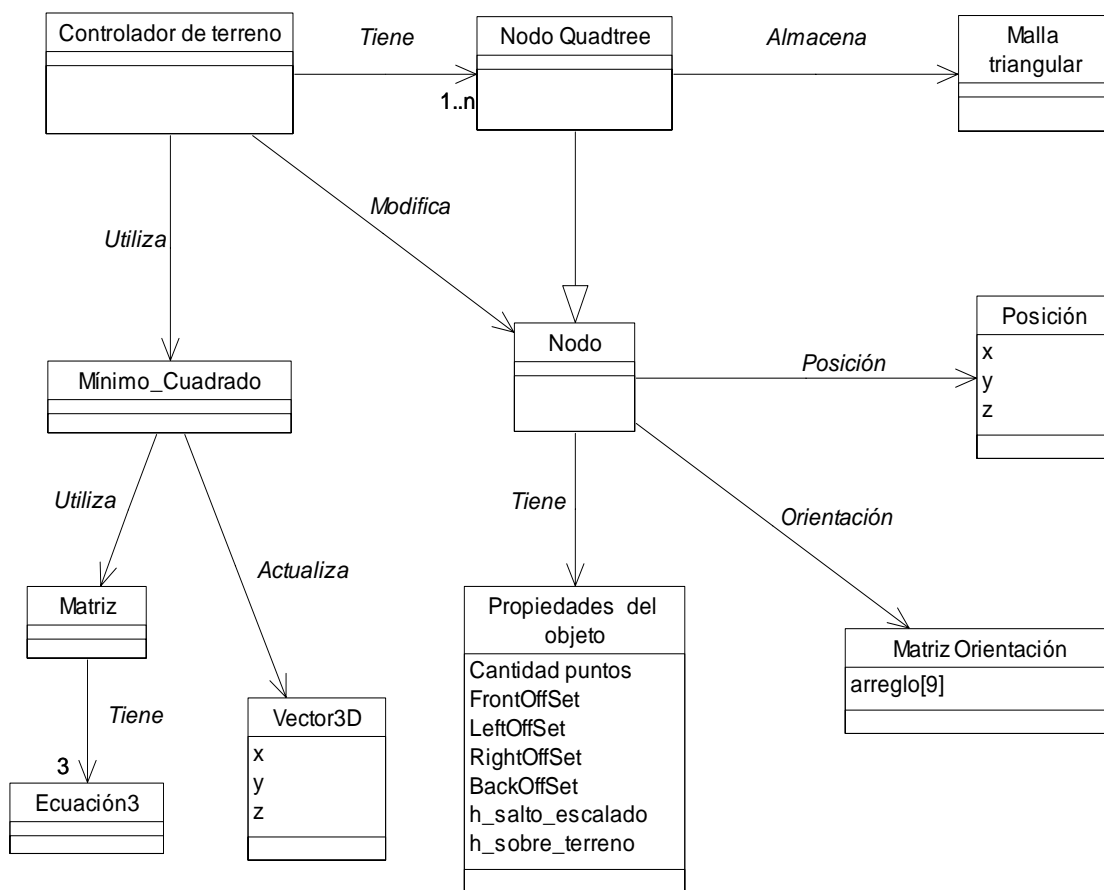


FIG. 15 MODELO DEL DOMINIO

2.3.1 Glosario de términos del modelo del dominio.

A continuación se presentan un conjunto de conceptos que conforman el glosario de términos del modelo del dominio, con el propósito de facilitar un mayor entendimiento de los términos manejados en el diagrama correspondiente.

Los conceptos **Nodo**, **Nodo Quadtree**, **Malla triangular**, **Posición**, **Vector3D** y **Matriz Orientación** no pertenecen a la modelación realizada como parte de esta investigación, estos forman parte de modelaciones ya implementadas en la herramienta los cuales usamos o modificamos según sea necesario.

Nodo: Estructura que representa a los objetos de la escena y que contiene datos como la posición, orientación y propiedades físicas.

Controlador de terreno: Encargado de modificar los nodos especificados para hacer el desplazamiento sobre el terreno.

Mínimo Cuadrado: Método matemático encargado de posibilitar el ajuste de 4 puntos a un plano, utiliza una Matriz y las coordenadas de los puntos a ajustar.

Vector3D: Encargado de poseer las coordenadas de los puntos de apoyo a ajustar en 3D.

Propiedades del Objeto: Encargado de almacenar las propiedades físicas del objeto que se utilizarán en el seguimiento del terreno.

Matriz de orientación: Estructura encargada de almacenar los vectores de orientación del nodo, left, up, dir.

Posición: Encargado de almacenar las coordenadas x, y, z de la posición.

Malla triangular: Estructura que almacena los triángulos de la malla del terreno.

Nodo quadtree: Estructura que almacena la malla triangular y posee las propiedades de los nodos.

Ecuación3: Encargado crear ecuaciones con tres incógnitas.

Matriz: Estructura que almacena tres ecuaciones y es capaz de resolver un sistema de ecuaciones con las mismas.

2.4 Captura de requisitos

Los requisitos constituyen capacidades o condiciones que el sistema debe cumplir, facilitando el entendimiento entre usuarios y clientes del módulo a diseñar.

2.4.1 Requisitos funcionales

Los siguientes requisitos establecen las funcionalidades e instrucciones que el módulo debe cumplir en su implementación.

1- Crear controlador de terrenos.

1.1- Inicializar la cantidad de terrenos.

1.2- Inicializar el coeficiente de rozamiento de un terreno.

2- Destruir controlador de terrenos.

2.1- Destruir lista de terrenos.

3- Crear propiedades físicas de los objetos.

3.1- Inicializar el desplazamiento derecho.

3.2- Inicializar el desplazamiento izquierdo.

3.3- Inicializar el desplazamiento delantero.

3.4- Inicializar el desplazamiento trasero.

3.5- Inicializar la cantidad de puntos de la base.

3.6- Inicializar la altura de salto o escalado.

3.7- Inicializar la altura sobre el terreno.

4- Asociar propiedades físicas a un nodo.

4.1- Asignar al objeto las propiedades físicas.

4.2- Poner al objeto una referencia a las propiedades físicas.

5- Configurar propiedades físicas de los objetos.

5.1- Definir si el seguimiento se hará con 1, 3 o con 4 puntos de apoyo.

5.1.1- Verificar que la cantidad de puntos especificada sea 1, 3 o 4.

5.1.2- Asignar por defecto cantidad de puntos 4.

5.2- Definir altura sobre el terreno.

5.3- Definir altura de escalado.

5.4- Configurar los 4 desplazamientos respecto al centro del objeto.

5.4.1- Definir valor del desplazamiento delantero.

5.4.2- Definir valor del desplazamiento a la izquierda.

5.4.3- Definir valor del desplazamiento a la derecha.

5.4.4- Definir valor del desplazamiento trasero.

5.5- Configurar los coeficientes de rozamiento de un objeto.

6 - Determinar los puntos de apoyo. (Para 1, 3 y 4 puntos).

6.1- Tomar del nodo el vector Localización.

6.2- Tomar del nodo el vector Dirección.

6.3- Tomar del nodo el vector Izquierda.

6.4- Tomar del nodo el vector Normal.

6.5- Tomar del nodo el desplazamiento delantero.

6.6- Tomar del nodo el desplazamiento izquierdo.

6.7- Tomar del nodo el desplazamiento derecho.

6.8- Tomar del nodo el desplazamiento trasero.

6.9- Tomar la cantidad de puntos con los que se hará el seguimiento.

6.10- Calcular los puntos de apoyo.

7 - Permitir el paso de un terreno a otro.

7.1- Buscar en qué terreno está cada punto.

8- Ubicar cada punto sobre el terreno.

9- Sumar a cada punto la altura sobre el terreno.

10- Actualizar el estado geométrico de un objeto (nodo) sobre el terreno.

10.1- Tomar los puntos de apoyo actualizados.

10.2- Tomar la cantidad de puntos con los que se hará el seguimiento.

10.3- Asignar al nodo el vector Localización actualizado.

10.4- Asignar al nodo el vector Dirección actualizado.

10.5- Asignar al nodo el vector Izquierda actualizado.

10.6- Asignar al nodo el vector Normal actualizado.

11- Actualizar la normal del terreno de un objeto.

12- Actualizar el coeficiente de rozamiento para un objeto.

13- Informar el resultado del seguimiento de terrenos.

- 14- Calcular el ajuste de los 4 puntos por el método matemático de mínimos cuadrados.
- 14.1- Tomar los valores de los vectores de desplazamiento del objeto.
 - 14.2- Calcular la ecuación número 1 de la matriz con los 4 puntos entrados.
 - 14.3- Calcular la ecuación número 2 con los 4 puntos entrados.
 - 14.4- Calcular la ecuación número 3 con los 4 puntos entrados.
 - 14.5- Resolver las ecuaciones 1 y 2 eliminando las X.
 - 14.6- Resolver las ecuaciones 1 y 3 eliminando las X.
 - 14.7- Resolver el sistema de ecuaciones resultantes de los anteriores cálculos, eliminando las Y.
 - 14.8- Obtener el valor de la constante C de la ecuación del plano.
 - 14.9- Calcular la constante B de la ecuación del plano teniendo C.
 - 14.10- Calcular la constante A de la ecuación del plano teniendo B y C.
 - 14.11- Obtener la ecuación del plano equilibrado.
 - 14.12- Hallar el valor t de la recta paramétrica para el punto 1 en el espacio, teniendo el punto y la normal del plano.
 - 14.13- Proyectar el punto 1 en el plano equilibrado.
 - 14.14- Hallar el valor t de la recta paramétrica para el punto 2 en el espacio, teniendo el punto y la normal del plano.
 - 14.15- Proyectar el punto 2 en el plano equilibrado.
 - 14.16- Hallar el valor t de la recta paramétrica para el punto 3 en el espacio, teniendo el punto y la normal del plano.

- 14.17- Proyectar el punto 3 en el plano equilibrado.
- 14.18- Hallar el valor t de la recta paramétrica para el punto 4 en el espacio, teniendo el punto y la normal del plano.
- 14.19- Proyectar el punto 4 en el plano equilibrado.
- 14.20- Devolver arreglo con los 4 puntos ajustados al plano anteriormente equilibrado.

2.4.2 Requisitos no funcionales

Los siguientes requisitos son propiedades o cualidades que el módulo debe tener, estableciendo de esta forma aspectos que regulan el comportamiento de este.

- Requerimientos de Hardware

Compatibilidad con tarjetas gráficas de la familia NVIDIA (Geforce 3, Geforce 4, FX 5200).

- Restricciones en la implementación

Se utilizará el lenguaje de programación C++ el cual se regirá por la filosofía de Programación Orientada a Objetos.

- Requerimientos de Seguridad

Disponibilidad: Se garantizará el acceso a la información y los dispositivos o mecanismos utilizados para lograr la seguridad no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado.

- Requerimientos de Usabilidad

Para los niveles apropiados de usabilidad se requiere de un personal altamente calificado en el lenguaje de programación C++, con conocimiento básico sobre la “Herramienta de Desarrollo para Sistemas de Realidad Virtual”. Todo será implementado con terminologías del idioma Inglés.

- Requerimientos de Soporte

Compatibilidad con el Sistema Operativo Windows.

- Rendimiento

Como aplicación de tiempo real, debe tener alto grado de velocidad de procesamiento o cálculo, tiempo de respuesta y de recuperación y disponibilidad.

- Legales

Se regirá por las normas ISO 9000.

2.5 Modelo de casos de usos del sistema

En esta sección se reconocen el o los posibles actores del sistema a desarrollar y se conciben los casos de uso del sistema, procesos en los cuales se dan resultados de valor para un actor determinado. Además, se seleccionan los casos de uso correspondientes al primer ciclo de desarrollo para hacerles sus especificaciones textuales en formato expandido.

2.5.1 Actores del sistema

Los actores son los roles que un usuario o usuarios del sistema llevan a cabo en algún momento del tiempo. También pueden ser otros sistemas con los que el sistema en proceso de modelado tiene interacción. Estos estimulan al sistema con eventos de entrada o la recepción de algún resultado que este produzca.

En este caso particular el sistema que va a hacer uso de la biblioteca (SceneToolKit), constituye el actor del sistema, específicamente será llamado (sistema usuario).

TABLA 1 ACTOR DEL SISTEMA.

Actores	Justificación
Sistema que va a hacer uso de la herramienta SceneToolKit: sistema usuario	Es el que se beneficiará con las funcionalidades que brinda el módulo de clases, a grosso modo: cargar uno o varios terrenos y ajustar objetos en la escena para lograr el seguimiento de terrenos, configurando el nivel de realismo que desee.

2.5.2 Descripciones de casos de uso

Los casos de uso describen el comportamiento del sistema desde el punto de vista del usuario. Por lo tanto, se establece un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades que debe cumplir el módulo a implementar. Los casos de uso pueden incluir funcionalidades de otros casos de uso manteniendo sus comportamientos.

Descripciones de casos de uso propuestos:

TABLA 2 CASO DE USO DEL SISTEMA, MANIPULAR CONTROLADOR.

CU - 1	Manipular controlador.
Actor	Sistema usuario.
Descripción	Permite crear un controlador de seguimiento de terrenos cargando un terreno.
Referencia	R1,R1.1, R1.2, R2, R2.1

TABLA 3 CASO DE USO DEL SISTEMA, MANIPULAR PROPIEDADES FÍSICAS DE LOS OBJETOS.

CU - 2	Manipular propiedades físicas de los objetos.
Actor	Sistema usuario.
Descripción	Permite crear y asociar propiedades físicas a un objeto (nodo).
Referencia	R3, R3.1, R3.2, R3.3, R3.4, R3.5, R3.6, R3.7, R4, R4.1, R4.2

TABLA 4 CASOS DE USO DEL SISTEMA, CONFIGURAR PROPIEDADES DE LOS OBJETOS.

CU - 3	Configurar propiedades de los objetos.
Actor	Sistema usuario.
Descripción	Permite configurar las propiedades físicas a un objeto (nodo), además de los parámetros por los que se registrará el seguimiento de terrenos.
Referencia	R5, R5.1, R5.1.1, R5.1.2, R5.2, R5.3, R5.4, R5.4.1, R5.4.2, R5.4.3, R5.4.4, R5.5

TABLA 5 CASOS DE USO DEL SISTEMA, ACTUALIZAR OBJETO.

CU - 4	Actualizar Objeto
Actor	Sistema usuario.
Descripción	Permite la actualización de un objeto (nodo) en la escena, además se actualizan los valores del coeficiente de rozamiento y normal del terreno.
Referencia	R7, R7.1, R8, R9, R10, R10.1, R10.2, R10.3, R10.4, R10.5, R10.6, R11, R12, R13, CU extendido Ajustar 4 puntos y CU incluido Calcular puntos de apoyo.

TABLA 6 CASOS DE USO DEL SISTEMA, CALCULAR PUNTOS DE APOYO.

CU - 5	Calcular puntos de apoyo
Actor	CU Actualizar Objeto.
Descripción	Determina los puntos de apoyo con los que se hará el seguimiento de terrenos.
Referencia	R6, R6.1, R6.2, R6.3, R6.4, R6.5, R6.6, R6.7, R6.8, R6.9, R6.10.

TABLA 7 CASOS DE USO DEL SISTEMA, AJUSTAR 4 PUNTOS.

CU - 6	Ajustar 4 puntos
Actor	CU Actualizar Objeto.
Descripción	Dado 4 puntos, estos son ajustados para que se pueda hacer el seguimiento de terrenos (para 4 puntos de apoyo).
Referencia	R14, R14.1, R14.2, R14.3, R14.4, R14.5, R14.6

2.5.3 Modelo de casos de uso

El modelo de casos de uso describe la funcionalidad propuesta del nuevo sistema, en este caso, del módulo de seguimiento de terrenos. Se describen aquellos aspectos del sistema que son relevantes al propósito del modelo y a un apropiado nivel de detalle, dando una visión general de las relaciones de las funcionalidades o procesos fundamentales.

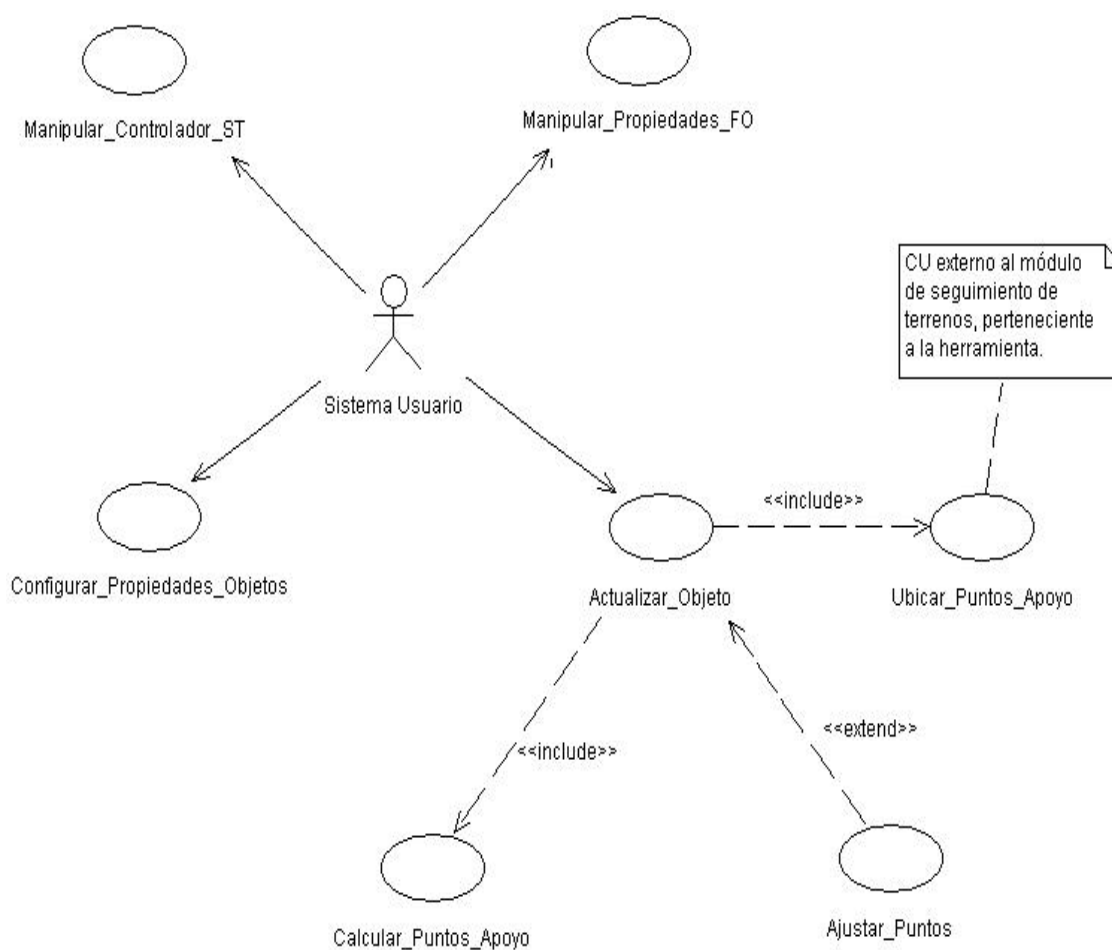


FIG. 16 DIAGRAMA DE CASOS DE USOS DEL SISTEMA.

El Sistema Usuario inicia los CU Manipular controlador de seguimiento de terrenos, Manipular propiedades físicas de los objetos, Configurar propiedades físicas de los objetos y Actualizar Objeto, los cuales brindan las siguientes funcionalidades que serán desarrolladas.

El primer CU crea un controlador para los objetos que van a hacer seguimiento de terrenos y lo destruye.

El segundo CU asocia las propiedades físicas creadas al objeto (nodo) que se desplazará por el mundo virtual.

El tercer CU define con cuántos puntos se hará el seguimiento y la altura que tomarán los puntos sobre el terreno después de actualizados, la altura de escalado, el desplazamiento del objeto y el coeficiente de rozamiento de un objeto.

El CU Actualizar objeto, invoca al CU incluido Calcular puntos de apoyo, el cual calcula los puntos con los que hará el seguimiento, los devuelve y el CU padre modifica el estado de cada nodo, actualiza la posición y orientación de un objeto sobre el terreno, invoca al CU externo Ubicar puntos de apoyo, el cual ubica los puntos sobre el terreno.

Seguidamente el CU Actualizar objeto, llama al CU extendido Ajustar 4 puntos, donde este se encarga de ajustarlos a un mismo plano equilibrado y luego los actualiza ya ajustados y proyectados al plano, listos para hacer el seguimiento de terrenos.

2.5.4 Especificación de los casos de uso en formato expandido

Cada caso de uso tiene una descripción que describe la funcionalidad que se construirá en el sistema propuesto, las tablas presentadas a continuación forman parte de este formato expandido, donde se argumentan con mayor profundidad los flujos operacionales de cada caso de uso.

TABLA 8 CU EXPANDIDOS, MANIPULAR CONTROLADOR.

Caso de Uso	
CU- 1	Manipular Controlador
Actores	Sistema Usuario
Propósito	Controlar el seguimiento de terrenos de los objetos en el mundo.
Resumen	Se inicia cuando el Sistema Usuario solicita cargar una malla triangular (terreno), creándose un controlador de seguimiento de terrenos y finaliza el caso de uso. En caso de querer destruir algún controlador ya creado, es destruida la lista de terrenos y finaliza el caso de uso.
Referencias	R1,R1.1, R1.2, R2, R2.1
Precondiciones	Que exista una malla triangular (terreno).
Curso normal de los eventos	
Acción del actor	Respuesta del sistema
1- Solicita crear o destruir un controlador. - Si solicita crear ir a Sección “Crear” . - Si solicita destruir ir a Sección “Destruir” .	
Sección “Crear”	

Acción del actor	Respuesta del sistema
1- Solicita crear controlador.	
	1.1-Se inicializa la variable correspondiente a la cantidad de terrenos.
	1.2.- Se inicializa la variable correspondiente al coeficiente de rozamiento de un terreno.
Sección “Destruir”	
Acción del actor	Respuesta del sistema
1- Solicita destruir controlador.	
	1.1- Destruir lista de terrenos.
Prioridad	Crítico

TABLA 9 CU EXPANDIDOS, MANIPULAR PROPIEDADES FÍSICAS DE LOS OBJETOS.

Caso de Uso	
CU- 2	Manipular propiedades físicas de los objetos.
Actores	Sistema Usuario
Propósito	Manejar las propiedades físicas de los objetos, desde su creación hasta su control.
Resumen	Se inicia cuando el Sistema Usuario invoca crear propiedades físicas de los nodos o asociar estas propiedades a los nodos correspondientes. Para lo primero se inicializan las variables desplazamiento derecho, desplazamiento izquierdo, desplazamiento delantero, desplazamiento trasero, se inicializan además las variables cantidad de puntos de la base, la altura de salto o escalado y la altura sobre el terreno. Luego para asociar estas propiedades al objeto (nodo), se le pasa una referencia a estas propiedades y finaliza el caso de uso.
Referencias:	R3, R3.1, R3.2, R3.3, R3.4, R3.5, R3.6, R3.7, R4, R4.1, R4.2
Precondiciones	Que exista un objeto (nodo).
Curso normal de los eventos	
Acción del actor	Respuesta del sistema

<p>1- Solicita crear propiedades físicas o asociar propiedades físicas a un nodo.</p> <p>- Si solicita crear propiedades físicas de un nodo ir a Sección “Crear Propiedades”.</p> <p>- Si solicita asociar propiedades físicas a un nodo ir a Sección “Asociar propiedades”.</p>	
Sección “Crear Propiedades”	
Acción del actor	Respuesta del sistema
1- Solicita crear propiedades físicas de un nodo.	
	1.1- Se inicializa la variable correspondiente al desplazamiento derecho.
	1.2- Se inicializa la variable correspondiente al desplazamiento izquierdo.
	1.3- Se inicializa la variable correspondiente al desplazamiento delantero.
	1.4- Se inicializa la variable correspondiente al desplazamiento trasero.
	1.5- Se inicializa la cantidad de puntos de la base (puntos para que se haga el seguimiento de terreno).
	1.6- Se inicializa la altura de salto o escalado.
	1.7- Se inicializa la altura sobre el terreno.
Sección “Asociar Propiedades”	

Acción del actor	Respuesta del sistema
1- Solicita asociar propiedades físicas a un nodo.	
	1.1- Se le asigna al objeto las propiedades físicas.
	1.2- Se le pone al objeto una referencia a las propiedades físicas.
Prioridad	Crítico

TABLA 10 CU EXPANDIDOS, CONFIGURAR PROPIEDADES DE LOS OBJETOS.

Caso de Uso	
CU- 3	Configurar propiedades de los objetos.
Actores	Sistema Usuario
Propósito	Definir las propiedades de los objetos.
Resumen	Se inicia cuando el Sistema Usuario invoca la configuración de una propiedad de un objeto. Se especifica con cuántos puntos se hará el seguimiento, la altura que tomarán los puntos sobre el terreno después de actualizados, la altura de escalado y los desplazamientos del objeto. El caso de uso finaliza cuando queda configurado uno de estos atributos.
Referencias	R5, R5.1, R5.1.1, R5.1.2, R5.2, R5.3, R5.4, R5.4.1, R5.4.2, R5.4.3, R5.4.4, R5.5
Curso normal de los eventos	
Acción del actor	Respuesta del sistema
1- Solicita la configuración de una de las propiedades del objeto, cantidad de puntos de la base, altura sobre el terreno, altura de escalado, desplazamientos del objeto respecto a su centro o coeficiente de	

<p>rozamiento de un objeto.</p> <ul style="list-style-type: none"> - Si solicita cantidad de puntos de la base, ir a Sección “Configurar puntos de la base”. - Si solicita altura sobre el terreno ir a Sección “Configurar altura sobre el terreno”. - Si solicita altura de escalado ir a Sección “Configurar altura de escalado”. - Si solicita los desplazamientos del objeto respecto al centro de este ir a Sección “Configurar desplazamientos del objeto”. - Si solicita coeficiente de rozamiento de un objeto ir Sección “Configurar coeficiente de rozamiento de un objeto”. 	
Sección “Configurar puntos de la base”	
Acción del actor	Respuesta del sistema
1- Especifica la cantidad de puntos de la base.	

	1.1- Si es 1, 3 o 4 se asigna el valor especificado a la variable “cantidad de puntos de la base”.
Curso alternativo de los eventos	
Acción del actor	Respuesta del sistema
	1.1- Se le asigna por defecto la cantidad de puntos 4, al valor de la variable “cantidad de puntos de la base”.
Sección “Configurar altura sobre el terreno”	
Acción del actor	Respuesta del sistema
1- Especifica el valor para la altura sobre el terreno.	
	1.1- Se le asigna el valor a la variable “altura sobre el terreno”.
Sección “Configurar altura de escalado”	
Acción del actor	Respuesta del sistema
1- Especifica un valor para la altura de escalado.	
	1.1- Se le asigna el valor a la variable “altura de escalado”.
Sección “Configurar desplazamientos del objeto”	
Acción del actor	Respuesta del sistema
1- Invoca ajustar la base del objeto en escena especificando los valores de la variable del desplazamiento delantero, desplazamiento a la izquierda, desplazamiento a la derecha y desplazamiento trasero.	
	1.1- Se le asigna el valor indicado a la variable “desplazamiento delantero”.
	1.2- Se le asigna el valor indicado a la variable

	“desplazamiento izquierdo”.
	1.3- Se le asigna el valor indicado a la variable “desplazamiento derecho”.
	1.4- Se le asigna el valor indicado a la variable “desplazamiento trasero”.
Sección “Configurar coeficiente de rozamiento de un objeto”	
Acción del actor	Respuesta del sistema
1- Especifica el valor para coeficiente de rozamiento de un objeto.	
	1.1- Se le asigna el valor a la variable “coeficiente de rozamiento de un objeto”.
Prioridad	Crítico

TABLA 11 CU EXPANDIDOS, ACTUALIZAR OBJETO.

Caso de Uso	
CU- 4	Actualizar objeto
Actores	Sistema Usuario
Propósito	Mantener actualizadas las variables de cada objeto.
Resumen	Se inicia cuando el Sistema Usuario invoca Actualizar objeto, se invoca al CU incluido Calcular puntos de apoyo, donde estos se calculan. Se determina la altura máxima que el objeto puede trepar, se obtiene el terreno por el que va cada punto, se ubica cada punto sobre el terreno, si no está en un terreno se invoca al caso de uso externo Ubicar puntos de apoyo y luego se le suma a cada punto la altura sobre el terreno. En dependencia de la cantidad de puntos para hacer el seguimiento se actualiza la posición y orientación de un objeto (nodo) sobre el terreno. (Si la cantidad de puntos para el seguimiento es 4 se invoca al CU extendido Ajustar 4 puntos). Se actualiza la normal y el coeficiente de rozamiento del terreno. El caso de uso finaliza cuando quedan actualizados geoméricamente el objeto en el terreno, la normal y el coeficiente de rozamiento del terreno.

Referencias:	R7, R7.1, R8, R9, R10, R10.1, R10.2, R10.3, R10.4, R10.5, R10.6, R11, R12, R13, CU extendido Ajustar 4 puntos y CU incluido Calcular puntos de apoyo.
Poscondiciones	Queda actualizado la orientación y posición de un objeto (nodo).
Curso normal de los eventos	
Acción del actor	Respuesta del sistema
1- Invoca modificar el estado de un objeto.	
	1.1- Invoca al CU incluido Calcular puntos de apoyo .
	1.2- Se determina la altura máxima que el objeto puede trepar.
	1.3- Obtener terreno por el que va cada punto.
	1.4- Si el terreno de un punto es -1 ir a Curso Alterno "Buscar nuevo terreno".
	1.5- Se ubica cada punto sobre el terreno.
	1.6- Invoca al CU extendido Ubicar puntos de apoyo .
	1.7- Se le suma a cada punto la altura sobre el terreno.
	1.8- Con los puntos de control transformados se verifica para cuántos puntos de apoyo se hará el seguimiento de terreno.
	1.9- Si es 1 punto ir a Curso Alterno "Actualizar Objeto (1 punto de apoyo)". - Si son 3 puntos ir a Curso Alterno "Actualizar Objeto (3 puntos de apoyo)". - Si son 4 puntos ir a Curso Alterno "Actualizar Objeto (4 puntos de apoyo)".
	1.10- Se actualiza la normal del terreno.
	1.11- Se actualiza el coeficiente de rozamiento de los terrenos.
	1.12- Se informa con éxito el seguimiento de terrenos.
Curso Alterno "Buscar nuevo terreno"	

Acción del actor	Respuesta del sistema
	1.3- Se busca el nuevo terreno para el punto dado, si retorna diferente de -1, ir a 1.7 en el curso normal de los eventos, sino retornar mensaje de choque.
Curso Alterno “Actualizar Objeto (1 punto de apoyo)”	
Acción del actor	Respuesta del sistema
	1.9- Se actualiza el centro del objeto y ir a 1.10 en el curso normal de los eventos.
Curso Alterno “Actualizar Objeto (3 puntos de apoyo)”	
Acción del actor	Respuesta del sistema
	1.9- Se calcula con la esquina delantera izquierda y la esquina delantera derecha ya transformadas la normal del objeto.
	1.9.1- Se relocaliza el vector izquierdo del objeto.
	1.9.2- Se relocaliza el vector dirección del objeto.
	1.9.3- Se actualiza el centro, el vector dirección, el vector izquierdo y la normal del objeto, ir a 1.10 en el curso normal de los eventos.
Curso Alterno “Actualizar Objeto (4 puntos de apoyo)”	
Acción del actor	Respuesta del sistema
	1.9- Invoca al CU extendido Ajustar 4 puntos .
	1.9.1- Se calcula con la esquina delantera izquierda, la esquina delantera derecha, la esquina trasera izquierda y la esquina trasera derecha ya transformadas y ajustadas la normal del objeto.
	1.9.2- Se relocaliza el vector izquierda del objeto.
	1.9.3- Se relocaliza el vector dirección del objeto.
	1.9.4.- Se calcula con la esquina delantera izquierda y el vector dirección la posición del objeto.
	1.9.5- Se actualiza el centro, el vector dirección, el vector izquierda y la normal del objeto, ir a 1.10 en el curso normal de los eventos.
Prioridad	Crítico

TABLA 12 CU EXPANDIDOS, CALCULAR PUNTOS DE APOYO.

Caso de Uso	
CU- 5	Calcular puntos de apoyo
Actores	CU Actualizar objeto.
Propósito	Mantener actualizadas las variables de cada objeto.
Resumen	Se inicia cuando el CU Actualizar objeto invoca calcular los puntos de apoyo con los que se hará el seguimiento. Después de determinado el estado geométrico del objeto y los vectores de desplazamiento, se calculan estos en dependencia de la cantidad de puntos que sean (1, 3 o 4) puntos.
Referencias:	R6, R6.1, R6.2, R6.3, R6.4, R6.5, R6.6, R6.7, R6.8, R6.9, R6.10.
Curso normal de los eventos	
Acción del actor	Respuesta del sistema
1- Invoca modificar el estado de un objeto.	
	1.1- Se determina el estado geométrico del objeto.
	1.2- Se determinan los vectores de desplazamiento del objeto.
	1.3- Se determina la cantidad de puntos con los que se hará el seguimiento.
	1.4- Verificar para cuántos puntos de apoyo se hará el seguimiento de terrenos.
	1.5- Si es 1 punto ir al Curso Alterno "Calcular 1 Punto de apoyo". - Si son 3 puntos ir al Curso Alterno "Calcular 3 Puntos de apoyo". - Si son 4 puntos ir al Curso Alterno "Calcular 4 Puntos de apoyo".
Curso Alterno "Calcular 1 Punto de apoyo"	

Acción del actor	Respuesta del sistema
	1.5- Asignar como único punto de apoyo el centro del objeto.
Curso Alterno “Calcular 3 Puntos de apoyo”	
Acción del actor	Respuesta del sistema
	1.5.1- Se calcula con el centro del objeto, el vector izquierda y el vector dirección, la esquina delantera izquierda del objeto.
	1.5.2- Se calcula con el centro del objeto, el vector derecha y el vector dirección, la esquina delantera derecha del objeto.
	1.5.3- Se asigna como puntos de apoyo del objeto el centro de este, la esquina delantera derecha y la esquina delantera izquierda.
Curso Alterno “Calcular 4 Puntos de apoyo”	
Acción del actor	Respuesta del sistema
	1.5- Se calcula con el centro del objeto, el vector izquierda y el vector dirección, la esquina delantera izquierda del objeto.
	1.5.1- Se calcula con el centro del objeto, el vector derecha y el vector dirección, la esquina delantera derecha del objeto.
	1.5.2- Se calcula con el centro del objeto, el vector derecha y el vector dirección trasero, la esquina trasera derecha del objeto.
	1.5.3- Se calcula con el centro del objeto, el vector derecha y el vector dirección trasero, la esquina trasera izquierda del objeto.
	1.5.4- Se asigna como puntos de apoyo del objeto la esquina delantera derecha, la esquina delantera izquierda, la esquina trasera izquierda y la esquina trasera derecha.
Prioridad	Crítico

TABLA 13 CU EXPANDIDOS, AJUSTAR 4 PUNTOS.

Caso de Uso	
CU- 6	Ajustar 4 puntos
Actores	CU Actualizar objeto.
Propósito	Ajustar 4 puntos a un plano 3D y devolver los 4 puntos.
Propósito	Se inicia cuando el CU Actualizar objeto define que se hará el seguimiento de terreno con 4 puntos. Se resuelven las ecuaciones y se ajustan los 4 puntos a un plano, se calculan los nuevos puntos ajustados para hacer el seguimiento.
Referencias	R14, R14.1, R14.2, R14.3, R14.4, R14.5, R14.6
Precondiciones	Que existan los 4 puntos a ajustar.
Curso normal de los eventos	
Acción del actor	Respuesta del sistema
1- El CU Actualizar estado geométrico de un objeto especifica que el seguimiento de terrenos se hará con 4 puntos.	
	1.1- Se calcula la ecuación número 1 con los 4 puntos entrados.
	1.2- Se calcula la ecuación número 2 con los 4 puntos entrados.
	1.3- Se calcula la ecuación número 3 con los 4 puntos entrados.
	1.4- Se resuelven las ecuaciones 1 y 2 eliminando las X.
	1.5- Se resuelven las ecuaciones 1 y 3 eliminando las X.
	1.6- Se resuelve el sistema de ecuaciones resultantes de los anteriores cálculos eliminando las Y.
	1.7- Se obtiene el valor de la constante C de la ecuación del plano.
	1.8- Se calcula la constante B de la ecuación del plano

	teniendo C.
	1.9- Se calcula la constante A de la ecuación del plano teniendo B y C.
	1.10- Se obtiene la ecuación del plano equilibrado.
	1.11- Se halla el valor t de la recta paramétrica para el punto 1 en el espacio teniendo el punto y la norma del plano.
	1.12- Se proyecta el punto 1 en el plano equilibrado.
	1.13- Se halla el valor t de la recta paramétrica para el punto 2 en el espacio teniendo el punto y la norma del plano.
	1.14- Se proyecta el punto 2 en el plano equilibrado.
	1.15- Se halla el valor t de la recta paramétrica para el punto 3 en el espacio teniendo el punto y la norma del plano.
	1.16- Se proyecta el punto 3 en el plano equilibrado.
	1.17- Se halla el valor t de la recta paramétrica para el punto 4 en el espacio teniendo el punto y la norma del plano.
	1.18- Se proyecta el punto 4 en el plano equilibrado.
	1.19- Se devuelve un arreglo con los 4 puntos ajustados al plano anteriormente equilibrado.
Prioridad	Crítico

Conclusiones

En el presente capítulo se especificaron las soluciones técnicas que se le dan a este proyecto, partiendo de aquí se diseñará una estructura de clases en correspondencia con las técnicas citadas y para la realización de los algoritmos planteados.

Se establecieron además los requisitos funcionales y no funcionales, propiciando el paso a la siguiente etapa, donde se realizaron las descripciones detalladas de los casos de uso a implementar, apoyados en el modelo de los casos de uso del sistema, permitiendo a su futuro usuario obtener los resultados esperados por el cliente.

Capítulo 3 Diseño del Sistema

Introducción

En el capítulo actual se describen los diagramas de clases del diseño dividido por paquetes para un mejor entendimiento de las clases y sus relaciones. Se presentan además los diagramas de secuencia de la realización de los casos de uso dividido por secciones, brindando una visión detallada de cómo se hará la implementación.

3.1 Diagramas de clases del Diseño

Importantes consideraciones a tener en cuenta para un fácil entendimiento del diagrama de clases del diseño:

La nomenclatura utilizada para los diagramas de clases, se explica en el epígrafe “Estándares de codificación” del siguiente capítulo.

Se agruparon las clases del diseño en dos paquetes, el primero llamado Clases de la Herramienta, donde se agrupan las relaciones entre las clases ya creadas en el proyecto. El segundo paquete, llamado Clases Terreno, donde se relacionan las clases creadas e implementadas pertenecientes al módulo de seguimiento de terrenos.

El paquete Clases Terreno utiliza las clases del paquete Clases de la Herramienta.



FIG. 17 DIAGRAMA DE PAQUETES DE CLASES DEL DISEÑO.

3.1.1 Diagrama de clase del paquete Clases de la Herramienta

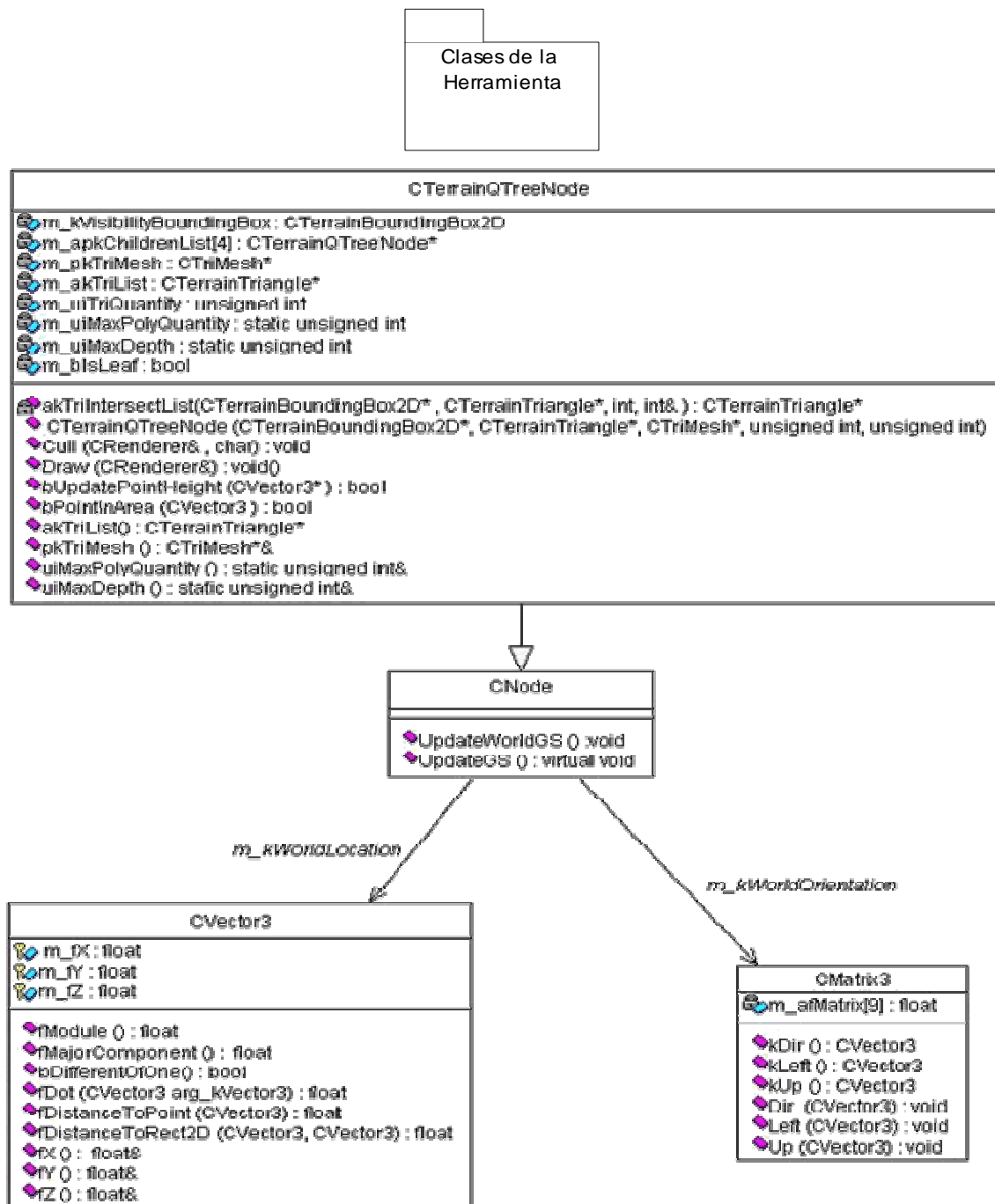


FIG. 18 DIAGRAMA DE CLASE DEL PAQUETE CLASES DE LA HERRAMIENTA.

3.1.2 Diagrama de clases de paquete Clases Terreno.

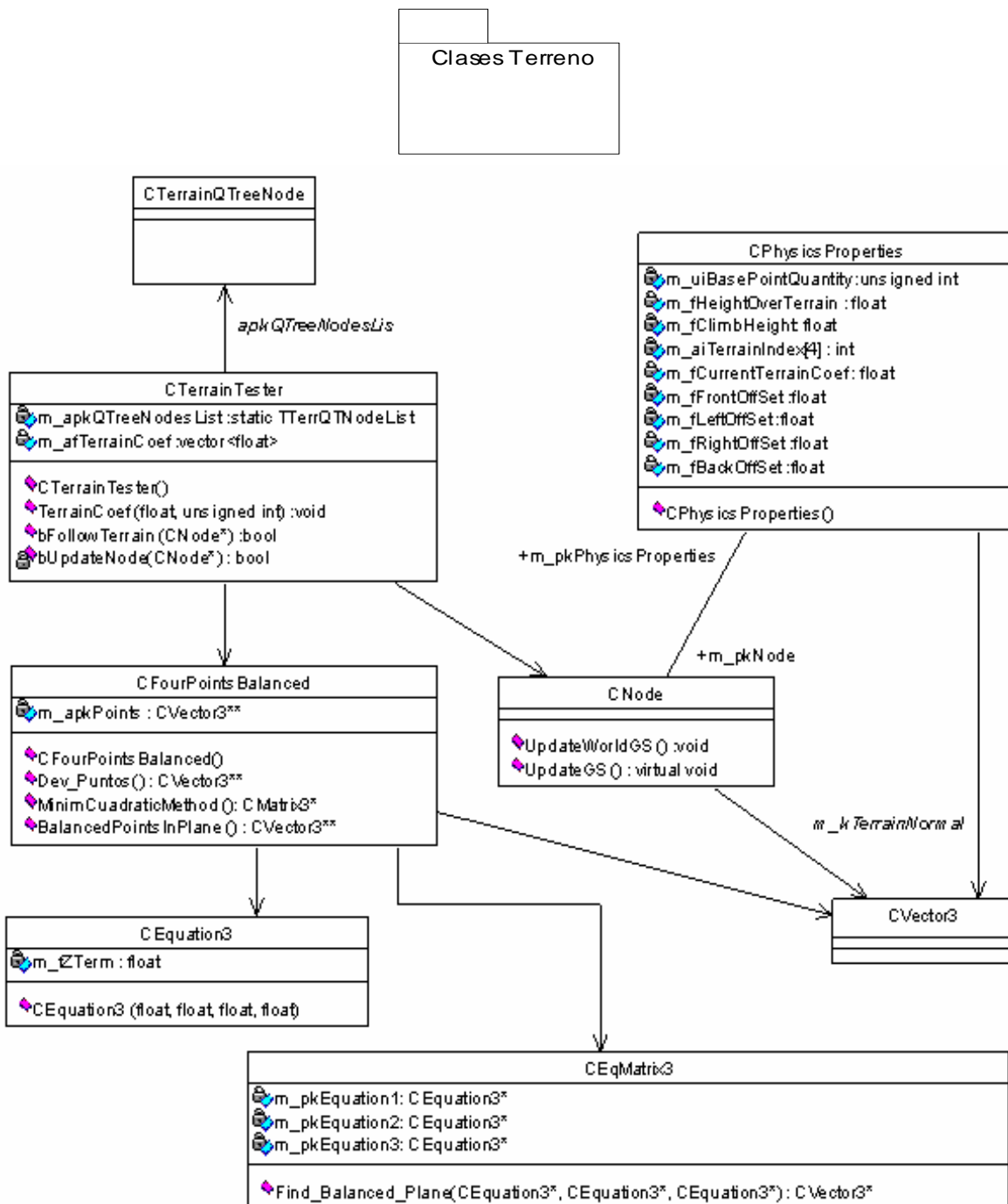


FIG. 19 DIAGRAMA DE CLASES DE PAQUETE CLASES TERRENO.

3.1.3 Descripción de las clases del diseño

A continuación serán descritas las clases más importantes relacionadas a la implementación del módulo, no serán detalladas las clases ya implementadas en la herramienta.

TABLA 14 DESCRIPCIÓN DE LA CLASE “CTERRAINTESTER”.

Nombre: CTerrainTester	
Tipo de clase: Controladora	
Atributo	Tipo
m_apkQTreeNodeList	static TTerrQTreeNodeList
M_afTerrainCoef	vector <float>
Para cada responsabilidad:	
Nombre:	CTerrainTester ()
Descripción:	Constructor de la clase
Nombre:	bUpdateNode(CNode* arg_pkNode)
Descripción:	Ubica los puntos sobre el terreno.
Nombre:	bFollowTerrain (CNode*arg_pkNode)
Descripción:	Realiza el seguimiento de terreno de un objeto (nodo).
Nombre:	TerrainCoef (float arg_fTerrainCoef, unsigned int arg_uiTerrainIndex)
Descripción:	Modifica coeficiente de los terrenos.

TABLA 15 DESCRIPCIÓN DE LA CLASE “CPHYSICSPROPERTIES”.

Nombre: CPhysicsProperties	
Tipo de clase: Entidad	
Atributo	Tipo
m_uiBasePointQuantity	insigned int
m_fHeigtOverTerrain	float
m_ffClimbHeight	float
m_aiTerrainIndex[4]	int
m_fCurrentTerrainCoef	float
m_fFrontOffSet	float
m_fLeftOffSet	float
m_fRightOffSet	float
m_fBackOffSet	float
Para cada responsabilidad:	
Nombre:	CPhysicsProperties()
Descripción:	Constructor de la clase.

TABLA 16 DESCRIPCIÓN DE LA CLASE “CFourPointsFitter”.

Nombre: CFourPointsFitter	
Tipo de clase: Entidad	
Atributo	Tipo
m_apkPoints	CVector3**
Para cada responsabilidad:	
Nombre:	CFourPointsFitter ()
Descripción:	Constructor de la clase.
Nombre:	MinimCuadraticMethod (CVector3 *arg_pkPoint1, CVector3* arg_pkPoint 2, CVector3 * arg_pkPoint 3, CVector3 * arg_pkPoint 4)
Descripción:	Construye una matriz de 3 x 3.
Nombre:	BalancedPointsInPlane (CVector3 *punto1, CVector3 *punto2, CVector3 *punto3, CVector3 *punto4, CVector3 *plane)
Descripción:	Proyecta cada punto en el plano.
Nombre:	FitPoints (CVector3 *arg_pkPoint1, CVector3 *arg_pkPoint2, CVector3 *arg_pkPoint3, CVector3 *arg_pkPoint4)
Descripción:	Ajusta cada punto modificando sus valores iniciales.

TABLA 17 DESCRIPCIÓN DE LA CLASE “CEqMatrix3”.

Nombre: CEqMatrix3	
Tipo de clase: Entidad	
Atributo	Tipo
m_pkEquation1	CEquation3*
m_pkEquation2	CEquation3*
m_pkEquation3	CEquation3*
Para cada responsabilidad:	
Nombre:	CEqMatrix3 (CEquation3* arg_pkEquation1, CEquation3* arg_pkEquation2, CEquation3* arg_pkEquation3)
Descripción:	Constructor de la clase.
Nombre:	MatrixSolve (CEquation3* arg_pkEquation1, CEquation3* arg_pkEquation2, CEquation3* arg_pkEquation3)
Descripción:	Resuelve la matriz entrada por parámetros. (halla el plano equilibrado)

3.2 Diagramas de Secuencia

El Lenguaje Unificado de Modelado (UML), provee un medio gráfico para representar la interacción entre los objetos a lo largo del tiempo mediante los diagramas de secuencia. Estos muestran el flujo de mensajes de un objeto a otro y como tales, representan los métodos y los eventos soportados por un objeto o clase. Seguidamente se muestran los diagramas de secuencias correspondientes a los casos de uso del módulo a implementar.

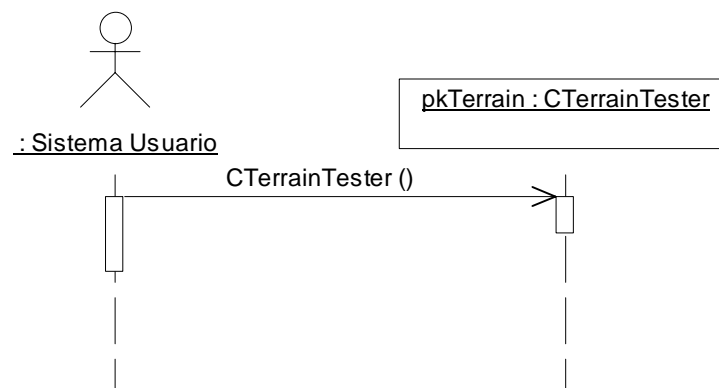


FIG. 20 DIAGRAMA DE SECUENCIA “MANIPULAR CONTROLADOR ST”, ESCENARIO “CREAR CONTROLADOR”.

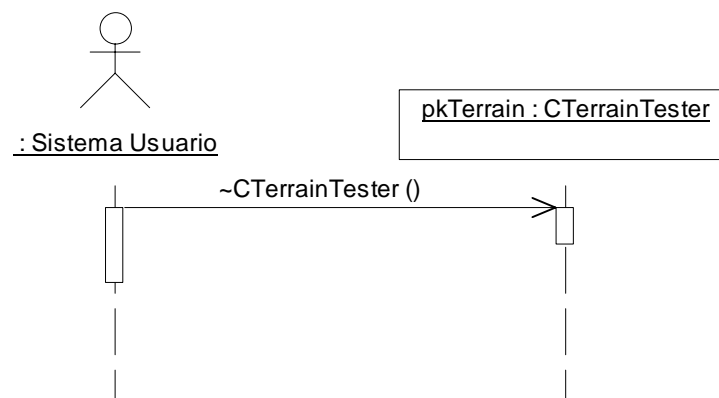


FIG. 21 DIAGRAMA DE SECUENCIA “MANIPULAR CONTROLADOR ST”, ESCENARIO “DESTRUIR CONTROLADOR”.

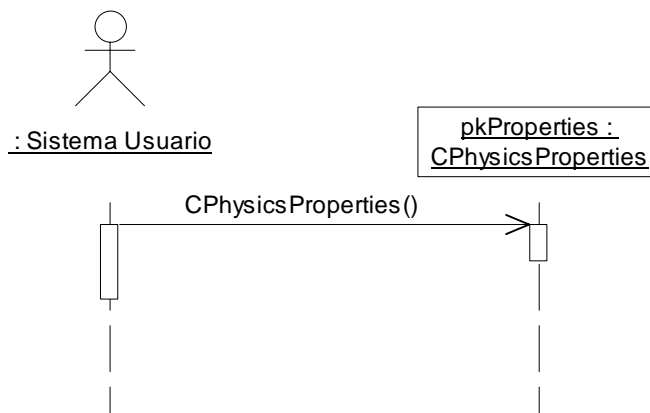


FIG. 22 DIAGRAMA DE SECUENCIA “MANIPULAR PROPIEDADES FÍSICAS DE LOS OBJETOS”, ESCENARIO “CREAR PROPIEDADES”.

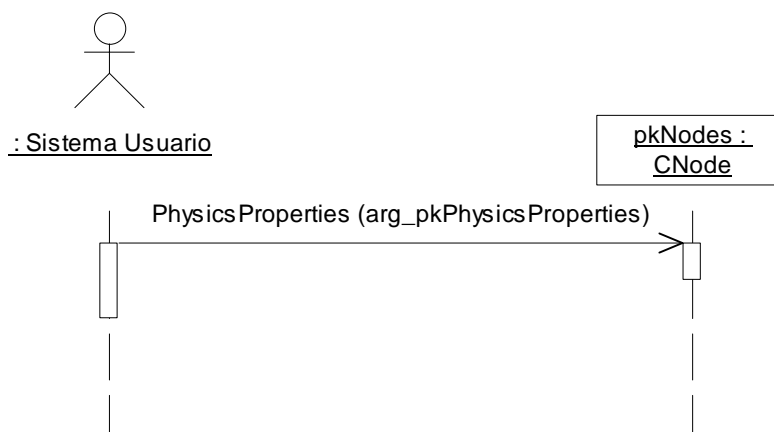


FIG. 23 DIAGRAMA DE SECUENCIA “MANIPULAR PROPIEDADES FÍSICAS DE LOS OBJETOS”, ESCENARIO “ASOCIAR CONTROLADOR”.

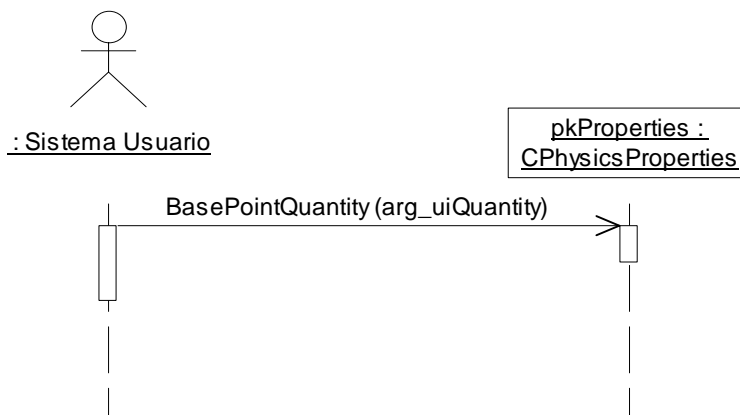


FIG. 24 DIAGRAMA DE SECUENCIA “CONFIGURAR PROPIEDADES DE LOS OBJETOS”, ESCENARIO “CONFIGURAR PUNTOS DE LA BASE”.

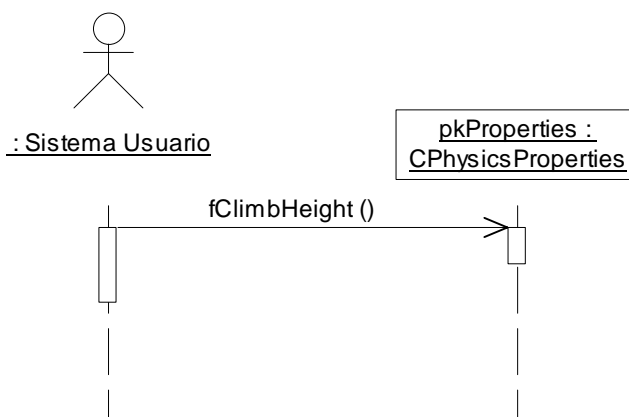


FIG. 25 DIAGRAMA DE SECUENCIA “CONFIGURAR PROPIEDADES DE LOS OBJETOS”, ESCENARIO “CONFIGURAR ESCALADO”.

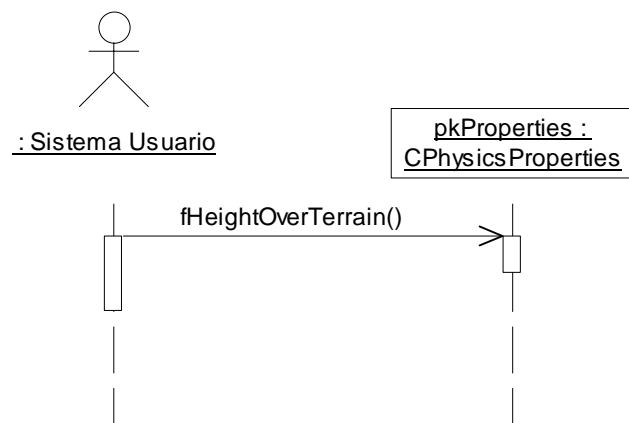


FIG. 26 DIAGRAMA DE SECUENCIA “CONFIGURAR PROPIEDADES DE LOS OBJETOS”, ESCENARIO “CONFIGURAR ALTURA SOBRE EL TERRENO”.

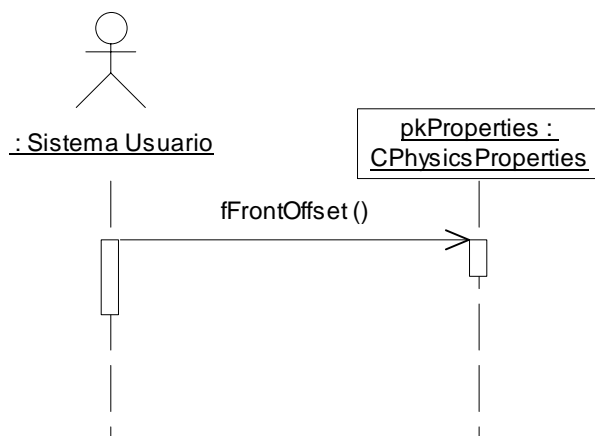


FIG. 27 DIAGRAMA DE SECUENCIA “CONFIGURAR PROPIEDADES DE LOS OBJETOS”, ESCENARIO “CONFIGURAR DESPLAZAMIENTO DE LOS OBJETOS (DELANTERO)”.

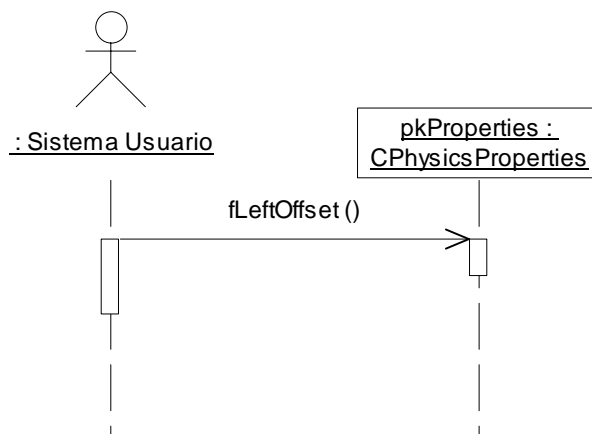


FIG. 28 DIAGRAMA DE SECUENCIA “CONFIGURAR PROPIEDADES DE LOS OBJETOS”, ESCENARIO “CONFIGURAR DESPLAZAMIENTO DE LOS OBJETOS (IZQUIERDO)”.

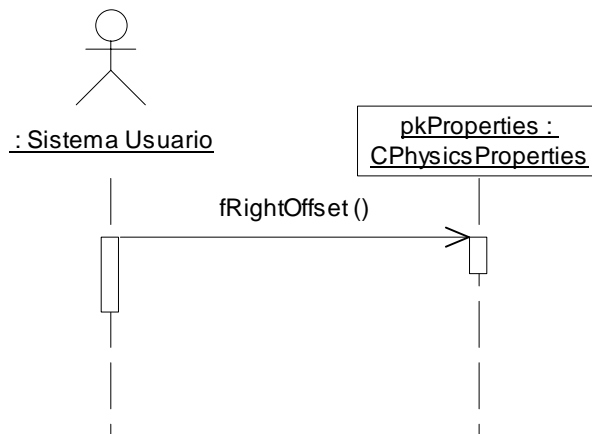


FIG. 29 DIAGRAMA DE SECUENCIA “CONFIGURAR PROPIEDADES DE LOS OBJETOS”, ESCENARIO “CONFIGURAR DESPLAZAMIENTO DE LOS OBJETOS (DERECHO)”.

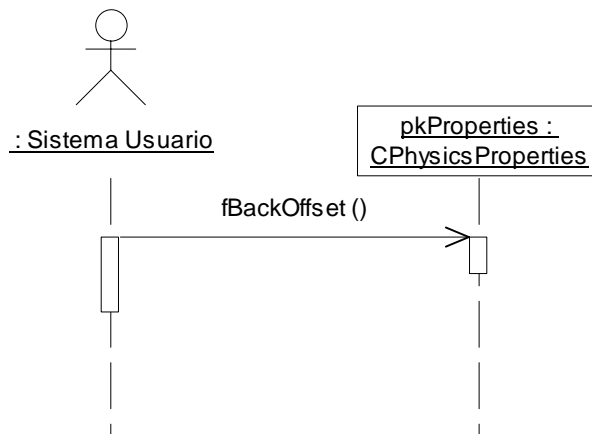


FIG. 30 DIAGRAMA DE SECUENCIA “CONFIGURAR PROPIEDADES DE LOS OBJETOS”, ESCENARIO “CONFIGURAR DESPLAZAMIENTO DE LOS OBJETOS (TRASERO)”.

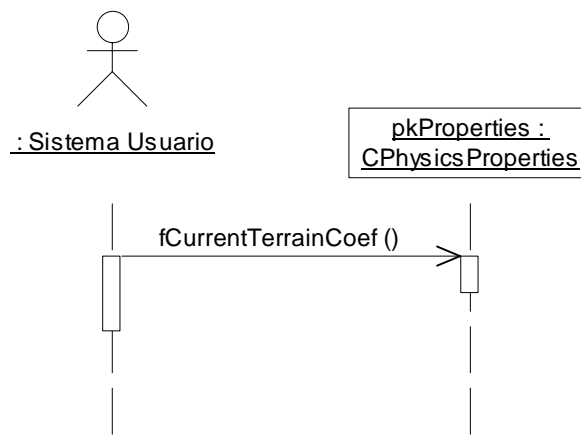


FIG. 31 DIAGRAMA DE SECUENCIA “CONFIGURAR PROPIEDADES DE LOS OBJETOS”, ESCENARIO “CONFIGURAR COEFICIENTE DE ROZAMIENTO DE LOS OBJETOS”.

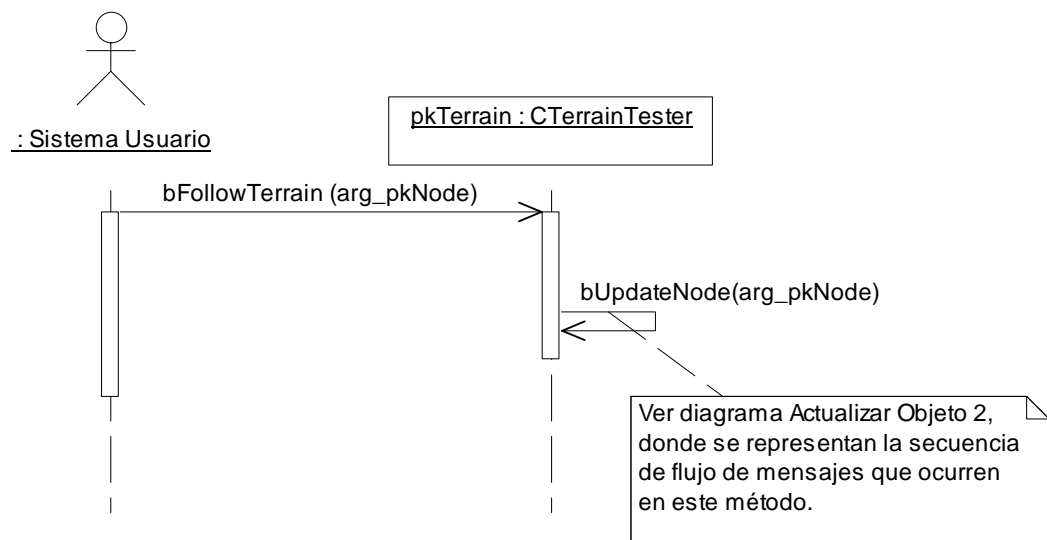


FIG. 32 DIAGRAMA DE SECUENCIA “ACTUALIZAR OBJETO 1”.

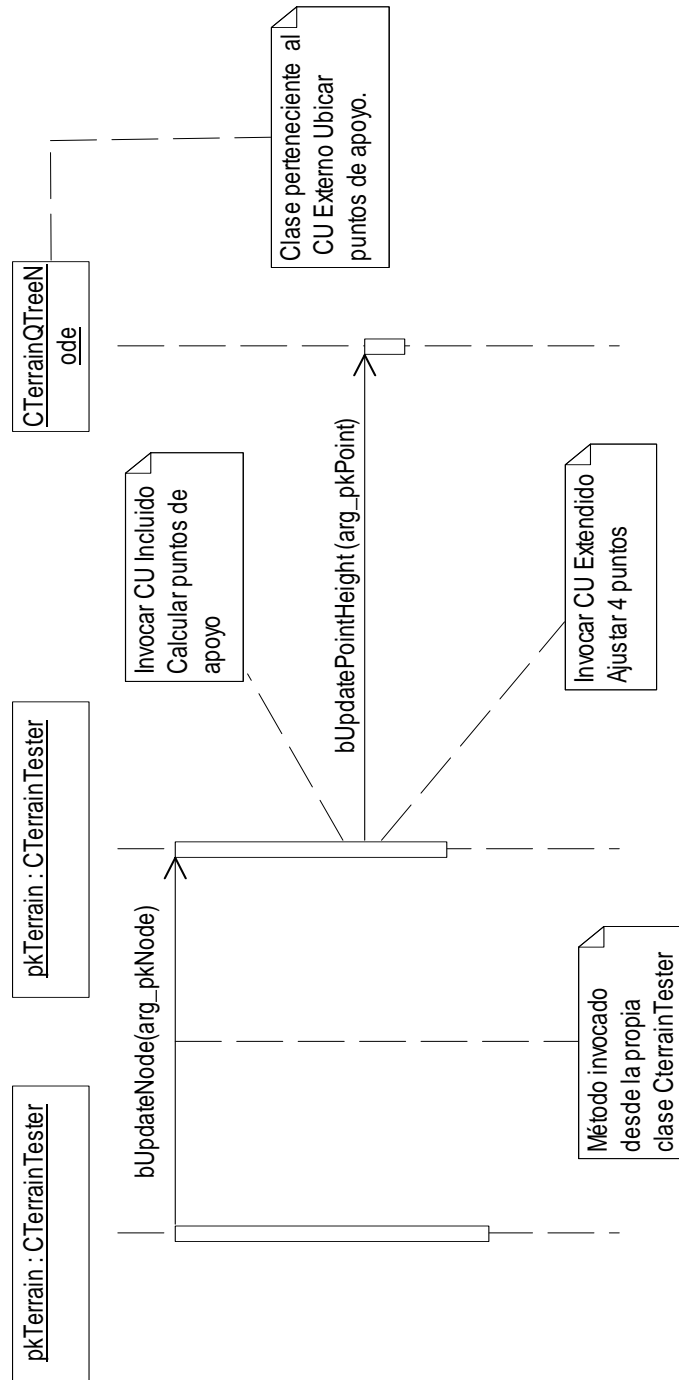


FIG. 33 DIAGRAMA DE SECUENCIA "ACTUALIZAR OBJETO 2".

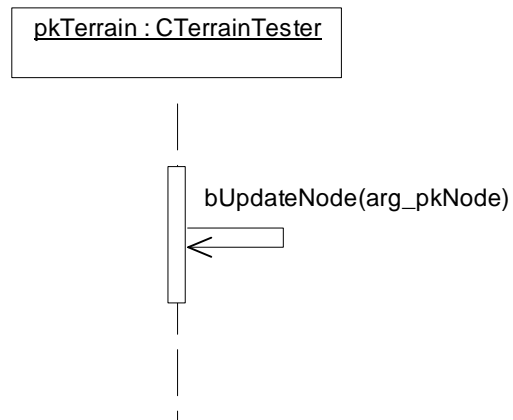


FIG. 34 DIAGRAMA DE SECUENCIA CU INCLUIDO “CALCULAR PUNTOS DE APOYO”.

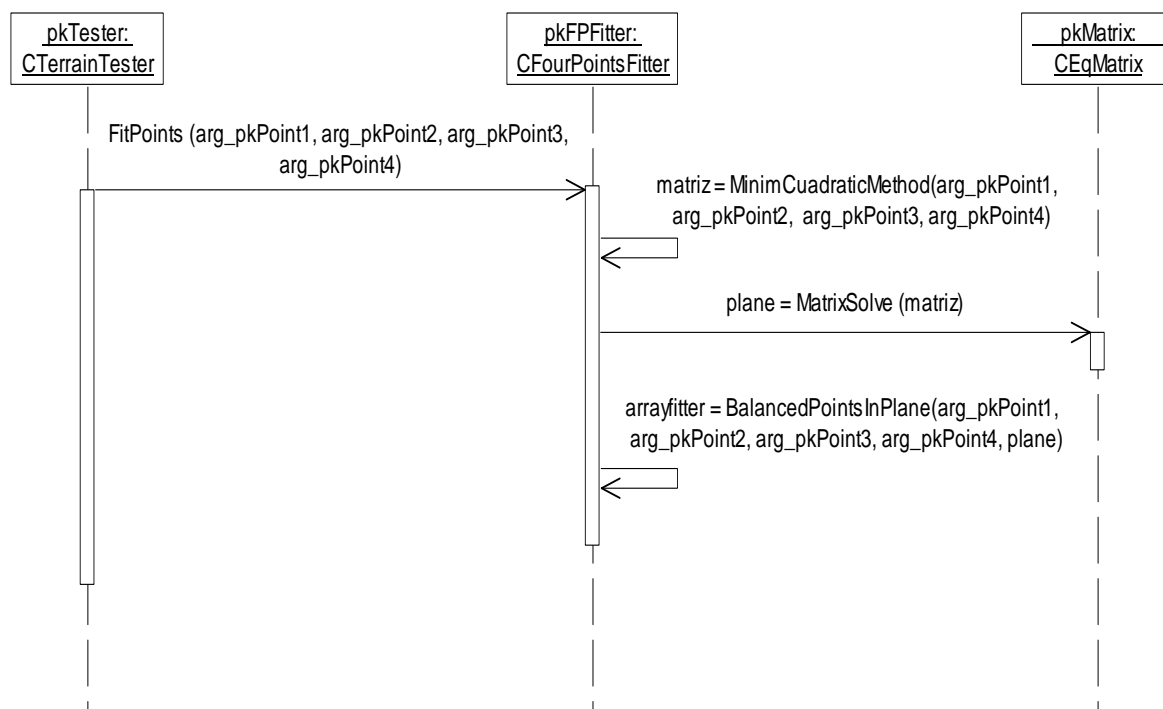


FIG. 35 DIAGRAMA DE SECUENCIA CU EXTENDIDO “CALCULAR 4 PUNTOS”.

Conclusiones

Al concluir este capítulo se está en condiciones para pasar a la etapa de implementación del proyecto. Se llevó a cabo un diseño completo del sistema, traducido en clases y sus relaciones, así como los diagramas de secuencias con sus mensajes, desarrollando las funcionalidades de los CU.

Capítulo 4 Implementación del Sistema

Introducción

En esta fase del proyecto se está en condiciones de traducir el diseño de clases elaborado en la etapa anterior, en componentes físicos, mediante el diagrama de componentes, los cuales se convierten en ficheros .cpp correspondientes a la implementación en C++.

Estándares de codificación

Este patrón está establecido para usuarios y programadores de la herramienta. De manera que sea más fácil el entendimiento en relación a la nomenclatura utilizada.

Se programará en inglés, debido a que las palabras son simples, no se acentúan y es un idioma muy difundido en el mundo informático. Se respetarán los estándares de codificación para C++ (indentado, uso de espacios y líneas en blanco, etc.).

El conocimiento de los estándares seguidos para el desarrollo del módulo permitirá un mayor entendimiento del código y es una exigencia de los autores de la misma que cualquier módulo que se añada debe estar codificado siguiendo estos estándares.

Nombre de los ficheros:

Se nombrarán los ficheros .h y .cpp de la siguiente manera:

STKNameOfUnits.cpp

Se usará **STK** para identificar el nombre de la herramienta.

Constantes:

Las constantes se nombrarán con mayúsculas, utilizándose el “_” para separar las palabras: MY_CONST_ZERO = 0;

Tipos de datos:

Los tipos se nombrarán siguiendo el siguiente patrón:

Enumerados: enum **EMyEnum** {**ME**_VALUE, **ME**_OTHER_VALUE};

Indicando con “**E**” que es de tipo enumerado. Nótese que las primeras letras de las constantes de enumerados son las iniciales del nombre del enumerado. Véase otro ejemplo:

```
enum ENodeType {NT_GEOMETRYNODE,...};
```

Estructuras: struct **SMyStruct** {...};

Indicando con “**S**” que es una estructura. Las variables miembros de la estructura se nombrarán igual que en las clases, leer más adelante.

Clases: class **CClassName**;

Indicando con “**C**” que es una clase

Declaración de variables:

Los nombres de las variables comenzarán con un identificador del tipo de dato al que correspondan, como se muestra a continuación. En el caso de que sean variables miembros de una clase, se le antepone el identificador “m_” (en minúscula), si son globales se les antepone la letra “g”, y en caso de ser argumentos de algún método, se les antepone el prefijo “arg_”.

Tipos simples:

```
bool bVarName;
```

```
int iName;
```

```
unsigned int uiName;
```

```
float fName;
```

```
char cName;
```

```
char* acName; // arreglo de caracteres
```

```
char* pcName; // puntero a un char
```

```
char** aacName; // bidimensional
```

```
char** apcName; // arreglo de punteros
```

```
bool m_bMemberVarName; //variable miembro
```

```
char gcGlobalVarName; //variable global, no se le antepone “m_”
```

```
short sName;
```

Instancias de tipos creados:

EMyEnumerated **eName**;

SMyStructure **kName**;

CClassName **kObjectName**;

CClassName* **pkName**; //puntero a objeto

CClassName* **akName**; //arreglo de objetos

CClassName* **akName**; // variable miembro de clase

IMyInterface* **plName**; //puntero interfaces

Métodos

En el caso de los métodos, se les antepondrá el identificador del tipo de dato de devolución, y en caso de no tenerlo (void), no se les antepondrá nada.

En el caso de los argumentos se les antepone el prefijo “arg_”

Constructor y destructor:

```
CClassName (bool arg_bVarName, float& arg_fVarName);
```

```
~CClassName ();
```

Funciones:

```
bool bFunction1 (...);
```

```
int* piFunction2 (...);
```

```
CClassName* pkFunction3 (...);
```

Procedimientos:

```
void Procedure4 (...);
```

Métodos de acceso a miembros

Los métodos de acceso a los miembros de las clases no se nombrarán “Gets” y “Sets”, sino como los demás métodos, pero **con el nombre** de la variable a la que se accede y sin “m_”:

```
int iMyVar; //variable
```

Obtención del valor:

```
int iMyVar();  
  
{  
  
    return iMyVar;  
  
}
```

Establecimiento del valor:

```
void MyVar(char* arg_iMyVar)  
  
{  
  
    iMyVar = arg_iMyVar;  
  
}
```

Obtención y establecimiento del valor:

```
int& iMyVar();  
  
{  
  
    return iMyVar;  
  
}
```

4.1 Diagrama de despliegue

El diagrama de despliegue lo compone una sola PC, siendo muy simple su representación, motivo por el cual se decidió omitir el mismo.

4.2 Diagrama de componentes

A continuación se muestra la relación entre el paquete de componentes STKTerrainFollowing y el paquete "STKSceneToolKit", es decir, la herramienta, representada de esta forma, debido a la estrecha relación que existe entre las clases de la herramienta y el módulo de seguimiento de terrenos.

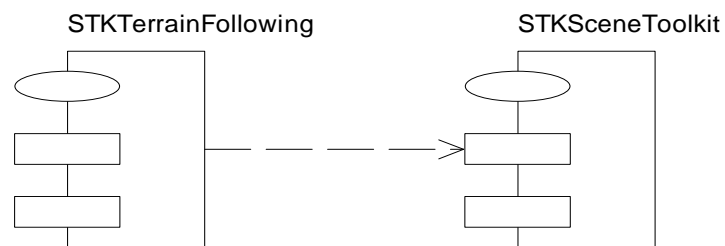


FIG. 36 PAQUETES DE COMPONENTES.

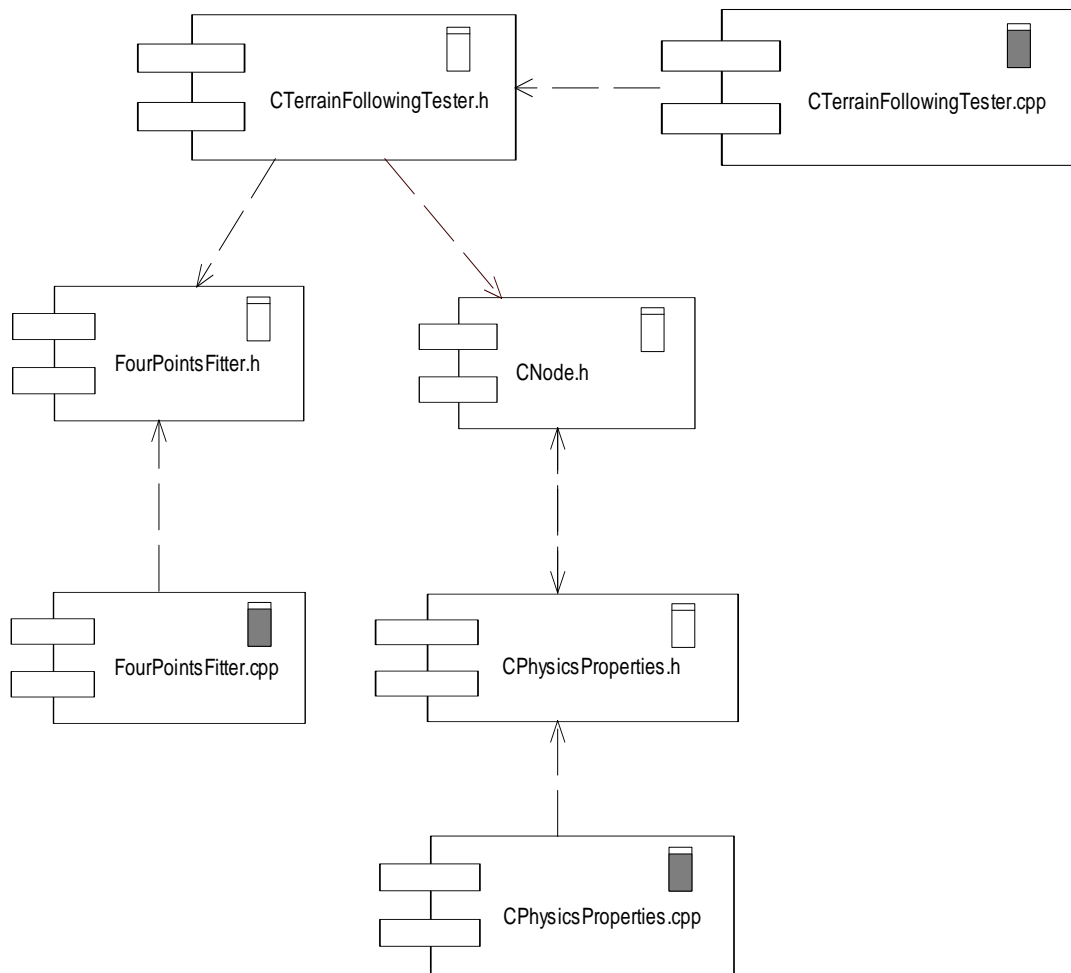


FIG. 37 DIAGRAMA DE COMPONENTES "STKFollowingTerrain".

Conclusiones

Con la culminación de este capítulo se encuentra todo preparado para pasar a la etapa de programación de los casos de uso desarrollados. Quedó establecido además, el estándar a seguir para la programación de los mismos y la relación entre los componentes a programar.

Conclusiones Generales

La revisión realizada sobre las técnicas, tecnologías y tendencias relacionadas con la implementación de seguimientos de terrenos virtuales, rindió los resultados esperados, ya que nos permitió alcanzar los objetivos propuestos.

El modelo matemático empleado (Mínimos Cuadrados) durante el desarrollo del trabajo facilitó el ajuste eficiente y práctico de los puntos de apoyo en el plano 3D. Lo que avala a este método de cálculo como uno de los más utilizados internacionalmente para el ajuste de superficies. Por otra parte el módulo propuesto se acopló exitosamente a la herramienta básica creada en el proyecto “Herramientas de Desarrollo para Sistemas de Realidad Virtual”; revalorizando así la utilidad de esta herramienta.

El módulo funcional obtenido satisface los requisitos del cliente y permite al programador establecer el nivel de realidad deseado en la escena al hacer seguimiento de terrenos los objetos. Puede ser utilizado tanto en juegos como en simuladores de conducción. Este módulo integrado a la biblioteca básica “Herramientas de Desarrollo para Sistemas de Realidad Virtual”, propicia un mayor realismo en los sistemas de Realidad Virtual.

Recomendaciones

- Utilizar el módulo desarrollado para el seguimiento de terrenos en juegos y/o simuladores de conducción.
- Utilizar la estructura de datos Octree para el almacenamiento de los terrenos.
- Extender el módulo de seguimiento de terrenos a otras cantidades de puntos, utilizando el método matemático de mínimos cuadrados implementado satisfactoriamente en este trabajo.

Referencias bibliográficas

Libros

[1]. Luna, Frank D. Introduction to 3D Game Programming with DirectX. WordWare Publishing, Inc. USA. 2003. Capítulo 13: *Basic Terrain Rendering*.

[2]. Alvarez Manuel, Lau Rogelio, Guerra Alfredo. Matemática numérica 2da Edición. Capítulo 4. Aproximación de funciones.

Libros digitales

[3]. Manuel Felicísimo Angel. [Consultado en noviembre 2006]. "Modelado Geométrico".

[3.1]. El modelo digital de elevaciones.

Disponible en: http://www.etsimo.uniovi.es/~feli/CursoMDT/Tema_2.pdf

[3.2]. La utilización de los MDT en los estudios del medio físico.

Disponible en: http://www.etsimo.uniovi.es/~feli/pdf/ITGE_150a.pdf

[4]. García Osorio César Ignacio y Martín Antón Luis Miguel. [Consultado en noviembre 2006]. "Memoria del Proyecto Fin de Carrera Robot Recuperador Web SpiderBot 1.0 Ignacio Cruzado Nuño". Disponible en:

<http://pisuerga.inf.ubu.es/lsi/Docencia/TFC/ITIG/icruzadn/Memoria/26.htm>

[5]. R. Escarpín Emilio. [Consultado en noviembre 2006]. "La realidad virtual, una tecnología educativa a nuestro alcance". Disponible en:

<http://www.sav.us.es/pixelbit/articulos/n15/n15art/art151.htm>

[6]. Guerrero Tala Fernando. [Consultado en febrero 2007]. "Lenguaje C++. Guía para Programadores". Disponible en:

<http://www.mailxmail.com/curso/informatica/cplusplus2>

[7]. Remolar Inmaculada. [Consultado en diciembre 2006]. "Aplicaciones de la Informática Gráfica". Disponible en:

<http://www3.uji.es/~jromero/grafica/Documentos/9-AplicacionesIG35.pdf>

- [8]. Remolar Inmaculada. [Consultado en noviembre 2006]. “Modelado geométrico”.
Disponible en: <http://www3.uji.es/~jromero/grafica>
- [8.1]. Modelado geométrico I.
<http://www3.uji.es/~jromero/grafica/Documentos/2-Modelado1IG35.pdf>
- [8.2]. Modelado geométrico II.
<http://www3.uji.es/~jromero/grafica/Documentos/8-Modelado2IG35.pdf>
- [9]. Friedich Walter Jhon y Diesel Von Stuart. [Consultado en diciembre 2006].
“Virtual Reality, VPL Research. Disponible en: <http://www.icc.uji.es/asignatura>
- [10]. W. Barrus John y C. Walters Richard. [Consultado en enero 2007]. “QOTA: a fast, multi-purpose algorithm for terrain following in virtual environments”. Disponible en:
<http://portal.acm.org/results.cfm>
- [11]. Luis Bosque Jose. [Consultado en noviembre 2006]. “Sistema de representación y Modelado de Sólidos. Disponible en:
<http://gatarvisa.escet.urjc.es/seminarios/modeladosolidos.ppt#301,60,9>.
- [12]. Torres J.C. [Consultado en noviembre 2006]. “Modelado de sólidos”. Disponible en: <http://lsi.ugr.es/~cad/teoria/Tema5/RESUMENTEMA5.PDF>
- [13]. Marcos Fernández. [Consultado en diciembre 2006]. “Ampliación de Informática Gráfica”. Disponible en:
http://informatica.uv.es/iiguia/AIG/web_teoría/Tema1_cn.pdf
- [14]. Fernández de laTorre Reynier y MSc. Geler Roffe Tatiana. [Consultado en diciembre 2006]. “Modelos Digitales de Elevación”. Disponible en:
http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=509
- [15]. Saunders Rob. [Consultado en noviembre 2006]. “Advanced Games Design, Theory and Practice”. Disponible en:
<http://www soi.city.ac.uk/~rob/module02/lectures/lecture04/Lecture04-8up.pdf>

Sitios Web

- [16]. Análisis de Lenguajes de Programación. Disponible en:
<http://www.vjuegos.org/modules.php?name=Content&pa=showpage&pid=4>
- [17]. Level of Detail 3D Graphics. Disponible en:
<http://books.google.es/books?id=CB1N1aa0MloC&dq=history+Continuous+Level+of+Detail+&psp=1>
- [18]. Mallas y Quadtree. Disponible en:
<http://gatarvisa.escet.urjc.es/seminarios/modeladosolidos>
- [19]. Métodos matemáticos para el ajuste de superficies. Disponible en:
<http://iit.jalisco.gob.mx/glosario/m.html>
- [20]. Realidad Virtual. Disponible en:
http://www.galeon.com/ultimopunto/cat_tecno/tecno_vr.htm
- [21]. Simulador de pérdidas de programación basado en modelos digitales de elevaciones. Disponible en:
<http://www.galeon.com/proplab/>
- [22]. Sistemas de realidad virtual Disponible en:
<http://vrm1.sgi.com/moving-worlds/index.html>
- [23]. Técnicas para simulación en tiempo real. Disponible en:
http://informatica.uv.es/iiquia/AIG/web_teoría/tema6.pdf
- [24]. The ACM Digital Library. Disponible en:
<http://portal.acm.org/results.cfm?query=CCS%3A%22I%2E3%2E7%22&coll=portal&dl=ACM&CFID=11111111&CFTOKEN=2222222>
- [25]. Tiempo real. Disponible en:
http://es.wikipedia.org/wiki/Tiempo_real.

Bibliografía consultada

- Luna, Frank D. Introduction to 3D Game Programming with DirectX, WordWare Publishing, Inc. USA. 2003.
- Manuel Alvarez, Rogelio Lau, Alfredo Guerra. Matemática numérica 2da Edición.
- Manuel Felicísimo, Angel. [Consultado en noviembre 2006]. "Modelado Geométrico". Disponible en: <http://www.etsimo.uniovi.es>

Apéndices

Glosario de abreviaturas

VPL Research: Visual Programming Language

CID2: Centro de Investigación y Desarrollo #2

SIMPRO: Simuladores Profesionales

SRV: Sistemas Realidad Virtual

3D: Tres dimensiones

2D: Dos dimensiones

LOD: Nivel de detalle

MDE: Modelo digitales de elevación

MDT: Modelos digitales del terreno

TIN: Triangulated irregular network

RIT: Red Irregular de Triángulos

URN: Uniform regular network

GRID: Red regular de malla cuadrada

CSG: Geometría sólida Constructiva

FEA: Análisis de elementos finitos

BSPtree: Binary Space Partition tree

CLOD: Continuous Level of Detail

CPU: Unidad Central de Procesamiento

HW: Hardware.

ROAM: Real-time Optimally Adapting Meshes

QOTA: Algoritmo de Rápida Orientación del Terreno

Glosario de términos

-A-

Adosado: Poner una cosa junto a otra que le sirve de respaldo o apoyo.

Algoritmos: Conjunto finito de pasos o instrucciones que se deben seguir para realizar una determinada tarea.

Altimetría: Parte de la topografía que enseña a medir las alturas: la altimetría estudia los métodos y técnicas para la representación del relieve del terreno.

Altitud: Altura de un punto de la Tierra con relación al nivel del mar.

Ambigua: Puede entenderse de varios modos o admitir distintas interpretaciones.

Animación: Simulación de un movimiento creado por la muestra de una serie de imágenes o fotogramas.

Arista: Línea que resulta de la intersección de dos superficies, considerada por la parte exterior del ángulo que forman.

-C-

Conjunto coplanar de puntos: Serie de puntos que están en el mismo plano geométrico.

Convexa: Se dice de la línea o superficie curva cuya parte más prominente está del lado del que mira.

-E-

Equiláteros: Se aplica a las figuras geométricas que tienen todos sus lados iguales.

-F-

Fotograma: Ver cada una de las imágenes que componen una animación.

-I-

Informática gráfica: Es la creación, manipulación, análisis e interacción de las representaciones pictóricas de objetos y datos por medio de un ordenador.

-M-

Malla: Forma de representar un modelo a partir de polígonos. Colección de vértices, aristas y polígonos conectados de forma que cada arista es compartida como máximo por dos polígonos.

Malla de un terreno: Es una cuadrícula o red de triángulos, pero con una altura por cada vértice que representará cómo se modela una transición suave de montañas a valles, simulando terrenos naturales.

Mapas de alturas: Son arreglos donde en cada elemento de este se especifica la altura de un vértice en particular de la cuadrícula del terreno.

Mapa de tonalidades de grises: Representan las diferentes tonalidades que toman los puntos en el terreno.

Métodos inmersivos de realidad virtual: Con frecuencia se ligan a un ambiente tridimensional creado por computadora el cual se manipula a través de cascos, guantes u otros dispositivos que capturan la posición y rotación de diferentes partes del cuerpo humano.

Modelos: Es un esquema teórico, habitualmente matemático, que representa el comportamiento y la evolución de un sistema definido mediante una serie de parámetros.

Modelos alámbricos: Son aquellos cuerpos que se representan a partir de las aristas que los componen.

Modelos de superficies de contorno (planas): Son un conjunto de líneas rectas (arcos) que no se cruzan y que unen un conjunto coplanar de puntos (vértices) definiendo un área simple (habitualmente convexa y sin agujeros).

Modelos de superficies de contorno (curvas): Se definen por ecuaciones cuadráticas (descritos mediante ecuaciones de segundo grado) y **splines** (representación por interpolación).

Modelos sólidos: Tienen en cuenta además de las características geométricas del cuerpo, los comportamientos físico mecánico de los mismos.

-P-

Poligonal: Del polígono o relativo a él.

Polígono: Figura geométrica plana limitada por segmentos rectos consecutivos no alineados, llamados lados.

-R-

Realidad virtual no inmersiva: Utiliza medios como el que actualmente ofrece Internet en el cual se puede interactuar en tiempo real con diferentes personas en espacios y ambientes que en realidad no existen sin la necesidad de dispositivos adicionales a la computadora.

Robótica: Parte de la ingeniería que se ocupa de la aplicación de la informática al diseño y uso de máquinas que sustituyan a las personas en la realización de diferentes tareas o funciones: la robótica es una de las disciplinas clásicas que forman parte de la inteligencia artificial.

-S-

Sistema de Realidad virtual: Sistema informático interactivo que ofrece una percepción sensorial al usuario de un mundo tridimensional sintético que suplanta al real.

Simulación: Una simulación es la ejecución (habitualmente computerizada) de un modelo que reproduce el comportamiento de un sistema sometido a unas condiciones predeterminadas, posiblemente cambiantes en el tiempo.

-T-

Tiempo real: Se procesa cada señal digital ofrecida en un ambiente virtual antes de mostrarse la siguiente señal. Instantánea relación entre el soporte físico (hardware) y la programática (software) de la computadora.

Textura: Imagen que sirve de “piel” a los modelos en un mundo virtual.

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

Topológica: De topología.

-V-

Vectorial: Relativo a los vectores.

Vector: Cantidad que expresa magnitud y dirección.

Vértices: Punto en el que se unen los lados de un ángulo o las caras de un poliedro. Punto de una curva en el que la curvatura es máxima o mínima. Extremo o punto más alto de una cosa.

Índice de figuras y tablas

Índice de figuras

FIG. 1 SISTEMA DE REFERENCIA ESPACIAL.....	24
FIG. 2 RED IRREGULAR DE TRIÁNGULOS	26
FIG. 3 EJEMPLO DE TRIANGULACIÓN.	27
FIG. 4 DESCOMPOSICIÓN DE UNA FIGURA BIDIMENSIONAL EN UN BSP-TREE.....	31
FIG. 5 EJEMPLO DE QUADTREE.....	32
FIG. 6 PARTICIÓN QUADTREE.....	33
FIG. 7 DIVISIÓN EN SECCIONES.	34
FIG. 8 REPRESENTACIÓN DE CUADRANTES.	34
FIG. 9 CUADRANTES PARCIALMENTE LLENOS.	35
FIG. 10 DESCOMPOSICIÓN DE SUPERFICIES EN QUADTREES Y SÓLIDOS EN OCTREES.	36
FIG. 11 EJEMPLO DE PARTICIÓN DE UN ÁRBOL OCTREE.....	37
FIG. 12 Y 13 REPRESENTACIÓN DE PIRÁMIDE CON UN OCTREE.....	37
FIG. 14 PROPIEDADES DE LOS OCTREES.....	38
FIG. 15 MODELO DEL DOMINIO.....	51
FIG. 16 DIAGRAMA DE CASOS DE USOS DEL SISTEMA.....	64
FIG. 17 DIAGRAMA DE PAQUETES DE CLASES DEL DISEÑO.	81
FIG. 18 DIAGRAMA DE CLASE DEL PAQUETE CLASES DE LA HERRAMIENTA.	82
FIG. 19 DIAGRAMA DE CLASES DE PAQUETE CLASES TERRENO.	83
FIG. 20 DIAGRAMA DE SECUENCIA “MANIPULAR CONTROLADOR ST”, ESC “CREAR CONTROLADOR”	86
FIG. 21 DIAGRAMA DE SECUENCIA “MANIPULAR CONTROLADOR ST”, ESC “DESTRUIR CONTROLADOR” ...	86
FIG. 22 DIAGRAMA DE SECUENCIA “MANIPULAR PROPIEDADES FÍSICAS DE LOS OBJETOS”, ESC “CREAR PROPIEDADES”	87
FIG. 23 DIAGRAMA DE SECUENCIA “MANIPULAR PROPIEDADES FÍSICAS DE LOS OBJETOS”, ESC “ASOCIAR CONTROLADOR”	87
FIG. 24 DIAGRAMA DE SECUENCIA “CONFIGURAR PROPIEDADES DE LOS OBJETOS”, ESC “CONFIGURAR PUNTOS DE LA BASE”.	87
FIG. 25 DIAGRAMA DE SECUENCIA “CONFIGURAR PROPIEDADES DE LOS OBJETOS”, ESC “CONFIGURAR ESCALADO”	88
FIG. 26 DIAGRAMA DE SECUENCIA “CONFIGURAR PROPIEDADES DE LOS OBJETOS”, ESC “CONFIGURAR ALTURA SOBRE EL TERRENO”	88
FIG. 27 DIAGRAMA DE SECUENCIA “CONFIGURAR PROPIEDADES DE LOS OBJETOS”, ESC “CONFIGURAR DESPLAZAMIENTO DE LOS OBJETOS (DELANTERO)”	88
FIG. 28 DIAGRAMA DE SECUENCIA “CONFIGURAR PROPIEDADES DE LOS OBJETOS”, ES “CONFIGURAR DESPLAZAMIENTO DE LOS OBJETOS (IZQUIERDO)”	89
FIG. 29 DIAGRAMA DE SECUENCIA “CONFIGURAR PROPIEDADES DE LOS OBJETOS”, ESC “CONFIGURAR DESPLAZAMIENTO DE LOS OBJETOS (DERECHO)”	89

FIG. 30 DIAGRAMA DE SECUENCIA “CONFIGURAR PROPIEDADES DE LOS OBJETOS”, ESC “CONFIGURAR DESPLAZAMIENTO DE LOS OBJETOS (TRASERO)”	89
FIG. 31 DIAGRAMA DE SECUENCIA “CONFIGURAR PROPIEDADES DE LOS OBJETOS”, ESC “CONFIGURAR COEFICIENTE DE ROZAMIENTO DE LOS OBJETOS”	90
FIG. 32 DIAGRAMA DE SECUENCIA “ACTUALIZAR OBJETO 1”	90
FIG. 33 DIAGRAMA DE SECUENCIA “ACTUALIZAR OBJETO 2”	91
FIG. 34 DIAGRAMA DE SECUENCIA CU INCLUIDO “CALCULAR PUNTOS DE APOYO”	92
FIG. 35 DIAGRAMA DE SECUENCIA CU EXTENDIDO “CALCULAR 4 PUNTOS”	92
FIG. 36 PAQUETES DE COMPONENTES	102
FIG. 37 DIAGRAMA DE COMPONENTES “STKFOLLOWINGTERRAIN”	103

Índice de tablas

TABLA 1 ACTOR DEL SISTEMA	61
TABLA 2 CASO DE USO DEL SISTEMA, MANIPULAR CONTROLADOR	62
TABLA 3 CASO DE USO DEL SISTEMA, MANIPULAR PROPIEDADES FÍSICAS DE LOS OBJETOS	62
TABLA 4 CASOS DE USO DEL SISTEMA, CONFIGURAR PROPIEDADES DE LOS OBJETOS	62
TABLA 5 CASOS DE USO DEL SISTEMA, ACTUALIZAR OBJETO	63
TABLA 6 CASOS DE USO DEL SISTEMA, CALCULAR PUNTOS DE APOYO	63
TABLA 7 CASOS DE USO DEL SISTEMA, AJUSTAR 4 PUNTOS	63
TABLA 8 CU EXPANDIDOS, MANIPULAR CONTROLADOR	66
TABLA 9 CU EXPANDIDOS, MANIPULAR PROPIEDADES FÍSICAS DE LOS OBJETOS	67
TABLA 10 CU EXPANDIDOS, CONFIGURAR PROPIEDADES DE LOS OBJETOS	69
TABLA 11 CU EXPANDIDOS, ACTUALIZAR OBJETO	72
TABLA 12 CU EXPANDIDOS, CALCULAR PUNTOS DE APOYO	75
TABLA 13 CU EXPANDIDOS, AJUSTAR 4 PUNTOS	77
TABLA 14 DESCRIPCIÓN DE LA CLASE “CTERRAINTESTER”	84
TABLA 15 DESCRIPCIÓN DE LA CLASE “CPHYSICSPROPERTIES”	84
TABLA 16 DESCRIPCIÓN DE LA CLASE “CFOURPOINTS FITTER”	85
TABLA 17 DESCRIPCIÓN DE LA CLASE “CEQMATRIX3”	85