UNIVERSIDAD CATÓLICA ANDRÉS BELLO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA INFORMÁTICA

DISEÑO Y CONSTRUCCIÓN DE UN AMBIENTE DE INTERACCIÓN ORIENTADO AL DESARROLLO DE UN ESCRITORIO EXPERIMENTAL 3D Y A LA ADAPTACIÓN DE LOS DISPOSITIVOS DE DETECCIÓN DE MOVIMIENTOS QUE PERMITAN SU CONTROL

Este Jurado; una vez realizado el examen del presente trabajo ha evaluado su contenido con el resultado: 20 puntos (Mención Honorífica y Mención Publicación).

J U R A D O E X A M I N A D O R

Firma:	Firma:	Firma:
Nombre: Ramón Porras	Nombre: Biagio Cante	Nombre: Assaf Yamin

REALIZADO POR	Álvarez Hernández, Luis Eduardo.
	Valor García, Adrián Gabriel.
PROFESOR GUIA	Lic. Porras, Ramón.
FECHA	Octubre de 2008.

ÍNDICE GENERAL

DEDICATORIA - LUIS EDUARDO ÁLVAREZ HERNÁNDEZ i

AGRADECIMIENTOS - LUIS EDUARDO ÁLVAREZ HERNÁNDEZ ii

DEDICATORIA - ADRIÁN GABRIEL VALOR GARCÍA iii

AGRADECIMIENTOS - ADRIÁN GABRIEL VALOR GARCÍA iv

AGRADECIMIENTOS- COMUNES v

ÍNDICE GENERAL vi

LISTA DE TABLAS x

LISTA DE FIGURAS xi

SINÓPSIS xiii

INTRODUCCIÓN 1

CAPÍTULO 1. PROBLEMA 2

1.1 Planteamiento del problema 2

1.2 Objetivo General 2

1.3 Objetivos Específicos 3

1.4 Justificación 3

1.5 Limitaciones 5

1.6 Alcance 5

CAPÍTULO 2. MARCO TEÓRICO 7

2.1 Entorno de escritorio 7

2.2 Interacción hombre – ordenador (Human Interface Computer) 7

2.2.1 Principios de la Interacción Hombre Ordenador 9

2.3 Realidad virtual 9
2.3.1 Características de un sistema de realidad virtual 10

2.4 Computación gráfica 11
2.4.1 Aplicaciones 11

2.5 Gráficos 3D por ordenador 12
2.5.1 Fases para la creación de elementos gráficos 3D 12

2.6 Rastreadores ópticos 13

2.7 Radiación infrarroja 14
2.7.1 Diodo emisor de luz infrarroja: 15
2.7.2 Sensores infrarrojos: 17

2.8 Material reflectivo 17
2.8.1 Láminas reflectivas grado ingeniería: 17
2.8.2 Láminas reflectivas alta intensidad 18

2.9 Wii Remote 20

2.10 WiimoteLib 22

2.11 DirectX 22
2.11.1¿Por qué utilizar DirectX? 23
2.11.2 ¿Cómo funciona DirectX? 23

2.12 3Ds max 26

2.13 Panda DirectX 26

2.14 Visual estudio 2005 27

2.15 .NET Framework 27

2.16 Visual C# 29

2.17 Bluetooth 29

2.18 Blue Soleil 30

CAPÍTULO 3. MARCO METODOLÓGICO 31

3.1 Concepto Inicial 33

3.2 Diseño e implementación del prototipo inicial 33

3.3 Refinamiento del prototipo 33

3.3.1 Análisis de los requerimientos del prototipo *34*
3.3.2 Diseño del software *34*
3.3.3 Construcción del software *34*
3.3.4 Pruebas del software *34*
3.3.5 Diseño del hardware *34*
3.3.6 Construcción del hardware *35*
3.3.7 Pruebas del hardware *35*
3.3.8 Integración y evaluación del prototipo *35*

3.4 Completar prototipo final 36

3.5 Producto final 36

CAPÍTULO 4. DESARROLLO 37

4.1 Concepto Inicial 37

4.2 Diseño e implementación del prototipo inicial 39

4.3.Refiniamiento del Prototipo 39
4.3.1 Iteración I *39*
4.3.1.1 Análisis de requerimientos del prototipo 40
4.3.1.2 Diseño del software 40
4.3.1.3 Construcción del software 41
4.3.1.4 Pruebas del software 44
4.3.1.5 Diseño del hardware 45
4.3.1.6 Construcción del hardware 46
4.3.1.7 Pruebas del hardware 47
4.3.1.8 Integración y evaluación del prototipo 47
4.3.2 Iteración II *48*
4.3.2.1 Análisis de requerimientos del prototipo 48
4.3.2.2 Diseño Software 49
4.3.2.3 Construcción Software 51
4.3.2.4 Pruebas Software 53
4.3.2.5 Diseño del hardware 54
4.3.2.6 Construcción del hardware 55
4.3.2.7 Pruebas del hardware 56
4.3.2.8 Integración y evaluación del prototipo 56
4.3.3 Iteración III *57*
4.3.3.1 Análisis de requerimientos del prototipo 57
4.3.3.2 Diseño del software 58
4.3.3.3 Construcción del software 59
4.3.3 4 Pruebas del software 61
4.3.3.5 Diseño del hardware 61
4.3.3.6 Construcción del hardware 62
4.3.3.7 Pruebas del hardware 63
4.3.3.8 Integración y evaluación del prototipo 63

4.4 Completar prototipo final 63

4.5 Producto final 64

CAPÍTULO 5. RESULTADOS 65

CONCLUSIONES 67

RECOMENDACIONES 69

BIBLIOGRAFÍA 70

APÉNDICE A: DISEÑO E IMPLEMENTACIÓN DEL HARDWARE DEL PROTOTIPO INICIAL 77

APÉNDICE B. DISEÑO DEL CIRCUITO EMISOR DE INFRARROJO 82

APÉNDICE C. DISEÑO DEL REGULADOR DE VOLTAJE DE 12 VOLTIOS A 3 VOLTIOS 85

APÉNDICE D. CASOS DE USO 87

APÉNDICE E. INTEGRACIÓN DE COMPLEMENTOS DE USABILIDAD DEL HARDWARE EMISOR-RECEPTOR DE INFRARROJO 91

ANEXO A: MATERIAL REFLECTIVO 94

ANEXO B: EXTRACTOS DE LA DOCUMENTACIÓN DE LA LIBRERÍA WIMOTELIB 96

ANEXO C: PATENTE DEL WII REMOTE 119

ANEXO D: ALGORITMOS RELEVANTES 144

LISTA DE TABLAS

Tabla 1. Requerimientos de Software y Hardware, iteración I 40
Tabla 2. Requerimientos de Software y Hardware, iteración II 48
Tabla 3. Requerimientos de Software y Hardware, iteración III 57
Tabla 4. CU. Mover cursor 88
Tabla 5. CU. Obtener foco de ícono 88
Tabla 6. CU. Hacer clic en ícono 89
Tabla 7. CU. Eliminar ícono 90
Tabla 8. CU. Mover ícono 90

LISTA DE FIGURAS

Figura 1. Diagrama del espectro electromagnético....14
Figura 2. Fotografía Infrarroja de un mamífero....15
Figura 3. Representación simbólica del diodo LED....16
Figura 4. Arreglo de LEDs infrarrojos....17
Figura 5. Láminas reflectivas grado ingeniería....18
Figura 6. Láminas reflectivas alta intensidad....19
Figura 7. Láminas reflectivas alta intensidad....19
Figura 8. Wii Remote....21
Figura 9. Barra sensora....21
Figura 10. Funcionamiento del DirectX....24
Figura 11. Fases del prototipado evolutivo....31
Figura 12. Adaptación de las fases del prototipado evolutivo....32
Figura 13. Íconos 2D usados como modelo para el modelado 3D....41
Figura 14. Diseño de uno de los íconos con la aplicación 3Ds Max 2009....42
Figura 15. Modelos 3D de los íconos del escritorio y del cursor....42
Figura 16. Ventana de exportación del Panda DirectX....43
Figura 17. Modelos cargados en el ambiente virtual tridimensional....44
Figura 18. Baquelita para el ensamblaje del circuito emisor de infrarrojo....46
Figura 19. Hardware emisor-receptor de infrarrojo, iteración I....46
Figura 20. Accesorio reflector de infrarrojo, iteración I....47
Figura 21. Paralelepípedo, específicamente ortoedro....50
Figura 22. Técnica para obtener la distancia del wiimote a los puntos infrarrojos....51
Figura 23. Modelo cargado para establecer los límites del escritorio....52
Figura 24. Vista externa y en perspectiva del escritorio tridimensional limitado por el modelo....52
Figura 25. Ventana de configuración previa a iniciar el escritorio 3D....53
Figura 26. Hardware emisor-receptor de infrarrojo, Iteración II....55
Figura 27.Accesorios reflectivos, Iteración II....56
Figura 28. Ventana de confirmación de eliminación de ícono....59
Figura 29.Ventana de configuración rediseñada....60
Figura 30. Nuevo cursor del escritorio tridimensional....60
Figura 31. Escritorio tridimensional con una de las apariencias alternativas..61
Figura 32. Hardware emisor-receptor de infrarrojo obtenido con la tercera iteración....62
Figura 33. Diseño básico del circuito emisor de infrarrojo....77
Figura 34. Diseño del circuito emisor de infrarrojo....79
Figura 35. Modificación de baquelita....80
Figura 36. Prueba de acoplamiento emisor-receptor de infrarrojo....80
Figura 37. Circuito emisor de infrarrojo....81
Figura 38. Diseño básico de los grupos del circuito emisor de infrarrojo....82
Figura 39. Diseño del circuito emisor de infrarrojo....83

Figura 40. Diseño del circuito regulador de voltaje para el dispositivo receptor de infrarrojo.....85
Figura 41. Diagrama de casos de uso del sistema. 87
Figura 42. Interruptor incorporado al hardware.....91
Figura 43. Conector hembra incorporado al hardware.....91
Figura 44. Conector macho de la fuente de poder del hardware. 91
Figura 45. Interruptor de encendido/apagado y conector de fuente de poder incorporados en el hardware. 92
Figura 46. Conector macho de la fuente de poder del hardware. 92
Figura 47. Lámina acrílica incorporada al hardware. 93

SINÓPSIS

Este Trabajo Especial de Grado titulado "Diseño y Construcción de un Ambiente de Interacción Orientado al Desarrollo de un Escritorio Experimental 3D y a la Adaptación de los Dispositivos de Detección de Movimientos que Permitan su Control" consiste en el diseño y construcción de un ambiente de interacción hombre-ordenador que permite controlar una aplicación de escritorio con los movimientos de la mano en las tres dimensiones.

El desarrollo de este trabajo se realizó basándose en la metodología de prototipado evolutivo, la cual consiste en realizar un prototipo inicial al que se le agregan nuevas funcionalidades y mejoras en cada iteración, y así se van generando productos funcionales durante la evolución del proyecto.

Se diseñó y desarrolló un software utilizando la aplicación de modelado 3Ds Max, el API DirectX y el lenguaje de programación C#; y se diseñó y construyó un hardware emisor-receptor de luz infrarroja que utiliza los principios básicos del rastreo óptico para determinar los movimientos de la mano del usuario y representarlos en el sistema.

Se implementaron una serie de funcionalidades clásicas de un entorno de escritorio que pueden ser aplicadas a los elementos mostrados en el ambiente tridimensional, demostrando así las capacidades de interacción hombre-ordenador que se pueden conseguir con las tecnologías usadas en este trabajo.

INTRODUCCIÓN

Desde la aparición de los ordenadores las personas han interactuado con éstos utilizando de manera clásica dispositivos como el teclado y el mouse y, gracias al creciente avance de la tecnología se ven productos cada vez más novedosos, los cuales utilizan estos avances con el objetivo de brindar a los usuarios la manera más natural de interactuar con estas máquinas que hoy en día son tan indispensables.

En la actualidad, una de las tecnologías de mayor atractivo es la relacionada con la detección de movimientos de las personas usando diversas técnicas que han sido aprovechadas especialmente en el área de los videojuegos, observándose así que tecnologías como ésta, que antes solo podían ser explotadas en áreas puntuales dado su alto costo, son ahora cada vez mas usadas para interactuar con elementos que usan en el día a día, con lo que se plantea la oportunidad de aprovechar esta tecnología para interactuar de manera más natural con los ordenadores.

La inquietud por la aplicación de las ya mencionadas técnicas al manejo del computador para ampliar y enriquecer sus alcances y posibilidades condujo al diseño y construcción de un ambiente de interacción que permita la adaptación de dispositivos de detección de movimientos, y al desarrollo de un escritorio experimental tridimensional que sea controlado por la mano del usuario.

CAPÍTULO 1. PROBLEMA

1.1 Planteamiento del problema

Desde los inicios de la era computacional y hasta el día de hoy, los usuarios de las computadoras han tenido que acoplarse al uso de diferentes dispositivos de control y de interacción con éstas (mouse, teclado, monitor, touchpad, etc.) sin que en muchas ocasiones estos dispositivos mantengan una relación parecida, en su modo de uso, a la manera en que se interactúa día a día con los objetos en el mundo real, por lo que es bastante común que los dispositivos de interfaz hombre-ordenador tengan limitantes bien sean en relación a la libertad de movimientos, visibilidad o retroalimentación al utilizarlos, y a pesar de ello este tipo de interacciones son vistas como la manera natural en la que se debe interactuar con un computador; estas interacciones han sido causa de estudio a través de los años con la intención de mejorar la experiencia del usuario, pero la mayoría de ellas han sido fallidas, aunque hay que destacar que en los últimos meses se han hecho algunos avances exitosos en este aspecto.

Uno de estos dispositivos es el mouse, que aunque su manipulación es hoy en día muy natural e intuitiva debido a su constante uso en los últimos años, no escapa de tener limitantes, como lo son el interactuar en un ambiente bidimensional y esto en parte ha imposibilitado el desarrollo de software, como las aplicaciones de escritorio en ambientes tridimensionales en donde el usuario interactúe con los elementos como lo haría en la realidad.

1.2 Objetivo General

Diseñar y construir un ambiente de interacción entre el usuario y el computador, que se oriente al desarrollo de un escritorio experimental

tridimensional y a la adaptación de los dispositivos de detección de movimientos de la mano que permitan el control del escritorio.

1.3 Objetivos Específicos

- Evaluar las tecnologías existentes para determinar las más adecuadas para el desarrollo del sistema.
- Evaluar funcionalidades acordes y viables que indiquen la mejor forma de interacción entre el usuario y el computador para ser implementados en un escritorio tridimensional, basándose en el estudio de las necesidades de usuarios reales y de escritorios ya existentes.
- Desarrollar un escritorio experimental tridimensional.
- Diseñar y adaptar el hardware necesario para la detección de movimientos de la mano.
- Desarrollar un software de comprobación de comunicación y ajuste del hardware.
- Desarrollar el componente de comunicación entre los dispositivos de hardware de detección de movimientos y el computador.

1.4 Justificación

En la actualidad, a pesar de los continuos intentos en la búsqueda de mejoras de los dispositivos de control de los computadores, éstos siguen siendo los mismos de hace décadas, principalmente representados por el teclado y el mouse, lo que ha limitado el desarrollo pleno de nuevas alternativas en la interacción hombre-ordenador y con esto el desarrollo de software diferente al de ambientes bidimensionales y acordes con estas

nuevas alternativas de interacción.

Son conocidos los grandes adelantos que se han hecho en el uso de tecnologías de reconocimiento de movimientos en áreas como los videojuegos, en donde se destaca el Nintendo Wii, con el cual de una manera nunca antes vista las personas interactúan con los juegos a través de un control inalámbrico que detecta sus movimientos. Lo que lleva a pensar que la implementación de esta tecnología adaptada para el uso de un computador se presenta como una opción innovadora y que, aunado a la capacidad de procesamiento de imágenes que existe hoy en día, el diseño y construcción de un ambiente de interacción de este tipo orientado al desarrollo de un escritorio experimental tridimensional y a la adaptación de dispositivos de detección de movimientos de la mano que permitan el control del mismo, luce como un buen ejemplo de lo que puede ser el futuro cercano en relación al manejo cotidiano de los computadores.

Los escritorios que se presentan en los actuales sistemas operativos son a nivel funcional muy parecidos, excepto algunos que se distinguen por sus características visuales debido a que le presentan al usuario ciertos efectos tridimensionales, lo que le produce más satisfacción, pero estos efectos siempre están limitados a la bidimensionalidad del mouse que es el que se utiliza para su manejo. Por tal motivo la investigación y el desarrollo de un escritorio experimental que contribuya a dar los primeros pasos para romper con la limitante antes expuesta surge como una alternativa viable en mira a los adelantos de la nueva era.

El desarrollo de este sistema permite integrar conocimientos adquiridos en la carrera de Ingeniería Informática, como lo son el diseño y la construcción de un software y de los dispositivos hardware encargados de interactuar con dicho software.

1.5 Limitaciones

En relación al escritorio experimental:

- La calidad de las características visuales (definición de imágenes, sombreados, iluminación) estará limitada, debido a que este aspecto involucra conocimientos más profundos en áreas como el diseño gráfico, y adentrarse en el perfeccionamiento de este aspecto sin este conocimiento implicaría el uso de más tiempo del estipulado.
- El espacio de trabajo tendrá un tamaño limitado en las tres dimensiones.
- Los íconos contenidos en el escritorio no ejecutarán aplicaciones.

En relación al dispositivo de detección de movimientos:

- El alcance de la señal de comunicación entre el dispositivo de detección de movimientos y el computador estará limitado por la distancia y por los obstáculos que exista entre ellos.
- Los movimientos con la mano que realice el usuario al interactuar con el software no podrán ser bruscos.

1.6 Alcance

En relación al escritorio experimental:

- Poseerá características visuales tridimensionales.
- El espacio de trabajo contemplará las tres dimensiones.
- Contendrá una serie de íconos que podrán ser cambiados de posición en el espacio de trabajo.
- Poseerá una serie de funcionalidades típicas de escritorio como eliminar, seleccionar, mover, hacer clic; y cambiar algunas

características básicas de la apariencia del escritorio.

En relación al dispositivo de detección de movimientos:

- Se basará en un accesorio reflector que se coloque en la mano del usuario y en un dispositivo emisor-receptor del lado del computador.
- La comunicación entre los dispositivos y accesorios será inalámbrica.
- El sistema contendrá un componente de comunicación entre el dispositivo de detección de movimientos y el computador.
- La manera en que el usuario interactuará con el software será usando el accesorio colocado en la mano y realizando movimientos naturales con ella en el espacio.

El desarrollo de este proyecto podría representar los primeros pasos de una investigación mas profunda por parte de grandes compañías de tecnología informática en lo que se refiere a la interacción y el manejo de los computadores que son usados día a día en los hogares, con miras a que en un futuro cercano los resultado de estas investigaciones sean la base de productos que sean utilizados masivamente y su uso sea natural, como lo es la interacción entre el ratón con las aplicaciones existentes en la actualidad.

CAPÍTULO 2. MARCO TEÓRICO

2.1 Entorno de escritorio

Un entorno de escritorio (en inglés, Desktop Environment) es un conjunto de software para ofrecer al usuario de un ordenador una interacción amigable y cómoda.

El software es una solución completa de interfaz gráfica de usuario o GUI, ofrece íconos, barras de herramientas, programas e integración entre aplicaciones con habilidades como, arrastrar y soltar (drag&drop).

En general cada entorno de escritorio se distingue por su aspecto y comportamientos particulares, aunque algunos tienden a imitar características de escritorios ya existentes.

El primer entorno moderno de escritorio que se comercializó fue desarrollado por Xerox en los años 1980. Actualmente el entorno más conocido es el ofrecido por la familia Windows aunque existen otros como los de Macintosh (Classic y Cocoa) y de código abierto (o software libre) como GNOME, KDE, CDE o Xfce. [Entorno escritorio. Sin fecha]

2.2 Interacción hombre – ordenador (Human Interface Computer)

La interacción hombre – ordenador tiene su origen en los años setenta con la explosión tecnológica que trajo consigo la necesidad de una comunicación directa entre ambos, de ahí que la HIC se ocupa fundamentalmente del análisis y diseño de interfaces de usuario.

La interfaz de usuario está constituida por una serie de dispositivos, tanto físicos como lógicos, que permiten al hombre interactuar de una manera precisa y concreta con un sistema. El planteamiento del área de HCI estudia las relaciones que se establecen entre las personas y la tecnología.

De esta forma, la interacción hombre ordenador se relaciona con el diseño de sistemas para que las personas puedan llevar a cabo sus actividades productivamente con un buen nivel de manejabilidad, usabilidad y amigabilidad. Esto se concreta en términos de simplicidad, fiabilidad, seguridad, comodidad y eficacia.

La HCI estudia:

- El hardware, el software y la repercusión de ambos en la interacción.
- Los modelos mentales de los usuarios frente al modelo del sistema.
- Las tareas que desempeña el sistema y su adaptación a las necesidades del usuario.
- El diseño dirigido al usuario, no a la máquina (*user-centered design*).
- El impacto organizacional.

La HCI comienza a ver sus frutos en el momento en que los ordenadores dejan de ser un misterio para los usuarios no profesionales de la informática, es decir, para los programadores, y salen al mercado. El paso necesario para que se diera esta popularización era la implementación de facilidades en el manejo de los programas, especialmente en las interfaces. Así, la interacción con el ordenador va pasando desde la línea de comandos en la que se le daban las órdenes en un lenguaje nada parecido al de las personas, a otras formas más amigables como los menús de opciones, los formularios que se rellenan o la actual manipulación directa, en la que el usuario tiene la posibilidad de manejar los objetos de una forma similar a la que acostumbra en la vida real.

Existen por tanto diversos factores que hay que considerar en la HCI y que además están interrelacionados entre sí:

a) Factores ergonómicos.

b) Factores psicológicos de los usuarios.

c) Factores de diseño del sistema.

d) Factores sociales y organizacionales. [Marcos. Sin fecha]

2.2.1 Principios de la Interacción Hombre Ordenador

Todas las reglas de diseño centrado en usuarios tienen su origen en estos principios:

1. Visibilidad del estado del sistema.
2. Correspondencia entre el sistema y el mundo real.
3. Control y libertad del usuario.
4. Consistencia y estándares.
5. Evitar errores.
6. Reconocimiento vs. Recuerdo.
7. Flexibles y eficientes.
8. Diseño minimalista.
9. Reconocer, diagnosticar y recuperarse de los errores.
10. Ayuda y documentación.

2.3 Realidad virtual

Son muchas y diversas las definiciones dadas para la realidad virtual, entre ellas podemos mencionar:

Un ambiente gráfico tridimensional (también llamado virtual) creado por ordenador, donde el o los usuarios pueden interactuar con el ambiente y los objetos gráficos que contiene el mundo virtual, utilizando técnicas de

manipulación directa, y obteniendo información visual, táctil y auditiva principalmente, pudiendo usarse más de un sentido a la vez.

La realidad virtual se refiere a las simulaciones en un ordenador del mundo real por medio de imágenes tridimensionales y componentes externos para permitir que lo usuarios interactúen con la simulación. Los usuarios se mueven por una realidad virtual como si estuviesen en un mundo real.

Una manera más sencilla de describir la realidad virtual sería como la manipulación de los sentidos humanos (hasta el momento el tacto, la visión y la audición) por medio de ambientes tridimensionales ofrecidos por la computadora en el que uno o varios participantes interactúan de manera rápida e intuitiva de manera tal que el usuario toma como real el entorno generado por la computadora.

La realidad virtual puede tener diferentes grados de "inmersión" (sensación de estar dentro del mundo virtual) dependiendo de los dispositivos de entrada y salida utilizados.

2.3.1 Características de un sistema de realidad virtual

Interacción: le permite al usuario manipular los datos dentro de un sistema de realidad virtual, permitiendo que éste responda a los estímulos de la persona, creando interdependencia entre ellos.

Inmersión: es la capacidad del sistema de realidad virtual de enfocar la atención del usuario, convirtiendo una base de datos en experiencias. Lo que se pretende es sustituir la percepción humana, por una percepción generada por el ordenador.

Multisensorialidad: se refiere a la integración de sonidos, imágenes y sensaciones hápticas de forma armónica.

Hardware especializado para un sistema de realidad virtual: Los sistemas de realidad virtual, en su propósito de lograr una inmersión total por parte del usuario, hacen uso de hardware especializado, dentro de los más comunes se pueden mencionar: Casco, Estereoscópico, Lentes, Estereoscópicos, Guante Electrónico, Trajes. [Marvi, Montezuma. Octubre 2005]

2.4 Computación gráfica

La computación gráfica es la rama de las ciencias de la computación que se encarga del estudio, diseño y trabajo del despliegue de imágenes en la pantalla de un ordenador a través de las herramientas proporcionadas por la física, la óptica, la térmica, la geometría, entre otros.

2.4.1 Aplicaciones

Está rama de computación es utilizada en:

- *Ingeniería*: Las actividades de ingeniería pueden dividirse en cinco áreas: diseño, análisis, dibujo, fabricación/construcción/procesamiento y control de calidad.
- *Ciencias*: Los científicos pueden emplear las gráficas por computadora de muchas maneras.
- *Arte*: El artista siempre ha podido escoger los medios de expresión adecuados para su talento. Las gráficas por ordenador son otro medio del que dispone para ampliar su libertad de expresión.
- *Educación*: Un beneficio importante que aportan los ordenadores y las gráficas es en el ambiente educativo así como también en el área de capacitación y sistemas especiales. [Computación Gráfica. Sin fecha]

2.5 Gráficos 3D por ordenador

El término gráficos 3D por ordenador se refiere a trabajos de arte gráfico que fueron creados con ayuda de computadoras y programas especiales 3D. En general, el término puede referirse también al proceso de crear dichos gráficos, o el campo de estudio de técnicas y tecnología relacionadas con los gráficos 3D.

Un gráfico 3D difiere de uno 2D principalmente por la forma en que ha sido generado. Este tipo de gráficos se origina mediante un proceso de cálculos matemáticos sobre entidades geométricas tridimensionales producidas en un ordenador, y cuyo propósito es conseguir una proyección visual en dos dimensiones para ser mostrada en una pantalla o impresa en papel. [Animación 3D. Sin fecha]

En general, el arte de los gráficos 3D es similar a la escultura o la fotografía, mientras que el arte de los gráficos 2D es análogo a la pintura. En los programas de gráficos por computadora esta distinción es a veces difusa: algunas aplicaciones 2D utilizan técnicas 3D para alcanzar ciertos efectos como iluminación, mientras que algunas aplicaciones 3D primarias hacen uso de técnicas 2D. [Wright R. y Lipchak, B. 2005]

2.5.1 Fases para la creación de elementos gráficos 3D

- *Modelado:* La etapa de modelado consiste en ir dando forma a objetos individuales que luego serán usados en la escena. Existen diversos tipos de geometría para modelador con NURBS y modelado poligonal o subdivisión de superficies. Además, aunque menos usado, existe otro tipo llamado "modelado basado en imágenes" el cual consiste en convertir una fotografía a 3D mediante el uso de diversas técnicas, de las cuales, la más conocida es la fotogrametría.

- *Texturizado*: En esta etapa se define la forma en que le afecta la luz al modelo, para ello se usan materiales y algoritmos que controlan la incidencia de la luz, produciendo materiales de tipo: Anisótropo, Lambert o Blin.
- *Iluminación*: Se basa en la creación de luces de diversos tipos puntuales, direccionales en área o volumen, con distinto color o propiedades. Este proceso es la clave de una animación.
- *Animación*: Los objetos se pueden animar en cuanto a transformaciones básicas en los tres ejes (XYZ), Rotación, Escala o Traslación.
- *Renderizado*: Se llama rénder al proceso final de generar la imagen 2D o animación a partir de la escena creada. Esto puede ser comparado a tomar una foto o en el caso de la animación, a filmar una escena de la vida real. Generalmente se buscan imágenes de calidad foto realista, y para este fin se han desarrollado muchos métodos especiales. Las técnicas van desde las más sencillas, como el rénder de alambre (wireframe rendering), pasando por el rénder basado en polígonos, hasta las técnicas más modernas como el Scanline Rendering, el Raytracing, la radiosidad o el mapeado de fotones. [Gráficos 3D por Computadora. 27/06/2008]

2.6 Rastreadores ópticos

Los rastreadores ópticos son dispositivos que utilizan alguna técnica para detectar movimientos; estas técnicas son muy diversas y la más utilizada en dispositivos de realidad virtual es denominada PLADAR (Pulsed Light Angle Direction And Range). Ésta utiliza luz infrarroja siendo la velocidad de la luz tan rápida que la distancia entre el emisor y receptor se calcula midiendo la fuerza o intensidad de luz recibida. [Hollands. 1996]

Otros sistemas de rastreo óptico están basados en cámaras. Una de las técnicas con cámaras consiste en dirigirla hacia el usuario, el cual tiene un conjunto de marcas reflectivas de luz sobre la parte del cuerpo que desea rastrear, cuando estas marcas se encuentran en el campo visual de la cámara se puede calcular la posición y orientación comparando la posición de las marcas vista por la cámara y la distancia real entre ellas. [Federico. 2005]

2.7 Radiación infrarroja

La radiación infrarroja, radiación térmica o radiación IR es un tipo de radiación electromagnética (Ver Figura 1) de mayor longitud de onda que la luz visible, pero menor que la de las microondas. Consecuentemente, tiene menor frecuencia que la luz visible y mayor que las microondas. Su rango de longitudes de onda va desde unos 700 nanómetros hasta 1 milímetro. La radiación infrarroja es emitida por cualquier cuerpo cuya temperatura sea mayor que 0 Kelvin (cero absoluto), es decir, -273 grados centígrados. El nombre de infrarrojo significa por debajo del rojo pues su comienzo se encuentra adyacente al color rojo del espectro visible. [¿Qué es el Infrarrojo?. 21/09/2000]

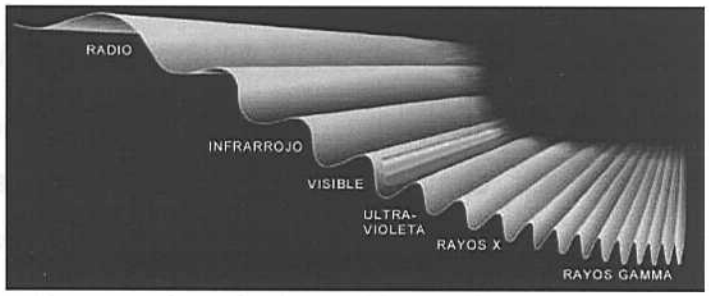


Figura 1. Diagrama del espectro electromagnético
Fuente: [¿Qué es el Infrarrojo?. 21/09/2000]

La materia, por su caracterización energética emite radiación. En general, la longitud de onda donde un cuerpo emite el máximo de radiación es proporcional a la temperatura de éste. De esta forma la mayoría de los objetos a temperaturas cotidianas tienen su máximo de emisión en el infrarrojo. Los seres vivos, en especial los mamíferos, emiten una gran proporción de radiación en la parte del espectro infrarrojo, debido a su calor corporal. (Ver Figura 2)

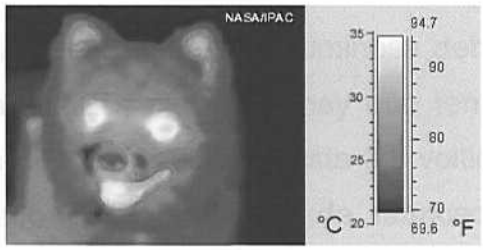


Figura 2. Fotografía Infrarroja de un mamífero
Fuente [Radiación Infrarroja. 13/07/2005]

Los infrarrojos se utilizan en los equipos de visión nocturna cuando la cantidad de luz visible es insuficiente para ver los objetos. La radiación se recibe y después se refleja en una pantalla. Los objetos más calientes se convierten en los más luminosos. [Radiación Infrarroja. 13/07/2005]

2.7.1 Diodo emisor de luz infrarroja:

También conocido como IRLED (Infra Red Light Emitting Diode) es un dispositivo semiconductor (diodo) que emite luz infrarroja de espectro reducido cuando circula corriente a través de él.

El dispositivo semiconductor está comúnmente encapsulado en una cubierta de plástico de mayor resistencia que las de vidrio que usualmente se emplean en las lámparas incandescentes. Usualmente un LED es una fuente

de luz compuesta con diferentes partes, razón por la cual el patrón de intensidad de la luz emitida puede ser bastante complejo. (Ver Figura 3)

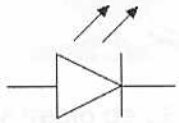


Figura 3. Representación simbólica del diodo LED
Fuente elaboración propia

Para obtener una buena intensidad luminosa debe escogerse bien la corriente que atraviesa el LED; para ello, hay que tener en cuenta que el voltaje de operación va desde 1,8 voltios hasta 3,8 voltios aproximadamente (lo que está relacionado con el material de fabricación) y la gama de intensidades que debe circular por él varía según su aplicación. Valores típicos de corriente directa de polarización de un LED común están comprendidos entre los 10 mA y los 40 mA. En general, los LEDs suelen tener mejor eficiencia cuanto menor es la corriente que circula por ellos, con lo cual, en su operación de forma optimizada, se suele buscar un compromiso entre la intensidad luminosa que producen (mayor cuanto más grande es la intensidad que circula por ellos) y la eficiencia (mayor cuanto menor es la intensidad que circula por ellos). [Nichia Unveils White LED with 150 lm/W Luminous. 21/12/2006]

Los diodos infrarrojos se emplean desde mediados del siglo XX en mandos a distancia (controles remotos) así como también en dispositivos detectores. Cuando se conectan una serie de diodos LEDs se pueden obtener emisores de luz infrarroja de mayor área de visualización y potencia. (Ver Figura 4)

Figura 4. Arreglo de LEDs infrarrojos
Fuente [Infrared Light Emitting Diode Illuminators Ready-to-Use PCB & Kits. Sin fecha]

2.7.2 Sensores infrarrojos:

Un sensor infrarrojo es un dispositivo electrónico capaz de medir la radiación electromagnética infrarroja de los cuerpos en su campo de visión.

2.8 Material reflectivo

Es un material especial usado frecuentemente para la confección de alertas viales y existen diversos tipos, es importante describir cada una de ellos para saber cuando utilizar uno u otro.

2.8.1 Láminas reflectivas grado ingeniería:

Básicamente las láminas reflectivas están conformadas por un material base en cuya cara inferior se aplica un pegamento protegido por un liner protector.

En la cara superior se aplica una película reflectora metálica y sobre esta una película de esferas microscópicas de vidrio orientadas electroestaticamente que son las que reflejan la luz.

Sobre esta película se aplica una capa de laca protectora que está en contacto con el medio ambiente y la protege de los agentes agresivos (rayos UV, viento, lluvia, etc.) (Ver Figura 5)

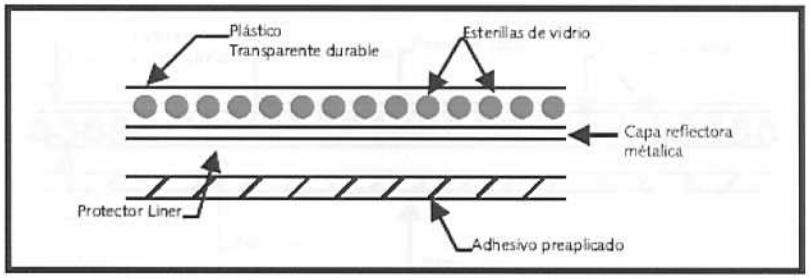


Figura 5. Láminas reflectivas grado ingeniería
Fuente:[Láminas Reflectivas.24/10/2005]

Este tipo de láminas se utilizan para la confección de carteles de señalización vial, sobre todo en caminos secundarios o en lugares donde el tránsito no es muy intenso. Se presentan en colores blanco, amarillo, rojo, verde, azul, naranja y, cebrado naranja y blanco.

2.8.2 Láminas reflectivas alta intensidad

Las láminas reflectivas de alta intensidad tienen el mismo principio de construcción que las láminas grado ingeniería, la diferencia radica en que se usa la técnica de encapsulamiento.

La película que cubre las esferillas de vidrio no descansa sobre las esferillas, sino sobre paredes construidas de forma hexagonal u otra forma similar. Esto permite la presencia de aire entre las esferillas y la película que la cubre, este tipo de construcción es lo que hace posible tener un brillo superior en 3.5 veces a las láminas grado ingeniería un grado de angularidad muy superior a cualquier tipo de superficie o lámina.

La presentación es similar a las láminas grado ingeniería. (Ver Figura 6)

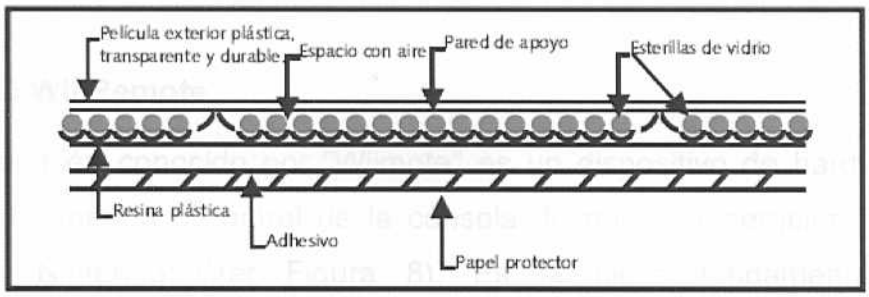


Figura 6. Láminas reflectivas alta intensidad.
Fuente [Láminas Reflectivas.24/10/2005]

Este tipo de láminas son utilizadas para la confección de carteles viales en lugares donde la visibilidad debe ser óptima. Actualmente es norma el uso este tipo de láminas para carteles en autopistas e identificación de transporte automotor de carga y pasajeros (Ver figura 7). [Láminas Reflectivas.24/10/2005]

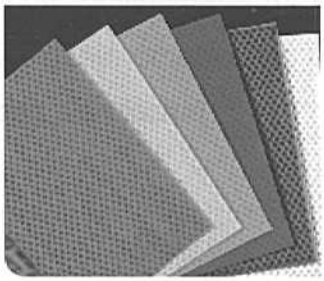


Figura 7. Láminas reflectivas alta intensidad.
Fuente [Láminas Reflectivas.24/10/2005]

Entre los principales beneficios se encuentra:

- Alta reflectividad inicial, estable y retenida.
- Mayor visibilidad en cualquier situación, angularidad hasta 90´

2.9 Wii Remote

También conocido por "Wiimote" es un dispositivo de hardware que sirve como mando o control de la cónsola de nueva generación Wii de la empresa Nintendo (Ver Figura 8). Es la pieza fundamental en el funcionamiento revolucionario de la cónsola dentro de la industria, ya que éste es capaz de detectar la distancia, altura, dirección y ángulos de inclinación, también incorpora un sistema de vibración y un mini altavoz. Además, incorpora la función de puntero, con la que se puede apuntar a la pantalla con el mando, para ello la consola envía luz infrarroja mediante una pequeña barra que contiene dos emisores infrarrojos en los extremos y que se coloca sobre o bajo el televisor (Ver Figura 9). El mando tiene un receptor que interpreta los datos, dicho receptor consta de una cámara detectora de luz infrarroja, y la detección de movimientos triaxial se realiza gracias a que el dispositivo cuenta con un acelerómetro. La conexión con la cónsola se realiza por Bluetooth. [Méndez. Sin fecha]

Figura 8. Wii Remote.
Fuente [Wii Remote. Sin Fecha]

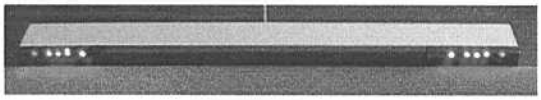


Figura 9. Barra sensora
Fuente [Jugando con el Wiimote y GNU/Linux. Sin Fecha]

El diseño del Wii Remote no se basa en los tradicionales mandos para los videojuegos. A diferencia de ellos, es similar a un control remoto de televisión, creado para ser utilizado en una sola mano y de la manera más intuitiva posible. En su cara frontal, el Wii Remote presenta los botones "A", "1", "2", "+", "-", "HOME" y "POWER", junto a una cruz direccional. En la parte anterior sólo presenta el botón "B", en un formato similar a un gatillo. Adicionalmente, en su parte frontal incluye un altavoz y cuatro luces numeradas que indican el número de jugador al que corresponde tal mando. En la parte inferior viene unida una correa de seguridad que se amarra a la muñeca para evitar soltar y dañar el control de forma accidental. [Méndez. Sin fecha]

Entradas

- Cámara infrarroja.
- Acelerómetro.
- 12 botones digitales.

Salidas

- Táctil – motor vibrador.
- Auditivo – pequeño altavoz.
- Visual – LEDs azules de estatus.

Otros

- Bluetooth.

- Puerto de expansión.
- Memoria en la placa.
- Baterías. [Lee. 2008]

2.10 WiimoteLib

WiimoteLib es una librería manejada con .NET que se encarga de ser una interfaz de comunicaciones entre el Nintendo Wii Remote y una aplicación desarrollada con .NET; fue desarrollada por Brian Peek Microsoft C# MVP y su última distribución es del año 2008. [.NET Manager Library for the Nintendo Wii Remote. 16/09/2008]

2.11 DirectX

Microsoft DirectX es una colección avanzada de interfaces de programación de aplicaciones (APIs) integrada a los sistemas operativos Microsoft Windows.

Este conjunto de APIs, mantiene una plataforma de desarrollo de aplicaciones multimedia estándar para aplicaciones Windows, permitiendo a los programadores de software acceder al hardware especializado sin tener que escribir código específico de cada tipo de hardware.

Cuando una aplicación o un juego son escritos para DirectX, el programador no tiene que preocuparse por exactamente qué tarjeta de sonido o por el adaptador gráfico que el usuario final podría tener en su máquina, DirectX se encarga de eso por él.

DirectX juega un papel en muchas funciones, incluyendo renderización 3D, reproducción de video, interfaces para joysticks y mouses, gestión de redes para multi-jugador y muchos más.

existe un HEL (capa de emulación de hardware) que se encarga de emular las funciones que el hardware no puede manejar. (Ver figura 10)

De esta manera se puede mantener la compatibilidad de DirectX con todo tipo de hardware más viejo y permite utilizar aplicaciones nuevas con estos hardwares, pero con cierta penalización en performance o en calidad. [DirectX. Sin fecha]

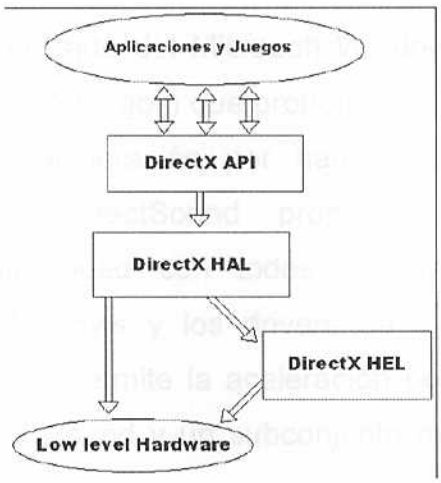


Figura 10. Funcionamiento del DirectX.
Fuente: [DirectX. Sin fecha]

El API de DirectX está compuesto de los siguientes componentes:

DirectDraw: es un administrador de la memoria de video, permite manipular la memoria de video con facilidad, mientras aprovecha todo tipo de capacidades de los tipos diferentes de adaptadores de video sin depender de un tipo particular de hardware.

Direct3D: es un conjunto completo de servicios gráficos 3D en tiempo real, que se encarga de todo el renderizado basado en software-hardware de todo el pipe-line gráfico (transformaciones, iluminación, y rasterización) y el acceso transparente a la aceleración del hardware. Direct3D es completamente escalable, permitiendo que todo o una parte del 3D pipe-line

sea acelerado por el hardware. Direct3D expone las avanzadas capacidades de los aceleradores del hardware 3D, incluyendo el z-buffering, el anti-aliasing, alpha blending, mipmapping, efectos atmosféricos, y perspective-correct texture mapping. La integración con otras tecnologías de DirectX permite a Direct3D tener otras características tales como video mapping, hardware 3D rendering, planos overlay en 2D, e inclusive sprites, proporcionando capacidades de gráficos 2D y 3D en aplicaciones multimedia.

DirectSound: es la parte del Microsoft Windows Game SDK (Kid de Desarrollo de Software para juegos) que proporciona utilidades de mezcla de sonido a baja-latencia, aceleración por hardware, y acceso directo al dispositivo de sonido. DirectSound proporciona esta funcionalidad manteniendo la compatibilidad con todos los programas basados en versiones previas de Windows y los drivers de distintos dispositivos de hardware. DirectSound 3D permite la aceleración por hardware de sonidos 3D utilizando sistemas surround y un subconjunto de la norma de EAX de Creative.

DirectInput: proporciona acceso rápido y consistente a palancas de mando analógicas y digitales (Joysticks). El API de DirectInput mantiene la compatibilidad con el Kit de Desarrollo de Software (SDK) Win32 de Microsoft, pero ha mejorado la capacidad de respuesta y la fiabilidad cambiando el modelo del driver de dispositivo. Los drivers de DirectInput también utilizan el registro de Windows para guardar las configuraciones para las palancas de mando normales, y la información de la calibración para las palancas de mando previamente configuradas.

DirectPlay: es una interfaz de software que simplifica el acceso a los servicios de comunicación. Provee a los juegos una manera de comunicarse entre sí que es independiente del medio de transporte subyacente, protocolo, o el servicio en línea. DirectPlay también hace posible lanzar y correr juegos fácilmente a través de Internet usando el API de DirectX.

DirectMusic: Proporciona a los programadores un API unificado para el manejo de música, aprovechando las características específicas de diversos tipos de hardware y manteniendo la compatibilidad con aplicaciones más antiguas.

2.12 3Ds max

3Ds Max es uno de los programas de animación 3D más utilizados. Dispone de una sólida capacidad de edición, una arquitectura de plugins y una larga tradición en plataformas Microsoft Windows. 3Ds Max es utilizado en mayor medida por los desarrolladores de videojuegos, aunque también en el desarrollo de proyectos de animación como películas o anuncios de televisión, efectos especiales y en arquitectura. [3Ds Max. 13/09/2008]

2.13 Panda DirectX

El Panda DirectX es un plugin para la aplicación 3Ds Max desarrollado por Pandasoft y de distribución gratuita cuya función es servir de exportador de archivos .X, esto quiere decir exportar el modelo creado en 3Ds Max a un formato específico para su posterior uso en otras aplicaciones usando DirectX.

Este exportador de archivos .X ha sido desde hace un tiempo, la principal herramienta para los interesados en el desarrollo de aplicaciones 3D usando los modelos exportados desde el 3Ds Max. La introducción de DirectX 8 supuso la introducción por parte de Microsoft de un exportador con el código fuente. Sin embargo, a pesar de que el exportador Microsoft es un buen ejemplo de código fuente, tiene algunos problemas con los modelos exportados. Por lo tanto, el Panda DirectX se ha posicionado como la alternativa al exportador de Microsoft y permite a aquellos que quieran exportar un archivo .X de su modelo, hacerlo sin la complicación de tener que

utilizar Visual Studio para depurar la versión de Microsoft. [PandaSoft. Sin fecha]

2.14 Visual estudio 2005

Es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones Web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C++, Visual C# y Visual J# utilizan el mismo entorno de desarrollo integrado (IDE), que les permite compartir herramientas y facilita la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes aprovechan las funciones de .NET Framework, que ofrece acceso a tecnologías clave para simplificar el desarrollo de aplicaciones Web ASP y Servicios Web XML. [Introducción a Visual Studio. Sin fecha]

2.15 .NET Framework

Es un entorno multilenguaje que permite generar, implantar y ejecutar aplicaciones y Servicios Web XML. Consta de tres partes principales:

- *Common Language Runtime*: el motor en tiempo de ejecución desempeña una función durante la ejecución como el desarrollo de los componentes. Cuando el componente se está ejecutando, el motor en tiempo de ejecución, es responsable de administrar la asignación de memoria, iniciar y detener subprocesos y procesos, y hacer cumplir la directiva de seguridad, así como satisfacer las posibles dependencias del componente sobre otros componentes. Durante el desarrollo, el papel del motor en tiempo de ejecución cambia ligeramente; a causa de la gran automatización que permite (por ejemplo, en la administración de memoria), el motor simplifica el trabajo del desarrollador. En concreto, funciones tales como la reflexión reducen

de forma espectacular la cantidad de código que debe escribir el desarrollador para convertir la lógica de empresa en componentes reutilizables.

- *Clases de programación unificadas:* El entorno de trabajo ofrece a los desarrolladores un conjunto unificado, orientado a objetos, jerárquico y extensible de bibliotecas de clases (API). Con la creación de un conjunto de API comunes para todos los lenguajes de programación, Common Language Runtime permite la herencia, el control de errores y la depuración entre lenguajes. Todos los lenguajes de programación, pueden tener acceso al entorno de trabajo de forma parecida y los desarrolladores pueden elegir libremente el lenguaje que desean utilizar.
- *ASP.NET:* construye las clases de programación de .NET Framework, lo que proporciona un modelo de aplicación Web con un conjunto de controles e infraestructura que facilitan la generación de aplicaciones Web. ASP.NET incluye un conjunto de controles que encapsulan elementos comunes de interfaz de usuario de HTML, como cuadros de texto, botones y cuadros de lista. Sin embargo, dichos controles se ejecutan en el servidor Web, y representan la interfaz de usuario en el explorador como HTML. En el servidor, los controles exponen un modelo de programación orientado a objetos que proporciona la riqueza de la programación orientada a objetos al desarrollador Web. ASP.NET también proporciona servicios de infraestructura, como la administración de estado y el reciclaje de procesos, que reduce aún más la cantidad de código que debe escribir el desarrollador y aumenta la confiabilidad de la aplicación. Asimismo, ASP.NET utiliza estos mismos conceptos para permitir a los desarrolladores la entrega de software como un servicio. [Introducción a Visual Studio. Sin fecha]

2.16 Visual C#

Microsoft Visual C# 2005 es un lenguaje de programación diseñado para crear una amplia gama de aplicaciones que se ejecutan en .NET Framework. C# es simple, eficaz, con seguridad de tipos y orientado a objetos. Con sus diversas innovaciones, C# permite desarrollar aplicaciones rápidamente y mantiene la expresividad y elegancia de los lenguajes de tipo C.

Visual Studio admite Visual C# con un editor de código completo, plantillas de proyecto, diseñadores, asistentes para código, un depurador eficaz y fácil de usar, además de otras herramientas. La biblioteca de clases .NET Framework ofrece acceso a una amplia gama de servicios de sistema operativo y a otras clases útiles y adecuadamente diseñadas que aceleran el ciclo de desarrollo de manera significativa. [Visual C#. Sin fecha]

2.17 Bluetooth

Bluetooth es una especificación industrial para Redes Inalámbricas de Área Personal (WPANs) cuya tecnología fue desarrollada con el propósito de crear un estándar para interconectar computadoras, dispositivos de comunicación y accesorios a través de radios inalámbricos de bajo consumo de energía, corto alcance y económicos. Su nombre fue asignado en honor de Harald Blaatand II (940-981), un rey vikingo que unificó Dinamarca y Noruega y era conocido como Bluetooth, y del cual se dice que tenía muchas capacidades para ayudar a la gente a comunicarse. [Tanenbaum. 2003]

Esta tecnología fue desarrollada y estandarizada por el consorcio de empresas denominado el SIG (Special Interest Group) de Bluetooth, el cual esta formado por un grupo de compañías que se desenvuelven en el área de las telecomunicaciones, computación, redes y electrónica; posteriormente las especificaciones de las capas de nivel físico y de acceso al medio fueron

también formalizadas por la IEEE bajo el estándar IEEE 802.15.1. La versión más reciente del estándar según la SIG de Bluetooth es la 2.1, la cual fue publicada en Julio del 2007, mientras que la última versión de la IEEE fue publicada en el 2005.

La especificación de Bluetooth define un canal de comunicación de máximo 720 Kb/s (1 Mbps de capacidad bruta) con rango óptimo de 10 metros (opcionalmente 100 metros con repetidores). La frecuencia de radio con la que trabaja está en el rango de 2,4 GHz a 2,48 GHz con amplio espectro y saltos de frecuenta con posibilidad de transmitir en Full Duplex con un máximo de 1600 saltos/s. Los saltos de frecuencia se dan entre un total de 79 frecuencias con intervalos de 1 Mhz; esto permite dar seguridad y robustez. La potencia de salida para transmitir a una distancia máxima de 10 metros es de 0dBm (1 mW), mientras que la versión de largo alcance transmite entre 20 dBm y 30 dBm (entre 100 mW y 1 W). [Bluetooth. 14/08/2008]

2.18 Blue Soleil

Blue Soleil es una herramienta de software desarrollada por IVT Corporation la cual se encarga de gestionar una amplia variedad de servicios y conexiones de dispositivos Bluetooth con el ordenador a través de una interfaz de usuario que monitorea la actividad de los dispositivos y los servicios establecidos en tiempo real. [About Blue Soleil. 16/09/2008]

CAPÍTULO 3. MARCO METODOLÓGICO

La metodología seleccionada fue la del prototipado evolutivo, el cual es un modelo de desarrollo de software basado en creación de prototipos que son diseñados y posteriormente evaluados, conformando de esta manera una serie de iteraciones que permiten ir ajustando los requerimientos conforme van surgiendo y desarrollando el producto.

Esta metodología está compuesta por una serie de fases como (Ver figura 11):

- Concepto inicial.
- Diseño e implementación del prototipo inicial.
- Refinar el prototipo hasta que sea aceptado.
- Completar prototipo final.
- Entrega del producto final.

Figura 11. Fases del prototipado evolutivo
Fuente [Garzás. Sin Fecha]

Como el ambiente de interacción a desarrollar consiste en una combinación de componentes de software y de hardware, las etapas mencionadas anteriormente sufrieron una serie de cambios que permitieran el desarrollo de ambos, presentándose principalmente una reestructuración de la fase "refinar el prototipo hasta que sea aceptado", en subfases que

permitiera el diseño, construcción y evaluación de la parte lógica y física, dichos cambios se presentan a continuación (ver figura 12)

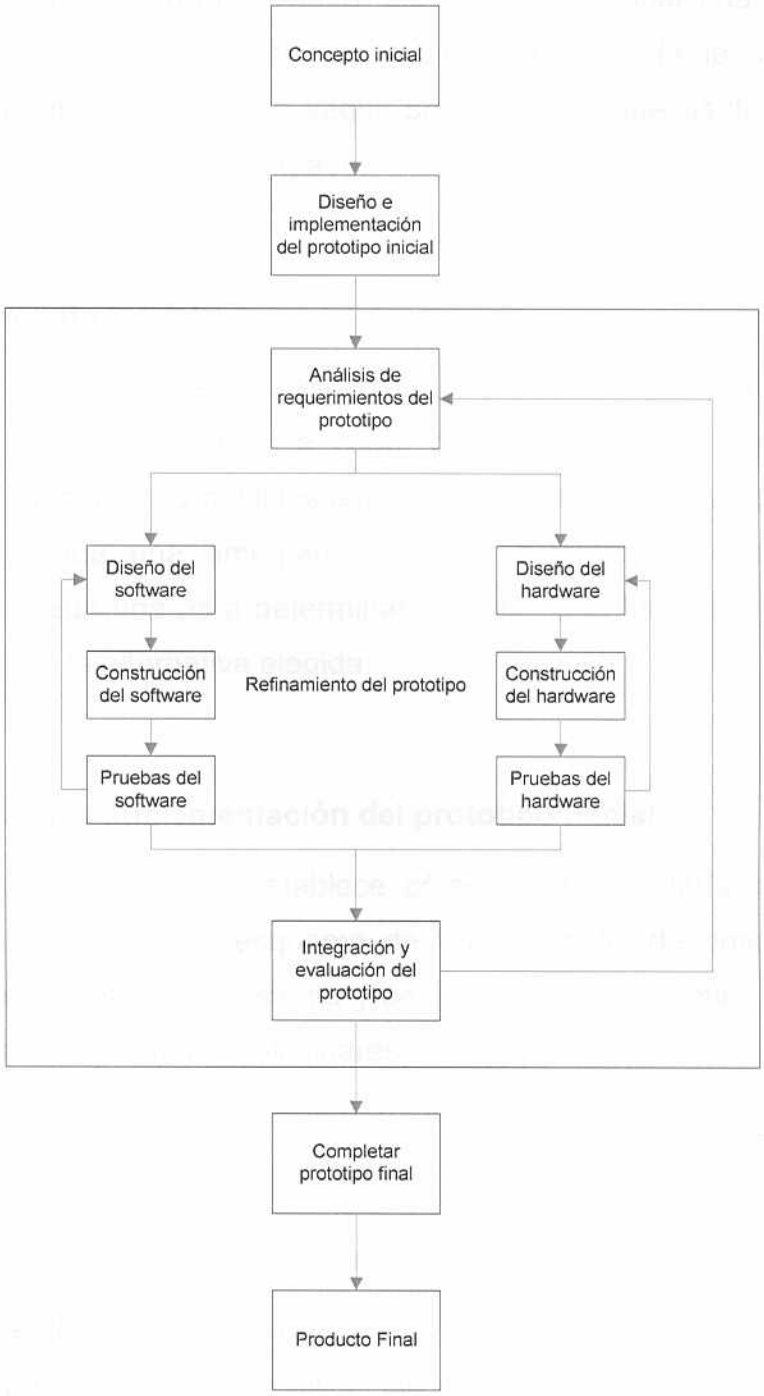


Figura 12. Adaptación de las fases del prototipado evolutivo
Fuente elaboración propia

La fases descritas anteriormente se encuentran constituidas por una serie de actividades que permitirán desarrollar los componentes de hardware y software que componen el proyecto, así como determinar cuáles son los resultados de cada fase, las cuales serán el insumo de la siguiente y permitirán determinar los pasos a seguir por medio de nuevas iteraciones a fin de cumplir con el objetivo planteado.

3.1 Concepto Inicial

El concepto inicial, permite establecer el producto a entregar, así como las primeras formas de llevarlo a cabo, por medio de una investigación documental se generan las posibles alternativas de solución y se evalúan los requisitos que cada una ameritan, a fin de establecer los alcances y limitaciones de cada una para determinar la factibilidad técnica y escoger la posible solución a la alternativa elegida.

3.2 Diseño e implementación del prototipo inicial

Durante esta fase, se establece cómo se desarrollaría la idea, se diseña la arquitectura o un esquema de trabajo, a fin de determinar los módulos que compondrán el sistema y se desarrolla un prototipo inicial que permita realizar las pruebas preliminares, a fin de definir los requerimientos de hardware y software.

3.3 Refinamiento del prototipo

El refinamiento del prototipo constituye la fase más importante de la metodología, porque permitirá realizar las iteraciones necesarias sobre los componentes de hardware y software hasta cumplir con el objetivo planteado, a continuación, se describen cada una de las subfases.

3.3.1 Análisis de los requerimientos del prototipo

Esta fase permite enfocar el prototipo base y los subsiguientes, hacia el objetivo planteado realizando las especificaciones a nivel de hardware y software.

3.3.2 Diseño del software

El diseño del software consiste en determinar cómo se llevará a cabo el software, cuáles son las herramientas que se utilizarán y la mejor manera de llevarlos a cabo.

3.3.3 Construcción del software

Consiste en el desarrollo del software en función de los requisitos establecidos y las herramientas seleccionadas.

3.3.4 Pruebas del software

Una vez completado el desarrollo del software, comienza una fase de verificación de la funcionalidad de la aplicación que permita enfocar el comportamiento hacia el logro de los objetivos, así como evaluar los resultados y determinar los ajustes necesarios.

3.3.5 Diseño del hardware

Se realiza un estudio de la información disponible, con la finalidad de definir las necesidades y prioridades para la construcción del hardware asimismo se evalúan las alternativas y se selecciona la solución técnica que cumpla con el objetivo planteado.

3.3.6 Construcción del hardware

En esta fase se lleva a cabo el proceso de construcción del hardware, en función del diseño realizado previamente.

3.3.7 Pruebas del hardware

Una vez completada la construcción del hardware, se realizan las pruebas pertinentes que permitan determinar el adecuado comportamiento y de ser necesario, replantear el diseño para realizar los ajustes pertinentes de acuerdo con los resultados obtenidos.

Las fases descritas anteriormente se desarrollan en paralelo por ser independientes entre sí y constituyen la parte fundamental del desarrollo del proyecto, ya que son las que permiten diseñar, construir y probar los elementos de software y hardware y así, en función de los productos obtenidos en cada una de ellas determinar los ajustes y las iteraciones siguientes.

3.3.8 Integración y evaluación del prototipo

Una vez alcanzada esta fase los elementos de hardware y software deben cumplir con el diseño establecido previamente con la finalidad de probar la interacción entre ambos, para obtener un resultado que determinen si es necesario realizar algún ajuste.

Luego de completada la integración entre ambos elementos se determina si es necesario realizar algún ajuste, dando a lugar a una nueva iteración desde la etapa de análisis.

3.4 Completar prototipo final

Una vez completada la fase de integración se procede a realizar los ajustes finales de forma que ambos componentes permitan tener el producto terminado y realizar las pruebas finales, para su posterior entrega.

3.5 Producto final

Completadas las fases descritas anteriormente el producto está listo para ser entregado y usado por el personal ajeno al desarrollo.

CAPÍTULO 4. DESARROLLO

4.1 Concepto Inicial

Habiendo definido que era necesario crear un ambiente de interacción hombre-ordenador en el cual una persona fuese capaz de controlar un ordenador con movimientos naturales de la mano se entró en un proceso de generación de ideas con el fin de que salieran a relucir distintas alternativas de solución tanto de hardware como de software para cumplir con los objetivos planteados.

Este proceso arrojó como resultado que este ambiente de interacción debía tener dos elementos esenciales: un dispositivo de hardware que fuese capaz de capturar los movimientos de la mano del usuario en las tres dimensiones y pudiese transmitir esta data a un ordenador; y un software que mostrara un ambiente virtual, representado por un escritorio tridimensional, en el que se interpretara la data para representar en el ambiente virtual los movimientos hechos por el usuario y en donde se pudiesen realizar algunas de las funcionalidades típicas de un escritorio.

Se evaluó la factibilidad técnica de los elementos involucrados en el proyecto. En primer lugar, para la detección de movimientos se determinó que el proceso de rastreo óptico era la opción más viable por lo que era necesaria una cámara infrarroja y algún dispositivo emisor de este tipo de luz de manera que la luz infrarroja emitida se reflejara en la mano del usuario y fuese captada por la cámara infrarroja. Para que el reflejo fuese mayor se determinó que se debían colocar unos accesorios reflectivos en la mano del usuario (Ver anexo A). Posteriormente, la data proveniente de la cámara infrarroja debía ser transmitida al ordenador para que ésta pudiese ser interpretada por el escritorio tridimensional.

La alternativa seleccionada como cámara infrarroja fue la que posee el Wii Remote de la empresa Nintendo, el cual es capaz de detectar varios puntos de éste tipo luz, es fácilmente adquirible y de relativo bajo costo. Adicionalmente este dispositivo posee la capacidad de comunicarse vía Bluetooth, lo que simplificaría las comunicaciones entre el dispositivo y el ordenador. En relación al dispositivo emisor de luz infrarroja se determinó que se debía diseñar y construir un circuito conformado por LEDs infrarrojos y para el reflejo de la luz se determinó que los accesorios debían ser construidos.

En relación con el software, ya habiendo definido que éste sería un escritorio con características visuales tridimensionales, se determinó que éste tendría un espacio limitado y que los elementos contenidos en él tendrían una apariencia similar a los elementos en los escritorios de los sistemas operativos más populares de la actualidad, aunque la calidad de las características visuales (definición de imágenes, sombreados, iluminación) sería limitada.

Para la creación del software se determinó que existen motores gráficos a través de los cuales se pueden desarrollar aplicaciones en donde se representan mundos virtuales tridimensionales los cuales se integran con gran cantidad de lenguajes de programación en donde se implementarían las funcionalidades del sistema.

Finalmente se definió que el sistema debía ser capaz de realizar en el escritorio tridimensional acciones típicas de este tipo de ambiente a través de la interpretación de la data proveniente del hardware emisor-receptor de infrarrojo; de manera que se produzca una interacción entre el usuario y el ordenador debido a que los movimientos de la mano del usuario manipularían el escritorio tridimensional.

4.2 Diseño e implementación del prototipo inicial

El diseño del circuito emisor de infrarrojo consistió en la colocación de varios LEDs cada uno con una resistencia para obtener un buen nivel de irradiación. Luego de diseñado se construyó el circuito emisor y se acopló con la cámara infrarroja. (Ver Apéndice A).

En relación al software se determinó que el prototipo inicial debía poseer un módulo de comunicaciones entre el hardware emisor-receptor de infrarrojo y el ordenador. Estas comunicaciones se realizarían vía Bluetooth utilizando en el ordenador el gestor de conexiones Blue Soleil. Luego de transmitidos los datos del receptor de infrarrojo (Wii Remote) al ordenador, éstos serían manipulados por las funciones de la librería WiimoteLib e interpretados por el software de manera que se representasen gráficamente los movimientos de la mano del usuario (Ver anexo B). Luego de diseñado el software del prototipo inicial se procedió a su desarrollo según los requerimientos planteados.

4.3.Refiniamiento del Prototipo

En esta etapa se realizó el refinamiento del prototipo inicial con el objetivo de obtener un producto final que cumpliese con las características planteadas. Este proceso de refinamiento consistió en la realización de tres iteraciones las cuales se describen a continuación.

4.3.1 Iteración I

A continuación se describen las actividades realizadas en cada una de las fases de la primera iteración.

4.3.1.1 Análisis de requerimientos del prototipo

El requerimiento principal en esta iteración fue el obtener un escritorio con los íconos 3D realizados y poder mover en él un cursor 3D una vez que el dispositivo de hardware haya sido optimizado y enfocado hacia su apariencia final. Específicamente los requerimientos de software y hardware fueron los siguientes (Ver tabla 1)

Requerimientos de software	Requerimientos de hardware
Diseño de modelos 3D semejantes a los íconos de los escritorios de los sistemas operativos de la actualidad.	Optimización del consumo eléctrico del emisor de infrarrojo.
Desarrollo de un ambiente virtual tridimensional.	Ampliación del campo visual del dispositivo receptor de infrarrojo.
Visualización de modelos 3D dentro del ambiente virtual	Ampliación del alcance y potencia del emisor infrarrojo.
Diseño de un cursor 3D usando colores llamativos.	Reducción de las dimensiones del hardware emisor-receptor de infrarrojo.
Implementación de movimiento del cursor en el ambiente virtual.	Construcción del accesorio reflector de infrarrojo a utilizar en el dedo índice.

Tabla 1. Requerimientos de software y hardware, iteración I
Fuente elaboración propia

4.3.1.2 Diseño del software

Una vez establecidos los requerimientos correspondientes, se procedió a diseñar un ambiente tridimensional que contuviese elementos 3D que representen los íconos de escritorio, así se diseñaron 11 modelos basados principalmente en los íconos que normalmente se observan en escritorios de sistemas operativos Windows, específicamente Windows Vista, estos serían: Internet Explorer, Mi PC, Windows Live Messenger, Papelera de Reciclaje, tres carpetas (vacía, con documentos y con sub-carpetas), CD ROM y tres íconos generales para Microsoft Word, Microsoft Excel y Microsoft Power Point. Para el diseño de los elementos 3D se seleccionó que una herramienta de modelado existente en el mercado, como 3Ds Max de

Autodesk en su versión 2009; así mismo con esta herramienta se diseñaría un modelo para el nuevo cursor 3D.

Para el desarrollo del ambiente virtual tridimensional que incluiría los modelos 3D se optó por el uso del API DirectX de Microsoft por su integración con el lenguaje C# y por la existencia de gran cantidad de documentación. Se estableció que en principio el ambiente tridimensional constaría de un espacio negro aparentemente infinito en el que se desplegasen los elementos 3D y el cursor se moviera de la misma manera en que se implementó en el prototipo inicial.

Luego de un proceso de investigación se llegó a la conclusión que para mostrar los modelos 3D en el ambiente virtual estos debían ser cargados con un formato específico desde un archivo extensión .X, para ello el 3Ds Max debía ser capaz de exportar el modelo diseñado a un archivo con dicha extensión, pero la misma no se encontraba disponible, por lo que se buscó un exportador externo para tal tarea, como el Panda DirectX. Igualmente se dio un proceso de investigación para hallar la forma adecuada de cargar un archivo .X a una aplicación DirectX sobre C#.

4.3.1.3 Construcción del software

Para el diseño de los íconos se buscaron sus correspondientes modelos en 2D dentro del sistema operativo Windows Vista (Ver Figura 13) para su posterior modelado en la herramienta 3Ds Max 2009.

Figura 13. Íconos 2D usados como modelo para el modelado 3D
Fuente elaboración propia

Luego se realizó el proceso de modelado 3D usando la variada gama de funcionalidades ofrecidas por 3Ds Max (Ver Figura 14) y tratando en lo posible de acercarse al diseño de los modelos 2D originales, así mismo se modelaron aquellos íconos con los que no se contaba con un modelo a seguir, incluyendo un diseño 3D para el cursor (Ver figura 15); la aplicación 3Ds Max no sólo sirvió para modelar, como es lo usual, sino que también se usó para aplicarle a los modelos materiales, como los colores.

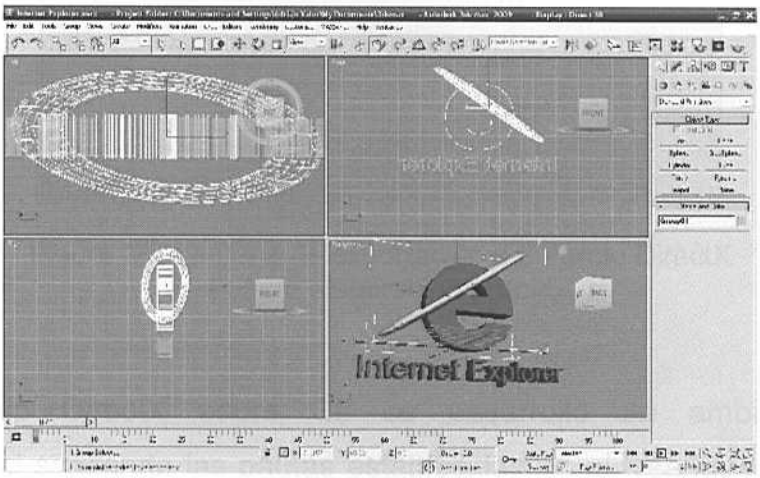


Figura 14. Diseño de uno de los íconos con la aplicación 3Ds Max 2009
Fuente elaboración propia

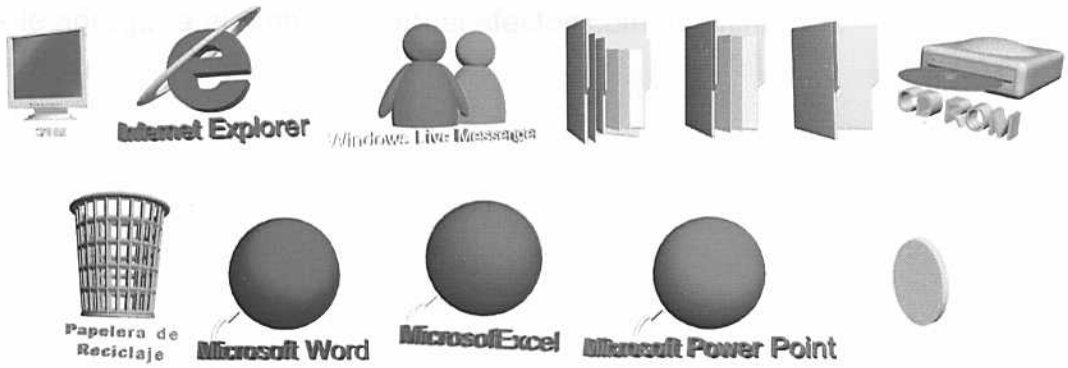


Figura 15. Modelos 3D de los íconos del escritorio y del cursor
Fuente elaboración propia

Al culminar este proceso se exportaron todos los modelos uno por uno a formato .X usando el plugin Panda DirectX para el 3Ds Max 2009 (Ver Figura 16).

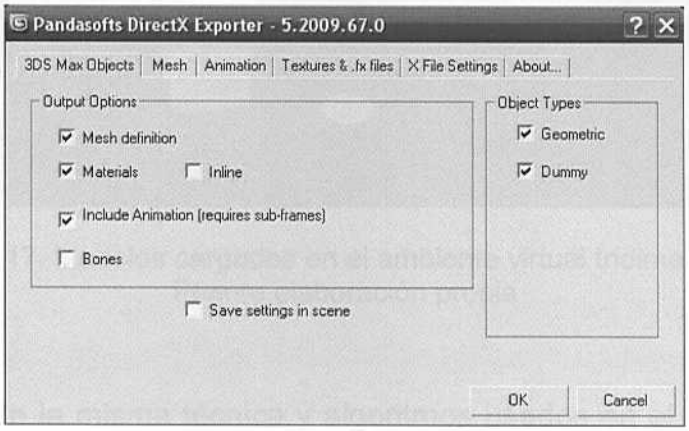


Figura 16. Ventana de exportación del Panda DirectX
Fuente elaboración propia

Usando DirectX sobre C# se desarrolló un ambiente virtual tridimensional siguiendo las pautas establecidas en el diseño, sobre éste se cargaron los modelos en formato .X usando los algoritmos adecuados previamente investigados en la etapa de diseño, llamándose ahora estos modelos en DirectX bajo el nombre de "mesh" (Ver Figura 17), luego también se le agregaría al ambiente virtual efectos como iluminación.

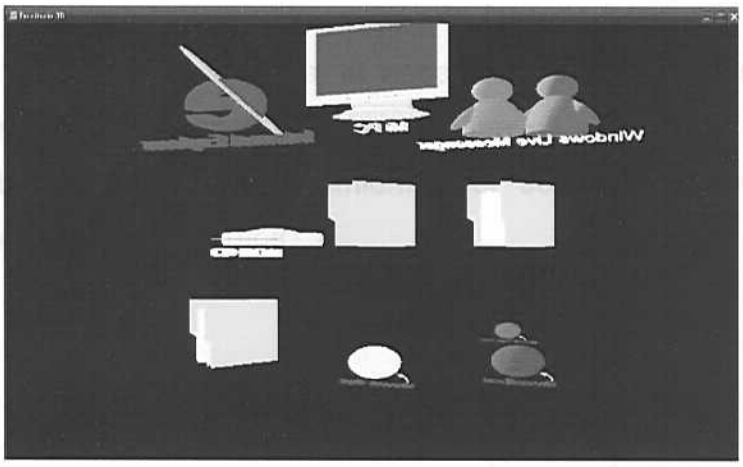


Figura 17. Modelos cargados en el ambiente virtual tridimensional
Fuente elaboración propia

Siguiendo la misma técnica y algoritmos usados en el prototipo inicial y, usando el nuevo cursor 3D de un fuerte color amarillo, se logró moverlo en el ambiente virtual en conjunto los íconos ya cargados

4.3.1.4 Pruebas del software

Las pruebas en esta iteración se basaron en comprobar la correcta visualización de los elementos 3D en el ambiente virtual, lo cual no ocurrió hasta después de varias iteraciones internas del software, observándose fallas en la visualización, como mostrarse los elementos sin color, invertidos o hasta no mostrarse o cargarse; estas situaciones provocadas principalmente a la poca destreza que se tenían con las nuevas herramientas utilizadas.

Evaluando bien los resultados obtenidos en las pruebas se localizaron los orígenes de las fallas y se depuraron las técnicas usadas en cada etapa, revisándose también otras documentaciones y probando métodos alternos previamente localizados en la etapa de diseño, para ello teniendo que regresar a esta etapa.

Al final las pruebas antes de pasar a la fase de integración arrojaron, por lo menos a nivel de apariencia, la necesidad de incluir en el ambiente virtual unos límites, como unas paredes, ya que se visualizaba a futuro un potencial problema en el funcionamiento del software si se dejaba el movimiento libre de los elementos.

4.3.1.5 Diseño del hardware

Una vez determinados los factores a mejorar se procedió a rediseñar el dispositivo emisor de infrarrojo. Para disminuir el consumo eléctrico del arreglo de LEDs infrarrojos se determinó que se debían colocar los LEDs infrarrojos en serie, de manera que la corriente que circulara por ellos fuese la mayor posible para que cada LED trabajara al máximo de su potencia, pero debido a estar colocados en serie, disminuyera su consumo eléctrico. (Ver Apéndice B)

Para que el dispositivo receptor de infrarrojo tuviese una mayor área de visualización se determinó que se debía colocar un mayor número de LEDs en el emisor de infrarrojo, de esta manera se aumentaría el área de irradiación del emisor y por consiguiente se aumentaría el área de visualización del receptor.

Como fase final del diseño del dispositivo emisor de infrarrojo se procedió a configurar la posición de cada grupo de LEDs en la baquelita universal de manera que se simplificara al máximo su construcción y se disminuyeran al máximo las dimensiones del dispositivo emisor debido a que éste sería encapsulado en un cajetín de dimensiones limitadas.

En relación a los accesorios reflectores de infrarrojo se determinó que el que se colocaría en el dedo índice del usuario debía ser diseñado en forma de anillo y construido con un material elástico de manera que éste lograra

ajustarse lo más posible a las dimensiones del dedo índice de cualquier usuario.

4.3.1.6 Construcción del hardware

Para la construcción del hardware en esta iteración modificó una baquelita universal, que se ajustara a las necesidades del emisor infrarrojo. (Ver Figura 18)

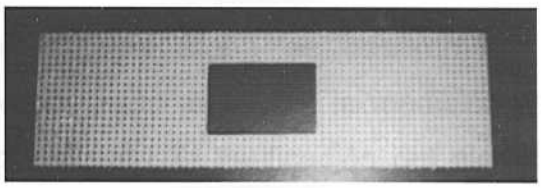


Figura 18. Baquelita para el ensamblaje del circuito emisor de infrarrojo.
Fuente elaboración propia

Posteriormente se construyó el circuito y a colocarlo en el cajetín en conjunto con el dispositivo receptor de infrarrojo (Wii Remote). (Ver Figura 19).

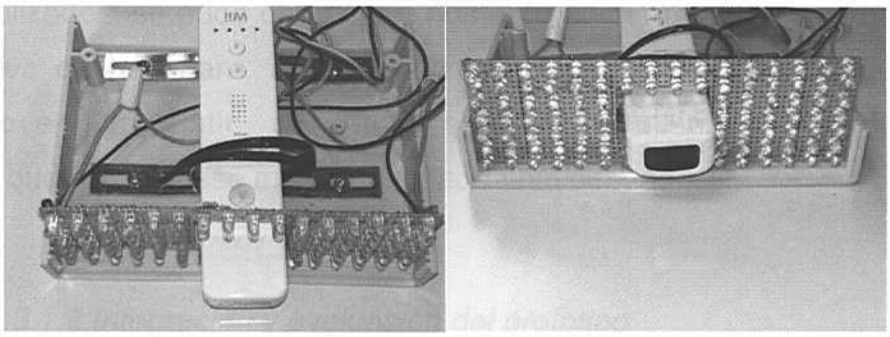


Figura 19. Hardware emisor-receptor de infrarrojo, iteración I
Fuente elaboración propia

De manera paralela se construyó el accesorio reflector de infrarrojo según los lineamientos planteados en su diseño. (Ver Figura 20).

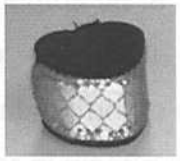


Figura 20. Accesorio reflector de infrarrojo, iteración I
Fuente elaboración propia

4.3.1.7 Pruebas del hardware

Las pruebas en esta etapa se basaron en la verificación del funcionamiento de cada grupo de LEDs, la medición del consumo total de corriente eléctrica del circuito, la verificación del alcance y la amplitud del área de la luz emitida por el circuito emisor de infrarrojo; adicionalmente se verificó el buen desempeño del accesorio reflector construido en esta iteración.

Con este diseño se obtuvo un menor consumo eléctrico y se aumentó la potencia de luz emitida por el dispositivo emisor de infrarrojo, lo que se traduce en el aumento del alcance de la luz infrarroja y, debido al aumento de LEDs utilizados, se pudo obtener un aumento en el área de visualización del dispositivo de hardware. Al ensamblar el circuito emisor y el receptor de infrarrojo en un cajetín, se obtuvo el encapsulamiento del hardware de manera que se acercase más al resultado final deseado.

4.3.1.8 Integración y evaluación del prototipo

Ambos productos obtenidos, de software y de hardware fueron integrados exitosamente, el objetivo principal de la iteración se cumplió al poder mover en el ambiente virtual un cursor 3D, los movimientos fueron fluidos y precisos.

En relación con el accesorio reflector de infrarrojos se determinó que el área reflectiva no era la suficientemente amplia como para darle al usuario la mayor flexibilidad en los movimientos de la mano al momento de manipular el escritorio tridimensional, esto debido a que la zona de reflexión del accesorio se perdía con facilidad al momento de realizar ciertos movimientos; por tal motivo se decidió replantear el diseño del accesorio.

4.3.2 Iteración II

A continuación se describen las actividades realizadas en cada una de las fases de la segunda iteración.

4.3.2.1 Análisis de requerimientos del prototipo

El requerimiento principal en esta iteración fue el lograr mover el cursor en el eje Z del escritorio tridimensional diseñando para ello un nuevo elemento de hardware necesario para tal fin. Los requerimientos específicos de software y hardware fueron los siguientes (Ver tabla 2)

Requerimientos de software	Requerimientos de hardware
Diseño e inclusión de límites en el ambiente virtual con el fin de crear un área de trabajo.	Construcción de un regulador de voltaje de 12 v DC a 3 v DC para alimentar al dispositivo receptor de infrarrojo (Wii Remote).
Desarrollo de una pantalla de configuración antes de inicializar el escritorio tridimensional.	Rediseño y construcción del accesorio reflectivo del dedo índice del usuario.
Implementación del movimiento del cursor en la dirección Z para el efecto de profundidad.	Diseño y construcción del accesorio reflectivo con el que se obtiene la distancia de la mano al hardware emisor-receptor de infrarrojo.

Tabla 2. Requerimientos de software y hardware, iteración II
Fuente elaboración propia

4.3.2.2 Diseño Software

El diseño en esta iteración se enfocó principalmente a conseguir en conjunto con el hardware, el movimiento en profundidad de los elementos mostrados en el escritorio, para ello se tendría que modificar algunos aspectos en el código hecho en el prototipo inicial para el movimiento del cursor para así considerar la data del eje Z. Dado que la librería WiimoteLib no contiene ningún método que devuelva la posición en Z o la distancia del Wii Remote al origen de luz infrarroja, se tuvo que calcular ésta por medio de una triangulación, usando dos puntos más de infrarrojo y fijos entre sí, estos agregados al hardware fueron contemplados ya en los requerimientos de la presente iteración. (Ver anexo C)

Para la inclusión de límites en el ambiente virtual se estableció el diseño de un paralelepípedo, específicamente un ortoedro (ver Figura 21), para limitar el área de trabajo del escritorio con seis planos, para ello se tendría que recurrir nuevamente al 3Ds Max para crear un nuevo modelo, y su apariencia tendría que ser de un color neutro, para que así resaltasen los íconos del escritorio y el cursor, y no se distrajera la vista del usuario con el fondo, así pues se escogió una textura que se ajustaba a las necesidades señaladas.

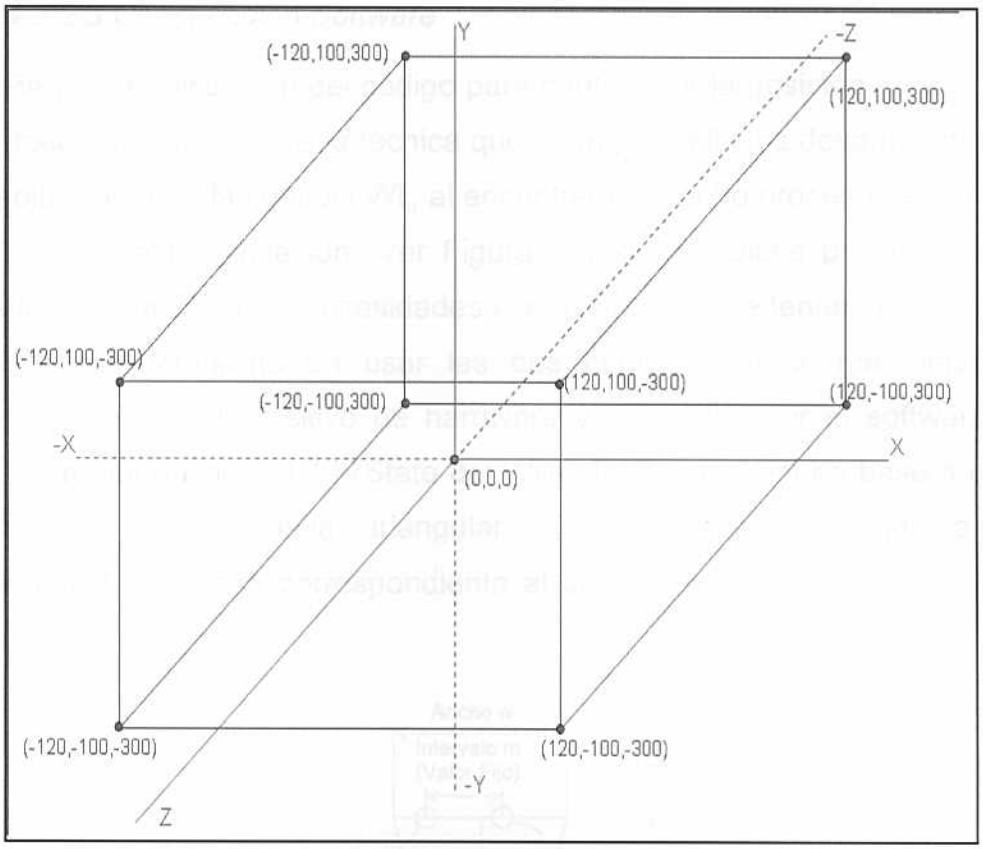


Figura 21. Paralelepípedo, específicamente ortoedro
Fuente elaboración propia

Para el desarrollo de la ventana de configuración se definió debía ser desarrollada usando netamente componentes de C# y, por las funciones que se realizarían en ella no debería ser muy grande; las funciones que inicialmente se implementarían sería la de cambiar la resolución de la pantalla, permitir ingresar antes de iniciar el ambiente virtual unos valores para el uso interno de los desarrolladores para las pruebas y, lo más importante es que ésta debía realizar una prueba de conexión con el dispositivo de hardware para detectar si la comunicación Bluetooth está establecida correctamente.

4.3.2.3 Construcción Software

Para la modificación del código para contemplar la posición en el Z se buscó documentación sobre la técnica que usan para ello los desarrolladores de videojuegos para la cónsola Wii, al encontrar que dicho proceso se realiza a través de una triangulación (Ver Figura 22) se usó dicho principio para adaptarlo de acuerdo a las necesidades específicas que se tenían, por lo que la modificación consistió en usar las dos coordenadas X más lejanas encontradas por el dispositivo de hardware y obtenidas por el software a través del método RawX del IRState del WiimoteLib, para así en base a una distancia fija de referencia, triangular y aplicar dicho resultado a la coordenada Z del mesh correspondiente al cursor para así moverlo en los tres ejes.

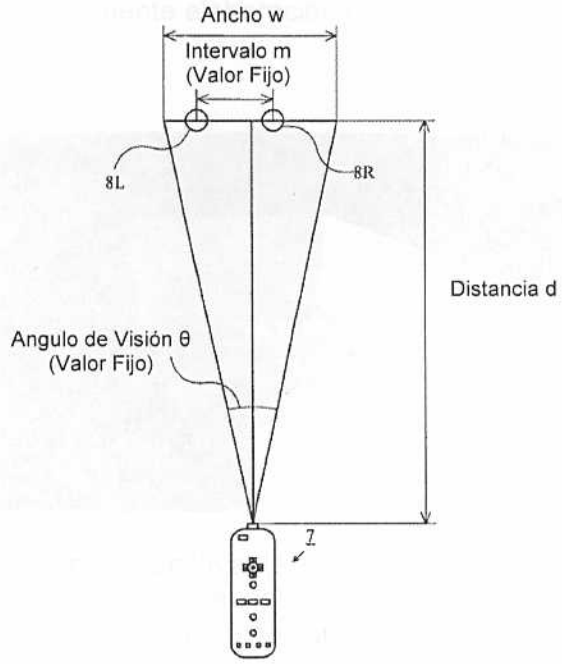


Figura 22. Técnica para obtener la distancia del wiimote a los puntos infrarrojos
Fuente [Nintendo patent application 2007/0060384. 13/09/2007]

Para la elaboración de los límites en el ambiente tridimensional se diseñó de la misma manera en que se habían hecho los íconos, un modelo

en 3Ds Max con la forma, tamaño y textura previamente establecida en la fase de diseño, se exportó usando Panda DirectX y se cargó con el algoritmo adecuado al ambiente tridimensional (Ver Figura 23 y 24).

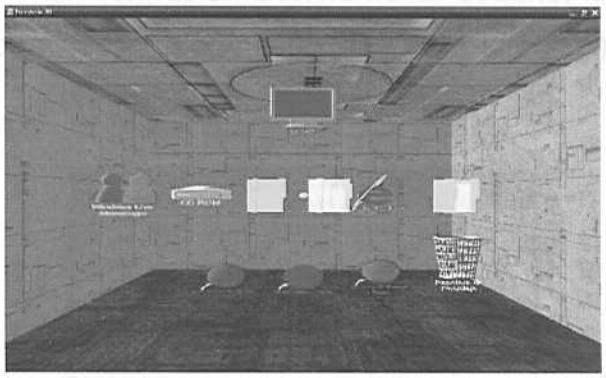


Figura 23. Modelo cargado para establecer los límites del escritorio
Fuente elaboración propia

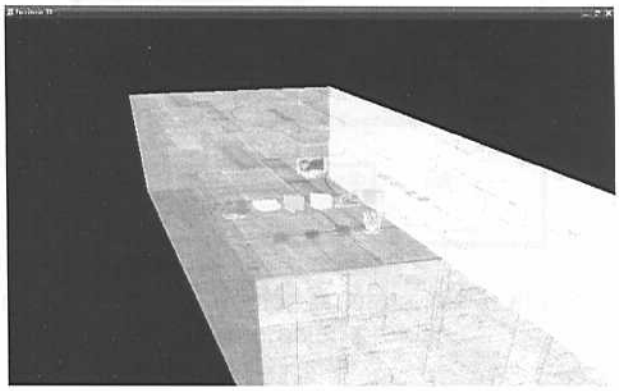


Figura 24. Vista externa y en perspectiva del escritorio tridimensional limitado por el modelo.
Fuente elaboración propia

Para el desarrollo de la ventana de configuración se usaron los componentes proporcionados por el lenguaje C# y la plataforma .NET para diseñar una ventana sencilla que cumpliera con las necesidades especificadas en los requerimientos y basado en el diseño previamente

definido, constando con tres campos para la modificación de los valores de sensibilidad usados internamente por los desarrolladores para las pruebas, un lista de selección para la elección de resolución de pantalla, una opción para establecer si se ejecutaría el escritorio 3D en pantalla completa, un espacio para mostrar la información pertinente al usuario, un hipervínculo para salir y el botón para iniciar el escritorio 3D. (Ver Figura 25)

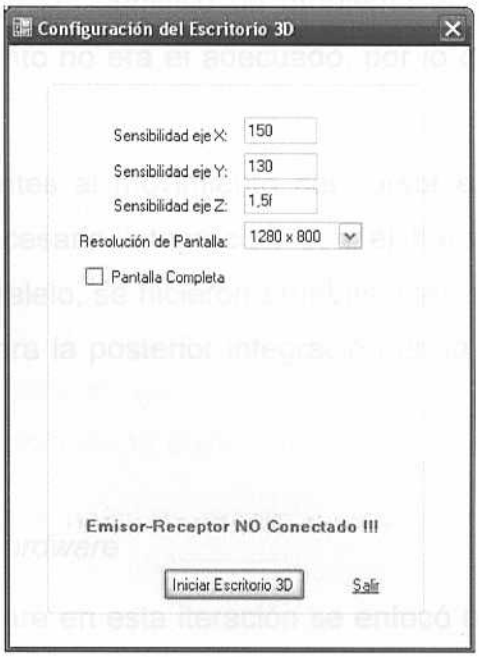


Figura 25. Ventana de configuración previa a iniciar el escritorio 3D
Fuente elaboración propia

4.3.2.4 Pruebas Software

Las pruebas fueron realizadas satisfactoriamente y consistieron principalmente en lo referente a la integración del nuevo modelo creado para limitar el escritorio y su influencia sobre la iluminación del ambiente y otros aspectos visuales como el posicionamiento adecuado en relación con las posiciones iniciales de los íconos y lo referente al tamaño del modelo, es decir, si la proporción del tamaño del modelo y los íconos guardaban

relación; esto requirió varios ajustes y mejoras que dieron paso a iteraciones internas de software.

También se realizaron pruebas de comunicación con el wiimote respecto a la ventana de configuración, resultando exitosas, el sistema logró reconocer de manera precisa si el dispositivo estaba conectado o no por Bluetooth y lo indicaba tal como fue programado en un mensaje amigable para el usuario, permitiendo así la iniciación correcta del escritorio tridimensional, sin embargo se identificó un problema al activar la pantalla completa, su comportamiento no era el adecuado, por lo que en posteriores iteraciones se excluiría.

Las pruebas referentes al movimiento del cursor en el eje Z fueron limitadas, debido a la necesaria integración con el hardware que estaba siendo desarrollado en paralelo, se hicieron pruebas internas de cálculo y se depuraron los procesos para la posterior integración en la siguiente fase de desarrollo.

4.3.2.5 Diseño del hardware

El diseño del hardware en esta iteración se enfocó en el diseño de un circuito regulador de voltaje (de 12v DC a 3v DC) para alimentar al dispositivo receptor de infrarrojo (Ver Apéndice C) de manera que la alimentación eléctrica de este dispositivo dependiese de la misma fuente de poder con la que se alimenta al dispositivo emisor y no de algún tipo de batería o componente ajeno al hardware.

En relación con los accesorios reflectores de infrarrojo se rediseñó el que sería colocado en el dedo índice del usuario con el objetivo de que tuviese una mayor área de reflexión y que ésta no se perdiese con facilidad. En tal sentido se diseñó el dispositivo con forma de dedal y se mantuvo la

idea de construirlo sobre un material elástico para que se ajustase a las dimensiones de la mano de cualquier usuario.

En esta iteración también se procedió a diseñar el accesorio que se encargaría de reflejar dos puntos de referencia de la mano del usuario a través de los cuales se obtendría la data relacionada con la ubicación en el eje Z (profundidad) de la mano. Se determinó que el accesorio debía ser lo más ligero y cómodo posible, y que no entorpeciera los movimientos del usuario; conjuntamente debía mantener su nivel de reflexión sin importar los movimientos realizados por el usuario.

4.3.2.6 Construcción del hardware

Para la construcción del hardware adaptó una baquelita universal de manera que se ajustara a las necesidades del circuito a construir, posteriormente se posicionaron todos los componentes y finalmente se construyó el circuito regulador de 12 voltios a 3 voltios.

Luego de construir el circuito regulador, se procedió a integrarlo y conectarlo en el cajetín del hardware emisor-receptor. (Ver Figura 26).

Figura 26. Hardware emisor-receptor de infrarrojo, Iteración II.
Fuente elaboración propia

De manera paralela se construyeron los accesorios reflectores de infrarrojo, inicialmente se realizó la construcción del que sería colocado en el dedo índice del usuario, y posteriormente el que serviría para obtener la distancia de la mano al dispositivo emisor receptor. (Ver Figura 27).

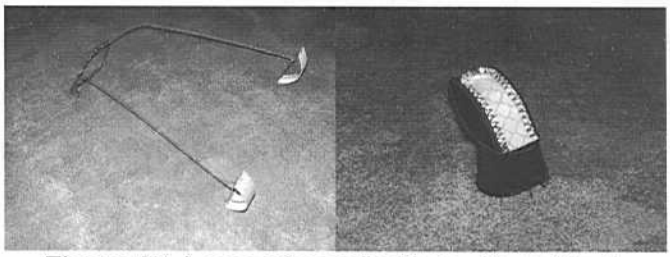


Figura 27.Accesorios reflectivos, Iteración II.
Fuente elaboración propia

4.3.2.7 Pruebas del hardware

En esta etapa se realizaron las pruebas relacionadas con el funcionamiento del regulador de voltaje; al verificar el voltaje de salida del mismo se determinó que no era el ideal para el funcionamiento normal del dispositivo receptor de infrarrojo por lo que se debía realizar un rediseño y modificación del circuito. Tal situación produjo que se realizara una nueva iteración de hardware según el modelo metodológico utilizado para el desarrollo del proyecto; esta nueva iteración se realizó y se obtuvo como resultado el funcionamiento ideal del regulador de voltaje; finalmente se realizaron las pruebas integrales de hardware en donde también se verificó el buen desempeño de los accesorios reflectivos. Estas pruebas arrojaron resultados satisfactorios.

4.3.2.8 Integración y evaluación del prototipo

La integración de realizó de manera exitosa, observándose que el nuevo accesorio usado en los dedos del usuario cubrió satisfactoriamente las

necesidades para las cuales fueron construidos, notándose una mejoría en la detección por parte del software de los puntos reflectivos ubicados en el dedo del usuario. Por otra parte, el accesorio con el que se captura la profundidad de la mano se integró con el software obteniendo resultados en general positivos, sin embargo, aunque el diseño resultó acertado, se notó un problema relativo a la estabilidad debido a que este presentaba una vibración que afectaba negativamente la precisión de los movimientos en el eje Z por lo que se planteó la necesidad de reconstruir dicho accesorio utilizando materiales más sólidos y proporcionen mayor estabilidad.

4.3.3 Iteración III

A continuación se describen las actividades realizadas en cada una de las fases de la tercera iteración.

4.3.3.1 Análisis de requerimientos del prototipo

El requerimiento principal en esta iteración fue el obtener un escritorio tridimensional con todas las funcionalidades implementadas. (Ver tabla 3)

Requerimientos de software	Requerimientos de hardware
Rediseño de la ventana de configuración con auto ajuste de sensibilidad y establecimiento de distancia inicial.	Colocación de elementos estéticos y de usabilidad en el hardware emisor-receptor de infrarrojo.
Implantación de las acciones a realizarse sobre los íconos del escritorio	Reconstrucción del accesorio reflectivo con el que se obtiene la distancia de la mano del usuario del sistema.
Cambio de diseño del cursor 3D	
Desarrollo de funcionalidad para cambiar apariencia al escritorio.	

Tabla 3. Requerimientos de software y hardware, iteración III
Fuente elaboración propia

4.3.3.2 Diseño del software

El diseño de esta iteración se enfocó en evaluar las funcionalidades a ser implementadas en el escritorio tridimensional, definiéndose que éstas serían aquellas básicas que se realizan normalmente para realizar las tareas en los escritorios de los sistemas operativos que existen en la actualidad, así se escogieron las acciones de: hacer doble clic en ícono, seleccionar ícono (ponerlo en foco), mover ícono y eliminar ícono; todas estas acciones tendrían que ser disparadas por los movimientos de la mano del usuario, por lo que se estableció un movimiento específico a cada acción, éstos tendrían que ajustarse a las limitaciones lógicas asociadas a la programación de los eventos en el software, escogiéndose los más idóneos (Ver Apéndice D).

El rediseño de la ventana de configuración vendría dado visualmente por la eliminación de los campos de sensibilidad X, Y y Z, ya que uno de los requerimientos de esta iteración fue el que dichos valores se ajustaran automáticamente, la solución a dicho problema se daría casi automáticamente al resolver el requerimiento de establecer una distancia inicial de posicionamiento en el eje Z, esto era necesario para que el usuario pudiera iniciar el sistema en una posición cómoda de su brazo y el comportamiento del sistema sea el adecuado, para ello el sistema tendría que guardar los valores iniciales de los dos puntos reflectivos del accesorio para determinar la profundidad de la mano, y con estos valores obtener los valores de sensibilidad.

Para el cambio de diseño del cursor 3D se recurriría de nuevo a la aplicación 3Ds Max, la figura a modelar sería un cubo inclinado para mejorar el efecto tridimensional, el color seleccionado sería el mismo que tenía el cursor anterior, es decir, amarillo intenso.

Para el cambio de apariencia del escritorio se buscarían distintas fotos que pudiesen ajustarse correctamente como texturas al modelo y que tuvieran una apariencia agradable.

4.3.3.3 Construcción del software

Para la implementación de las funcionalidades en los íconos del escritorio se desarrollaron las funciones en la programación del sistema que permitieran ejecutar esos eventos tal como se habían establecido en la etapa de diseño; para el caso de la eliminación de un ícono tuvo que adicionalmente diseñarse un modelo en 3Ds Max para la ventana de confirmación de eliminación del ícono. (Ver Figura 28)

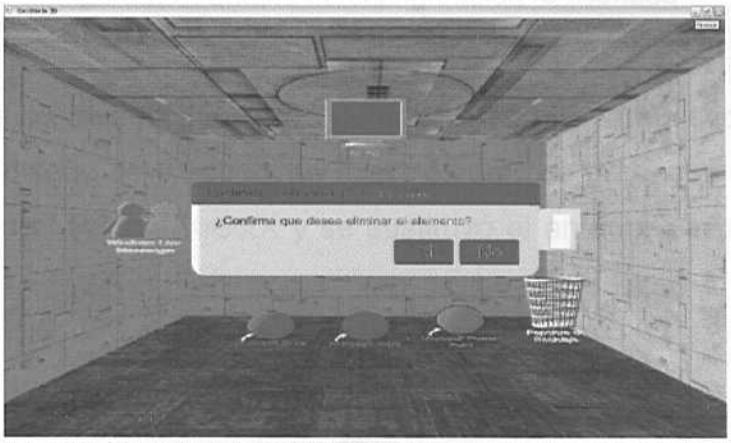


Figura 28. Ventana de confirmación de eliminación de ícono.
Fuente elaboración propia

El rediseño de la ventana de configuración consistió en la eliminación de los campos de sensibilidad, disminuyendo el tamaño de la ventana, igualmente se excluyó la opción de pantalla completa y se agregó un botón para iniciar la calibración del software para trabajar correctamente con el hardware (Ver Figura 29). En el código de programación fueron agregadas las funciones que usen la información proporcionada por el WiimoteLib para la definición de la distancia inicial de la mano del usuario y con ello el cálculo automático de la sensibilidad de los ejes X, Y y Z, basándose en el proceso definido en la fase de diseño.

Figura 29.Ventana de configuración rediseñada
Fuente elaboración propia

Para el modelado del nuevo cursor 3D se uso la aplicación 3Ds Max para crear el diseño de un cubo, se exportó como se ha hecho en las etapas anteriores con el Panda DirectX y se sustituyó el anterior por el nuevo (Ver Figura 30).

Figura 30. Nuevo cursor del escritorio tridimensional
Fuente elaboración propia

Para el cambio de apariencia del escritorio tridimensional se buscaron una serie de texturas adecuadas para ser usadas como se definió en la fase de diseño (Ver Figura 31).

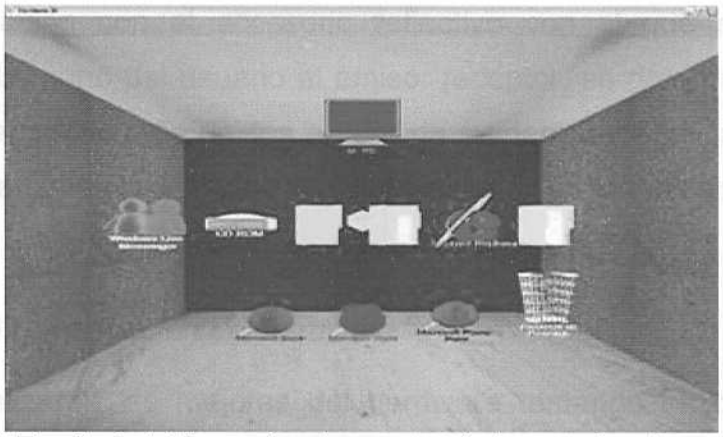


Figura 31. Escritorio tridimensional con una de las apariencias alternativas
Fuente elaboración propia

4.3.3 4 Pruebas del software

En esta etapa las pruebas se enfocaron principalmente al aspecto visual del escritorio, debido a que las referentes a los eventos de los íconos y la pantalla de configuración se realizarían en la fase de integración con el hardware. Por lo que se observó el nuevo cursor fue sustituido y visualizado exitosamente al igual que las nuevas texturas aplicadas al escritorio también, notando que sólo lucían bien aquellas que tuvieran un aspecto homogéneo.

4.3.3.5 Diseño del hardware

Como iteración final en el desarrollo del hardware se diseñaron los elementos estéticos que le dieran mayor usabilidad al hardware resultado de las iteraciones anteriores; los aspectos a mejorar fueron la inclusión de un interruptor de encendido/apagado del hardware, un mecanismo para conectar y desconectar la fuente de poder de manera práctica, un pulsador para la sincronización de las comunicaciones Bluetooth entre el hardware y el ordenador y, una lámina acrílica para proteger físicamente los LEDs emisores de infrarrojo.

En relación con el accesorio reflectivo con el que se obtiene la distancia de la mano del usuario al emisor-receptor, se determinaron cuales serían los materiales más rígidos a usar de manera de evitar al máximo las vibraciones causantes de fallas en la interpretación de la distancia.

4.3.3.6 Construcción del hardware

Se realizaron las mejoras del hardware tomando en cuenta que el resultado que se deseaba obtener debía ser el más parecido al que podría ofrecer cualquier hardware comercial, en tal sentido, se incorporaron los elementos necesarios para hacer que el hardware diseñado y construido no sólo fuese funcional sino también fácil de usar y estéticamente agradable.(Ver Figura 32). El detalle de la construcción del dispositivo emisor – receptor de infrarrojo de la tercera iteración se puede apreciar en el Apéndice E.

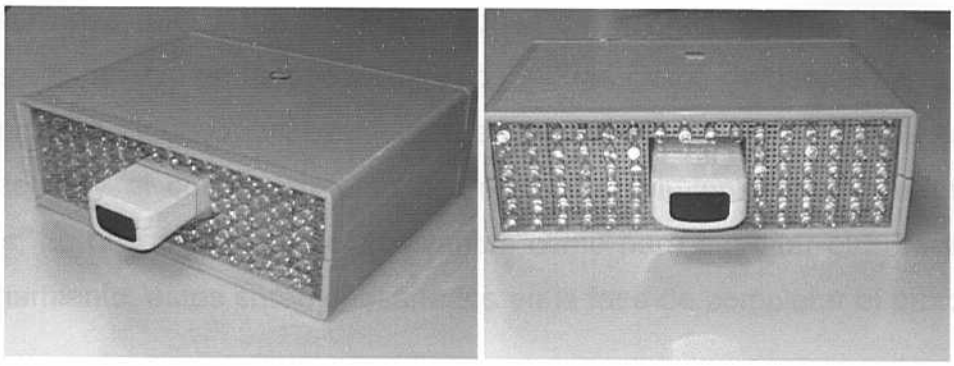


Figura 32. Hardware emisor-receptor de infrarrojo obtenido con la tercera iteración. Fuente elaboración propia

Finalmente se construyó con materiales más rígidos el accesorio que sirve para obtener dos puntos de referencia de la mano del usuario.

4.3.3.7 Pruebas del hardware

Se realizaron las pruebas del funcionamiento del hardware emisor-receptor de infrarrojo, las cuales consistieron, en la verificación del funcionamiento del interruptor de encendido/apagado y la verificación del pulsador de sincronización de comunicaciones Bluetooth entre el hardware y el ordenador. También se verificó el funcionamiento del accesorio construido con materiales más rígidos. Todas las pruebas arrojaron resultados satisfactorios.

4.3.3.8 Integración y evaluación del prototipo

La integración se realizó de manera exitosa, observándose una notable mejoría en la estabilidad del accesorio que detecta la profundidad de la mano, que influyó positivamente en el comportamiento del cursor en el eje Z, aumentando la precisión en los movimientos. Por otro lado, en lo referente a las acciones realizadas sobre los íconos del escritorio, se efectuaron tal como se estableció en la fase de diseño y en los casos de uso, sin embargo se detectaron algunos inconvenientes en lo referido al tiempo de respuesta del sistema y al comportamiento de algunos movimientos para iniciar los eventos, no obstante al no ser relevantes como para ameritar otra iteración de refinamiento, éstos serán subsanados en la fase de completar el prototipo final.

4.4 Completar prototipo final

Una vez finalizada la fase de integración se realizaron las pruebas y los ajustes finales en el sistema. En tal sentido se cambió el color del cursor del escritorio tridimensional para que fuese más representativo en el ambiente. También se modificó la iluminación del ambiente de manera que

se produjeran efectos más agradables para el usuario. Por otro lado se realizaron ajustes en la precisión y la respuesta del sistema a las acciones realizadas por el usuario, de manera que éstas se adecuaran a los movimientos naturales de la mano.

4.5 Producto final

Luego de la implementación de cada una de las fases de la metodología de prototipado evolutivo se obtuvo el producto final el cual cumple con cada uno de los requerimientos inicialmente planteados.

CAPÍTULO 5. RESULTADOS

Los resultados obtenidos con la realización de este Trabajo Especial de Grado fueron los siguientes:

- Se evaluaron las alternativas tecnológicas existentes para el desarrollo del proyecto y se obtuvo que el motor gráfico DirectX se ajustaba a las necesidades del proyecto para la creación del ambiente virtual tridimensional debido a la gran cantidad de documentación disponible. Conjuntamente se determinó que el lenguaje C# era el mas adecuado para el desarrollo del software debido a su simplicidad y compatibilidad con DirectX dado a que pertenecen a la misma casa matriz.
- Se implementó la metodología de desarrollo de prototipado evolutivo a través de la cual se perfeccionó el prototipo inicial hasta que éste cumpliese con los requerimientos funcionales y no funcionales establecidos inicialmente.
- Se desarrolló un ambiente virtual representado por un escritorio 3D que contiene modelos tridimensionales que se asemejan a los íconos de los escritorios de los sistemas operativos existentes.
- Se diseñó, construyó y adaptó un dispositivo de hardware emisor-receptor de infrarrojo y unos accesorios reflectivos con los cuales se pueden capturar los movimientos de la mano y transmitirlos al computador vía Bluetooth.
- Se creó un ambiente de interacción hombre-ordenador integrando el software desarrollado y los dispositivos de hardware construidos de manera que el usuario puede ver reflejados los movimientos de su mano en el ambiente virtual y realizar diversas acciones obteniendo una retroalimentación a través de sonidos y movimientos.

- Se implementaron una serie de funcionalidades básicas de un escritorio para interactuar con los elementos que éste contiene. Las funcionalidades implementadas fueron: mover cursor, seleccionar ícono, hacer clic en ícono, eliminar ícono y mover ícono, y con algunas de estas acciones se pueden ejecutar programas, abrir carpetas y cambiar la apariencia del escritorio.
- Se desarrolló un módulo de configuración a través del cual se pueden cambiar algunos aspectos del sistema, realizar ajustes del comportamiento del hardware y verificar la comunicación de éste con el software.
- Los objetivos de este Trabajo Especial de Grado fueron alcanzados satisfactoriamente, obteniendo un producto final que cumple con las necesidades inicialmente planteadas, creando bases para desarrollos e investigaciones futuras en lo que se refiere a las formas de interactuar con el ordenador.

CONCLUSIONES

Luego de finalizadas las actividades propuestas para el desarrollo del proyecto se obtuvieron las siguientes conclusiones:

- La metodología del prototipado evolutivo es adecuada para el desarrollo de proyectos debido a que permite ir perfeccionando el producto a través de diversas iteraciones hasta obtener el resultado deseado, y también debido a su flexibilidad para incorporar etapas que permitan el desarrollo de elementos de hardware que formen parte integral del sistema en desarrollo.
- DirectX es un API con el que se pueden crear gráficos tridimensionales, con amplia documentación y buena integración con el lenguaje de programación C# y que facilitó el desarrollo del ambiente tridimensional y de los elementos contenidos en él.
- El rastreo óptico es una técnica fácilmente implementable y económicamente viable para la detección de movimientos.
- El Wii Remote es un dispositivo que brinda muchas facilidades debido a que posee una cámara infrarroja de gran versatilidad y un sistema de comunicaciones integrado y estándar (Bluetooth), y gracias a estas características se pudo obtener el dispositivo encargado de recibir la luz infrarroja y transmitirla directamente al ordenador.
- Se concluyó que la obtención de la información de la posición de la mano en el espacio podría ser más viable con la utilización de otra técnica de reconocimiento de movimientos en donde el usuario no tuviese la necesidad de colocarse ningún accesorio, y así tener aún más libertad en sus movimientos.

- Con las capacidades técnicas adecuadas, la correcta selección de tecnologías y algo de ingenio, se pueden desarrollar maneras novedosas y naturales de interactuar con el ordenador.

RECOMENDACIONES

- Para alcanzar un mayor nivel visual del que se tiene en el sistema se pueden realizar mejoras en el diseño de los elementos del escritorio 3D, así como también en lo relacionado a sombras e iluminaciones del ambiente virtual.
- Para lograr un nivel mayor de inmersión en el ambiente virtual se deberían mejorar las percepciones visuales del usuario con el ambiente tridimensional.
- Para brindarle al usuario del sistema mayor libertad en sus movimientos durante la manipulación del sistema es recomendable reemplazar los accesorios colocados en su mano por algún otro tipo de dispositivo más discreto a través del cual se pueda obtener la posición de la mano en el espacio.
- En relación con el hardware emisor-receptor de infrarrojo se puede plantear la ampliación de su campo de visualización, de manera que éste le brinde mayor área movimientos al usuario del sistema.
- Para mejorar el nivel de multisensorialidad percibida por el usuario se puede integrar en los accesorios que éste use algún mecanismo que genere vibraciones y así obtener una retroalimentación táctil de lo que ocurre en el ambiente virtual.
- Se deberían realizar estudios formales de los movimientos naturales de las manos de las personas y de las acciones más adecuadas a implementar para interactuar con el sistema usando el hardware, evaluando factores como la ergonomía; de manera que la manipulación del sistema se ajuste más a los movimientos que realiza la persona en su día a día.

BIBLIOGRAFÍA

Libros

Autodesk (2007). *3ds Max 9* (1ª ed.). España: Ediciones Anaya Multimedia.

Charte, F. (2007). *Visual C# 2005*. España: Anaya Multimedia.

Landa, N. (2006). *DirectX: Programación de Gráficos 3D* (1ª ed.). Argentina: M.P. Ediciones.

[Hollands. 1996] Hollands, R (1996). *The Virtual Reality Homebrewer's*. Editorial John-Wiley & Sons.

MediaActive (2007). *El gran libro de 3D Studio Max 9* (1ª ed.). España: Marcombo Ediciones Técnicas.

Sharp, J; Jagger, J. (2002). *Microsoft Visual C# .NET*. Editorial McGraw-Hill. 1ra Edición.

[Tanenbaum. 2003]. Tanenbaum, A (2003). *Redes de Computadoras*. (4ª ed.) Editorial Pearson-Prentice Hall. Pag. 310 a 317

[Wright R. y Lipchak, B. 2005] Wright R.; Lipchak, B (2005). *Programación OpenGL (3ª* ed.). Editorial Anaya Multimedia.

Wright, C. (2003). *Superutilidades para C#* (1ª ed.). España: McGraw-Hill.

Trabajos Especiales de Grado

[Federico. 2005] Federico, Dennis. (2005) *Diseño y Construcción de un Set de Realidad Virtual*. Tesis de Pre-Grado, Escuela de Ingeniería Informática, Universidad Católica Andrés Bello

[Marvi, Montezuma. Octubre 2005] Martínez M., Montezuma J. (2005) *Diseño e Implementación de Ejercicios para Entrenamiento Quirúrgico Virtual en Cirugía Mínima Invasiva*. Tesis de Pre-Grado, Escuela de Ingeniería Informática, Universidad Católica Andrés Bello.

Documentos electrónicos

[About Blue Soleil. 16/09/2008]. *About Blue Soleil*. Blue Soleil. 16/09/2008. Disponible en: http://www.bluesoleil.com/products/

Consultado el 16/09/08/ 07:45 p.m.

[Animación 3D. Sin fecha] *Animación 3D*. Disponible en: http://www.3de.es. Consultado el 19/09/2008/ 09:00 a.m.

[Bluetooth. 14/08/2008]. *Bluetooth*. Wikipedia. 14/08/2008

Disponible en: http://es.wikipedia.org/wiki/Bluetooth

Consultado el 18/09/2008/ 09:00 a.m.

[Computación Gráfica. Sin fecha] *Computación Gráfica*. Sin fecha

Disponible en http://www.mitecnologico.com/Main/ComputacionGrafica

Consultado el 21/09/18/ 02:00 p.m.

[DirectX. Sin fecha]. *DirectX*. Sin fecha

Disponible en: http://www.jeuazarru.com/docs/ Consultado el 14/09/2008/ 02:30 p.m.

[Entorno escritorio. Sin fecha]. *Entorno de escritorio*. 27/05/2008

Disponible en: http://es.wikipedia.org/wiki/Desktop_environment

Consultado el 14/09/18/ 04:30 p.m.

[Gráficos 3D por Computadora. 27/06/2008]. *Gráficos 3D por Computadora*. Wikipedia. 27/06/2008. Disponible en: http://es.wikipedia.org/wiki/Graficos3D

Consultado el 18/09/2008/ 09:00 p.m.

[Jacks. Sin Fecha] *Jacks*. KAPTON Electronics S.A. Sin Fecha. Disponible en: http://sitios.seccionamarilla.com.mx/kapton/default.asp?dept_id=2

Consultado el 14/09/18/ 02:00 p.m.

[Jugando con el Wiimote y GNU/Linux. Sin Fecha] Sin Autor. Jugando con el Wiimote y GNU/Linux. Santa Coloma de Gramenet. Sin fecha

Disponible en:

http://iespuigcastellar.xeill.net/Members/vcarceler/articulos/jugando-con-el-wiimote-y-gnu-linux

Consultado el 16/09/08/ 09:00 p.m.

[Garzás. Sin Fecha] Garzás, Javier. *Gestión de Proyectos Software*. Sin fecha

Disponible en http://kybele.escet.urjc.es/documentos/SI/%5BSI-2006-07%5DT4_Bases_del_Desarrollo_Software.pdf

Consultado el 14/09/18/ 01:00 p.m.

[Infrared Light Emitting Diode Illuminators Ready-to-Use PCB & Kits. Sin fecha]. *Infrared Light Emitting Diode Illuminators Ready-to-Use PCB & Kits* Disponible en: http://www.govideo.co.nz Consultado el: 21/09/2008 2:00 p.m

[Interruptores basculantes. Sin Fecha] *Interruptores basculantes*. ARISTON ELECTRÓNICA, S.A. Sin Fecha

Disponible en http://www.ariston.es/cat/catalogoConsulta.aspx?pagina=67;1 Consultado el 20/09/18/ 01:00 p.m.

[Introducción a Visual Studio. Sin fecha] Sin autor. *Introducción a Visual Studio*. Microsoft. Sin fecha. Disponible en: http://msdn.microsoft.com/es-es/library/fx6bk1f4(VS.80).aspx Consultado el 14/09/18/ 02:00 p.m.

[Láminas Reflectivas.24/10/2005]. Láminas Reflectivas. GUIADE. 24/10/2005. Disponible en: http://www.giade.com.ar/vial_pdf/laminas_reflectivas.pdf

Consultado el 19/09/18/ 05:00 p.m.

[Lee. 2008] Lee, J. Interaction Techniques Using The Wii Remote. Human Computer Interaction Institute. 2008

Disponible en http://uxweek.com/slides/wiimote-projects-uxweek.pdf

Consultado el: 17/09/08/ 09:40 p.m.

Lee, J. *Wii*. Sin Fecha.

Disponible en http://www.cs.cmu.edu/~johnny/projects/wii/ Consultado el 10/12/2007 03:30 p.m

Lee, J. (2007).*Wiimote Finger Tracking -FAQ and Adv Discussion*

Disponible en http://procrastineering.blogspot.com/2007/12/wiimote-finger-tracking-faq-and-adv.html Consultado el 10/12/2007 05:30 p.m

[Marcos. Sin fecha] Marcos M, M. *Presente, pasado y futuro de la investigación en interacción hombre-ordenador aplicada a la documentación*. Sin fecha. Disponible en: http://www.mcmarcos.com/pdf/2002_presente-leon.pdf

Consultado el 19/09/18/ 04:30 p.m.

[Méndez. Sin fecha] Méndez, V. *Wii*. Sin fecha

Disponible en: http://www.jeuazarru.com/docs/

Consultado el 14/09/18/ 02:30 p.m.

[.NET Manager Library for the Nintendo Wii Remote. 16/09/2008] *.NET Manager Library for the Nintendo Wii Remote*. WiimoteLib. 16/09/2008

Disponible en: http://www.brianpeek.com/blog/pages/wiimotelib.aspx/

Consultado el 16/09/08/ 09:00 p.m.

[Nichia Unveils White LED with 150 lm/W Luminous. 21/12/2006]. Nichia Unveils White LED with 150 lm/W Luminous 21/12/2006

Disponible en: http://techon.nikkeibp.co.jp/

Consultado el 19/09/2008/ 11:30 p.m.

[Nintendo patent application 2007/0060384. 13/09/2007] N*intendo patent application 2007/0060384*. Google Patent Search. 13/09/2007. Disponible en http://www.google.co.ve/patents?id=IZijAAAAEBAJ&pg=PA1&lpg=PA1&dq=0211027+nintendo&source=web&ots=D6vTsHgn1D&sig=Yeq907gNh2evXpGO5ypc7BlgBaY&hl=es&sa=X&oi=book_result&resnum=5&ct=result#PPA1,M1

Consultado el 20/08/2008/ 07:40 p.m.

[PandaSoft. Sin fecha] *PandaSoft*. Sin Fecha

Disponible en http://www.andytather.co.uk/Panda/directxmax_info.aspx

Consultado el 20/09/2008/ 07:00 p.m.

Peek, B (2007*) Managed Library for Nintendo's Wiimote*

Disponible en:
http://blogs.msdn.com/coding4fun/archive/2007/03/14/1879033.aspx

Consultado el 12/12/2007/ 07:00 p.m.

[Radiación Infrarroja. 13/07/2005]. *Radiación Infrarroja*. Ventanas al Universo. 13/07/2005

Disponible en: http://www.windows.ucar.edu

Consultado el 19/09/2008/ 10:00 p.m.

[Visual C#. Sin fecha]. *Visual C#*. Microsoft. Sin fecha

Disponible en: http://msdn.microsoft.com/es-es/library/kx37x362(VS.80).aspx

Consultado el 14/09/18/ 02:00 p.m.

WiimoteLib-.NET Managed Library for the Nintendo Wii Remote. BrianPeek.com. Disponible

http://www.brianpeek.com/blog/pages/wiimotelib.aspx. Consultado el 16/07/08/ 09:30 p.m

Wiimote Project. Disponible http://www.wiimoteproject.com/

Consultado el 16/12/08/ 09:30 p.m.

[Wii Remote. Sin Fecha] *Wii Remote*. Sin fecha

Disponible http://www.home-stop.co.uk/shop/images/wii%20remote.jpg

Consultado el 16/09/08/ 09:30 p.m.

[3Ds Max. 13/09/2008]. *3Ds Max*. Wikipedia. 13/09/2008

Disponible en: http://es.wikipedia.org/wiki/3D_Studio_Max

Consultado el 14/09/18/ 03:30 p.m.

[¿Que es el Infrarrojo?. 21/09/2000]. *¿Que es el Infrarrojo?*. 21/09/2000

Disponible en: http://www.spitzer.caltech.edu

Consultado el 19/09/2008/ 10:30 p.m.

APÉNDICE A: DISEÑO E IMPLEMENTACIÓN DEL HARDWARE DEL PROTOTIPO INICIAL

Para el diseño del circuito se determinó que cada LED infrarrojo debía ser colocado con una resistencia. La corriente que debía circular por cada uno de los LEDs fue calculada en 10 mA (corriente necesaria para el funcionamiento de este tipo de LEDs) por lo que la resistencia a colocar debía ser de 1 KOhm. Todo el conjunto sería alimentado por una fuente de poder de 12 voltios. (Ver Figura 33)

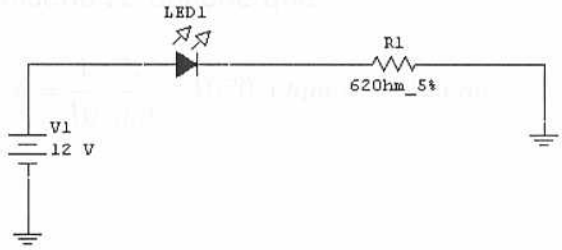


Figura 33. Diseño básico del circuito emisor de infrarrojo.
Fuente elaboración propia

Estos resultados fueron sustentados matemáticamente utilizando las ecuaciones de suma de potenciales que indica que:

$$V_f = V_1 + V_2 + ... + V_n$$

Por lo que aplicado al circuito diseñado se obtiene:

$$12v = 1{,}3v + 10{,}7v$$

Donde:

- 12 voltios representan el voltaje de la fuente de poder.
- 1,3 voltios representan el voltaje del LED emisor de infrarrojo.

- 10,7 voltios representa la diferencia de potencial en la resistencia.

Para determinar el valor adecuado para la resistencia se aplicó la Ley de Ohm que indica que:

$$V = I * R$$

Lo que implica que:

$$R = \frac{V}{I}$$

Que para el diseño se obtiene que:

$$R = \frac{10{,}7v}{10\ mA} = 1070\ Ohm \approx 1000\ Ohm = 1\ KOhm$$

Para obtener el área de irradiación necesaria se determinó que se debían colocar cincuenta (50) LEDs en serie, cada uno con una resistencia de 1 KOhm. (Ver Figura 34).

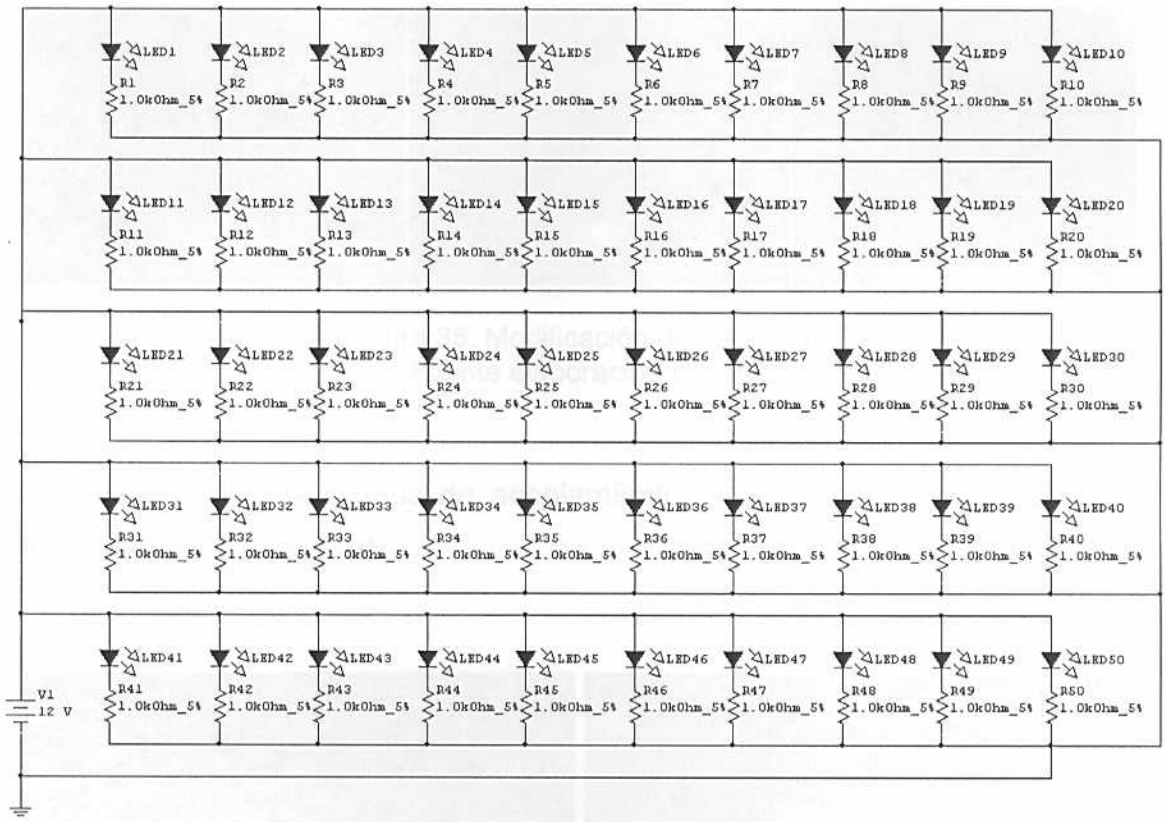


Figura 34. Diseño del circuito emisor de infrarrojo.
Fuente elaboración propia

La corriente total que circula por el circuito viene dada por la sumatoria de las corrientes de cada LEDs por lo que se obtiene:

$$I_{totak} = I_{led} * 50 = 10\ mA * 50 = 0{,}5\ A$$

Por lo que la fuente de poder principal a utilizar debía poder suplir por lo menos 0,5 A con 12 voltios DC.

Una vez finalizado el diseño del circuito emisor se procedió a construirlo, para lo cual se modificó una baquelita universal de manera que quedase en ella un espacio en donde se insertaría la cámara infrarroja. (Ver Figura 35).

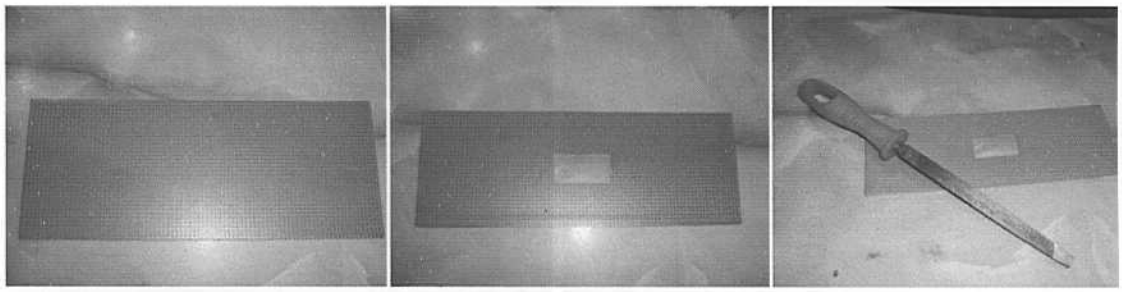


Figura 35. Modificación de baquelita
Fuente elaboración propia

Se realizó la prueba de acoplamiento entre el emisor y receptor de infrarrojo. (Ver Figura 36).

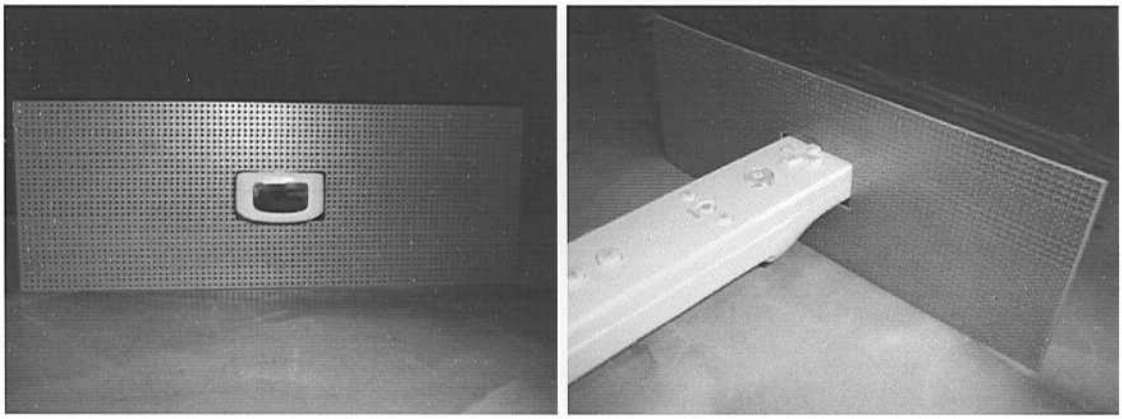


Figura 36. Prueba de acoplamiento emisor-receptor de infrarrojo
Fuente elaboración propia

Posteriormente se soldaron cada uno de los componentes del circuito emisor de infrarrojo y se le acopló la cámara infrarroja. (Ver Figura 37).

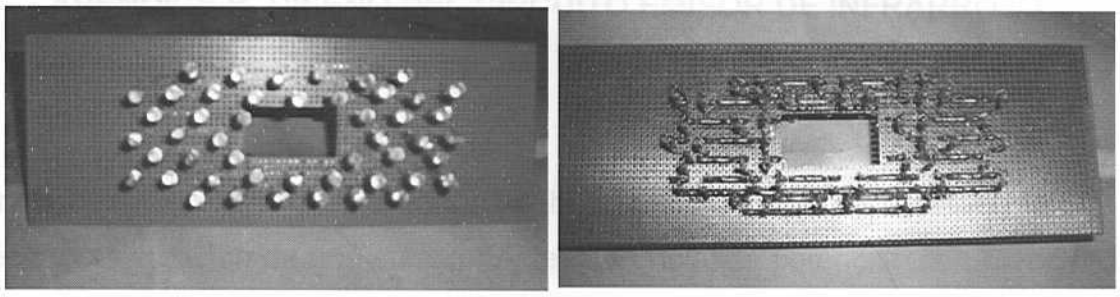


Figura 37. Circuito emisor de infrarrojo
Fuente elaboración propia

APÉNDICE B. DISEÑO DEL CIRCUITO EMISOR DE INFRARROJO

Para el diseño del circuito se determinó que cada LED infrarrojo trabaja a 1,3 voltios por lo que se debían colocar cinco de estos en serie; esta configuración da una sumatoria de diferencial de potencial de 6,5 voltios (1,3 voltios *5 LEDs IR) entre los LEDs inicial y final. Como era imprescindible (para el alcance del hardware) que estos LEDs trabajasen al máximo de potencia, la cual está aproximadamente entre 18 mA y 19 mA, se calculó que se debía colocar una resistencia de 300 Ohms en serie con los LEDs. Todo el conjunto sería alimentado por una fuente de poder de 12 voltios. (Ver Figura 38).

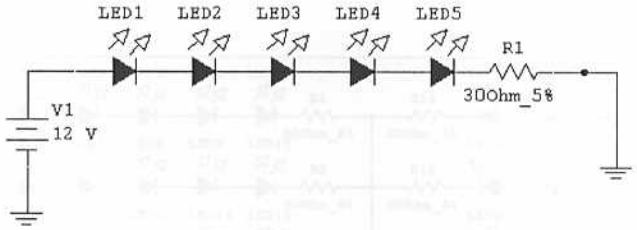


Figura 38. Diseño básico de los grupos del circuito emisor de infrarrojo.
Fuente elaboración propia

Estos resultados fueron sustentados matemáticamente utilizando las ecuaciones de suma de potenciales que indica que:

$$V_f = V_1 + V_2 + ... + V_n$$

Por lo que aplicado al circuito diseñado se obtiene:

$$12v = 1{,}3v + 1{,}3v + 1{,}3v + 1{,}3v + 1{,}3v + 5{,}5v\,.$$

Y para determinar el valor adecuado para la resistencia se aplicó la Ley de Ohm que indica que:

$$V = I * R$$

Lo que implica que:

$$R = \frac{V}{I}$$

Que para el diseño se obtiene que:

$$R = \frac{5{,}5\ v}{18\ mA} = 305{,}55\ Ohm \approx 300\ Ohm$$

Para obtener el área de irradiación necesaria se determinó que se debían colocar veinte (20) grupos de LEDs en serie, cada uno con una resistencia de 300 Ohm, con lo que se obtiene un circuito conformado por cien (100) LEDs infrarrojos. (Ver Figura 39).

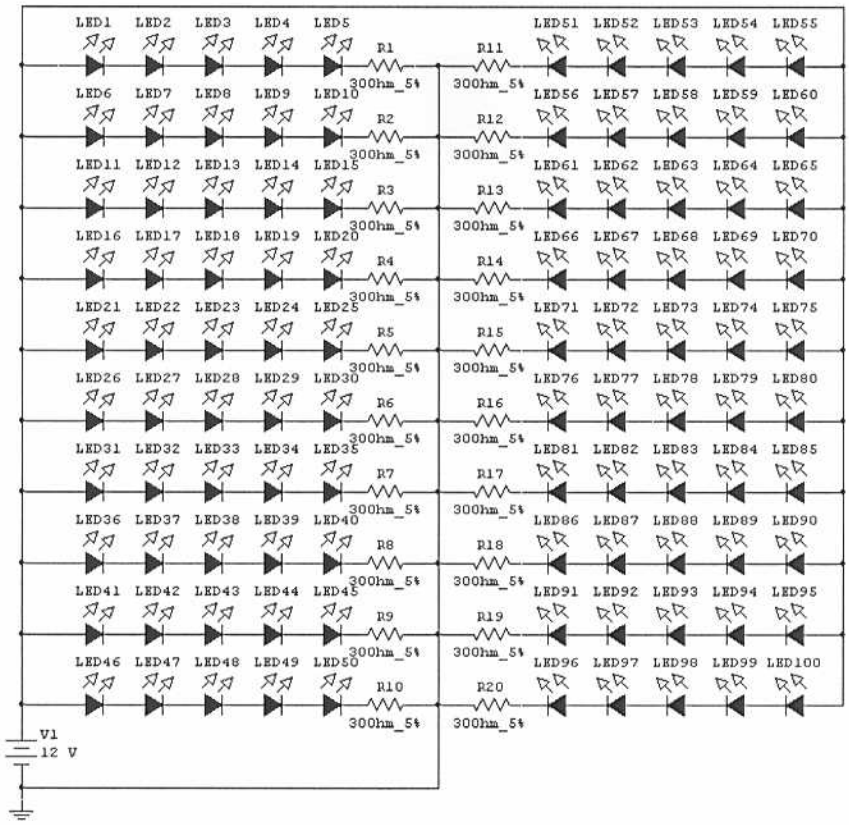


Figura 39. Diseño del circuito emisor de infrarrojo.

Fuente elaboración propia

La corriente total que circula por el circuito viene dada por la sumatoria de las corrientes de cada grupo de LEDs por lo que se obtiene:

$$I_{total} = I_{grupo} * 20 = 0{,}018 \ mA * 20 = 360 \ mA$$

Por lo que la fuente de poder principal a utilizar debía poder suplir por lo menos 360 mA con 12 voltios DC.

APÉNDICE C. DISEÑO DEL REGULADOR DE VOLTAJE DE 12 VOLTIOS A 3 VOLTIOS

Para el diseño del circuito regulador de voltaje de 12 v a 3 v utilizado para alimentar al dispositivo receptor de infrarrojo (Wii Remote) se utilizó un circuito integrado modelo 7805 el cual se encarga de regular la entrada de 12v (provenientes de la fuente de poder principal del hardware) a 5v, pero debido a que el consumo del dispositivo receptor (Wii Remote) es de 3v y que el integrado regula la tensión a 5v, se debieron configurar tres (3) diodos modelo 1N4001 en serie de manera que cada uno de ellos bajase el voltaje hasta conseguir los 3 voltios necesarios para alimentar el dispositivo receptor de infrarrojo. (Ver figura 40)

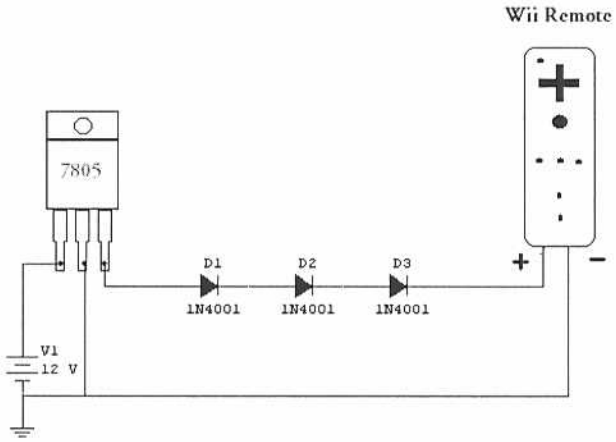


Figura 40. Diseño del circuito regulador de voltaje para el dispositivo receptor de infrarrojo.
Fuente elaboración propia

Esta configuración produce que los 12 v provenientes de la fuente principal del hardware sean inicialmente regulados a 5 v por el circuito integrado 7805 y que esta tensión de salida baje en 0,7 v (aproximadamente) con la colocación de cada diodo 1N4001 en serie, lo que da un voltaje final

de 2,9 voltios (aproximadamente). Este voltaje es ideal para el funcionamiento del dispositivo receptor de infrarrojo.

APÉNDICE D. CASOS DE USO

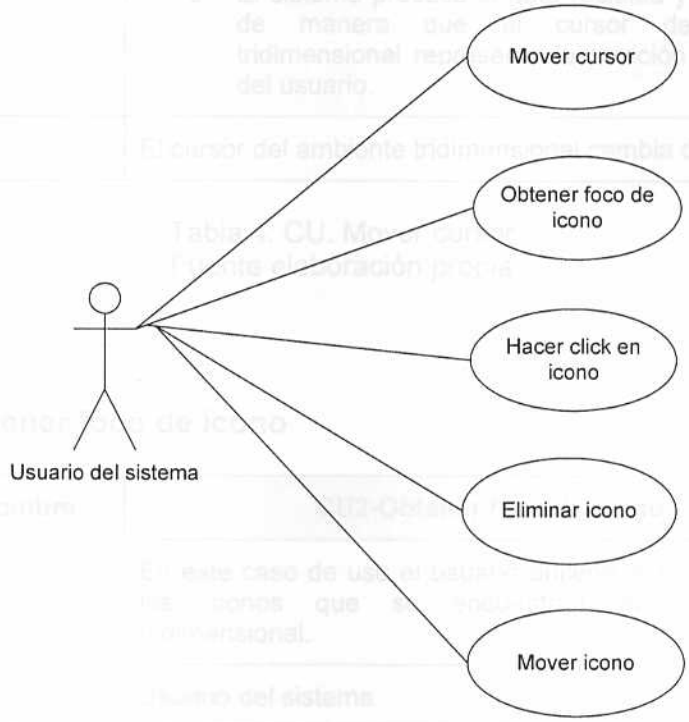


Figura 41. Diagrama de casos de uso del sistema.
Fuente elaboración propia

CU. Mover cursor

Nro. Caso y Nombre	CU1-Mover cursor
Descripción	En este caso de uso el usuario desplaza el cursor en el ambiente tridimensional a través de los movimientos de su mano.
Actores	Usuario del sistema
Pre-condición	Debe haberse realizado la calibración de profundidad del sistema.
Disparador	Movimientos de la mano del usuario
Flujo de Eventos	1. El usuario mueve la mano en donde tiene colocados los accesorios reflectivos en dirección al dispositivo emisor-receptor de infrarrojo. 2. El dispositivo emisor-receptor captura la posición de los puntos reflectivos y envía esa data vía

	Bluetooth al ordenador. 3. El sistema procesa la data recibida y la interpreta de manera que el cursor del ambiente tridimensional represente la posición de la mano del usuario.
Post condición	El cursor del ambiente tridimensional cambia de posición.

Tabla 4. CU. Mover cursor
Fuente elaboración propia

CU. Obtener foco de ícono

Nro. Caso y Nombre	CU2-Obtener foco de ícono
Descripción	En este caso de uso el usuario obtiene el foco de uno de los íconos que se encuentran en el escritorio tridimensional.
Actores	Usuario del sistema
Pre-condición	Debe haberse realizado la calibración de profundidad del sistema.
Disparador	Movimiento del cursor del escritorio tridimensional.
Flujo de Eventos	1. El usuario mueve el cursor hacia uno de los íconos del escritorio tridimensional. 2. El sistema calcula la distancia entre el cursor y el ícono. 3. Si la distancia entre ambos es lo suficientemente pequeña se obtiene el foco del ícono.
Post condición	El usuario obtiene el foco de uno de los íconos del escritorio tridimensional.

Tabla 5. CU. Obtener foco de ícono
Fuente elaboración propia

CU. Hacer clic en ícono

Nro. Caso y Nombre	CU3-Hacer clic en ícono
Descripción	En este caso de uso el usuario hace clic sobre uno de los

	íconos del escritorio tridimensional.
Actores	Usuario del sistema
Pre-condición	Debe haberse realizado la calibración de profundidad del sistema y obtenerse el foco de un ícono.
Disparador	Movimiento vertical (dos veces) del cursor en un ícono dentro del escritorio tridimensional.
Flujo de Eventos	1. El usuario mueve el cursor del escritorio tridimensional en sentido vertical ascendente. 2. El usuario mueve el cursor en sentido vertical descendente para regresar el cursor rápidamente al ícono al que se le desea hacer clic. 3. El usuario repite el movimiento vertical ascendente y descendente en un periodo de tiempo corto.
Post condición	El usuario ejecuta clic sobre el ícono.

Tabla 6. CU. Hacer clic en ícono
Fuente elaboración propia

CU. Eliminar ícono

Nro. Caso y Nombre	CU4-Eliminar ícono
Descripción	En este caso de uso el usuario elimina uno de los íconos del escritorio tridimensional.
Actores	Usuario del sistema
Pre-condición	Debe haberse realizado la calibración de profundidad del sistema y obtenerse el foco de un ícono.
Disparador	Movimiento horizontal (cuatro veces) del cursor en un ícono dentro del escritorio tridimensional.
Flujo de Eventos	1. El usuario mueve el cursor del escritorio horizontalmente. 2. El usuario vuelve a mover el cursor horizontalmente para regresarlo rápidamente al ícono al que se desea eliminar. 3. El usuario repite los movimientos horizontales tres veces en un periodo de tiempo corto. 4. El sistema despliega un cuadro de mensaje para

	la confirmación de la eliminación del ícono. 5. Si el usuario confirma la eliminación el ícono es eliminado del escritorio.
Post condición	El usuario elimina un ícono del escritorio tridimensional.

Tabla 7. CU. Eliminar ícono
Fuente elaboración propia

CU. Mover ícono

Nro. Caso y Nombre	CU5-Mover ícono
Descripción	En este caso de uso el usuario mueve uno de los íconos que se encuentran en el escritorio tridimensional.
Actores	Usuario del sistema
Pre-condición	Debe haberse realizado la calibración de profundidad del sistema.
Disparador	Giro de la mano y movimiento horizontal (una vez) del cursor en un ícono dentro del escritorio tridimensional.
Flujo de Eventos	1. El usuario mueve el cursor hasta posicionarlo del lado izquierdo de uno de los íconos del escritorio tridimensional. 2. El usuario gira la mano hacia la derecha. 3. El usuario mueve el cursor del escritorio horizontalmente hacia la izquierda atravesando el ícono y cerrando el dedo ocultándolo. 4. El sistema interpreta que se quiso agarrar el elemento, se pone en foco y permite su movimiento. 5. El usuario mueve el ícono hasta la posición deseada en el escritorio tridimensional. 6. El usuario despliega el dedo del cursor, lo cual fija el ícono en su nueva posición.
Post condición	El usuario mueve uno de los íconos del escritorio tridimensional.

Tabla 8. CU. Mover ícono
Fuente elaboración propia

APÉNDICE E. INTEGRACIÓN DE COMPLEMENTOS DE USABILIDAD DEL HARDWARE EMISOR-RECEPTOR DE INFRARROJO

Para poder controlar el encendido y apagado del hardware de una manera más apropiada, se procedió a la incorporación de un interruptor (Ver Figura 42).

Figura 42. Interruptor incorporado al hardware.
Fuente [Interruptores basculantes. Sin Fecha]

De igual manera se incorporó en el cajetín un conector hembra (Ver Figura 43) con el objetivo de conectar en él la fuente de poder que alimenta al hardware la cual posee un conector macho del mismo tipo (Ver Figura 44).

Figura 43. Conector hembra incorporado al hardware.
Fuente [Jacks. Sin Fecha]

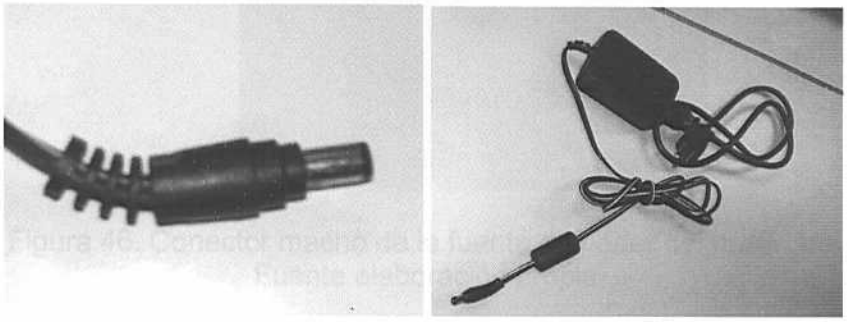


Figura 44. Conector macho de la fuente de poder del hardware.
Fuente elaboración propia

Los resultados de estas modificaciones se pueden apreciar a continuación (ver Figura 45)

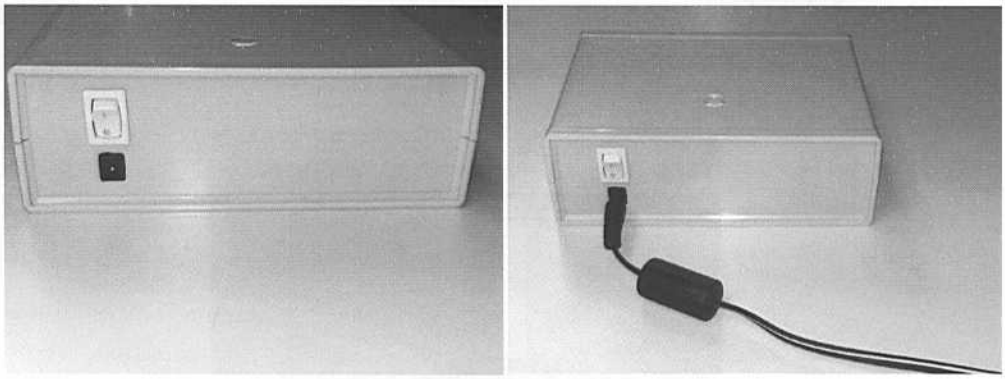


Figura 45. Interruptor de encendido/apagado y conector de fuente de poder incorporados en el hardware.
Fuente elaboración propia

Para mejorar la forma de realizar la sincronización de comunicaciones Bluetooth entre el dispositivo receptor de infrarrojos y el computador se integró un pulsador en la parte superior del cajetín (Ver Figura 46), a través del cual, se transmite de manera mecánica la acción del pulsar sobre los interruptores originales de la cámara infrarroja (Wii Remote).

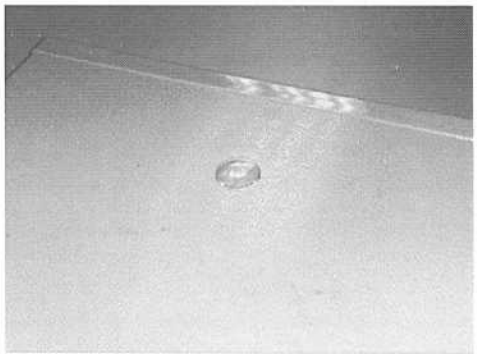


Figura 46. Conector macho de la fuente de poder del hardware.
Fuente elaboración propia

Finalmente se procedió a incorporar en la parte frontal del cajetín una lámina acrílica hecha a medida con el fin de proteger de cualquier golpe a los diodos LEDs sin interrumpir el nivel de irradiación de luz infrarroja. (Ver Figura 47)

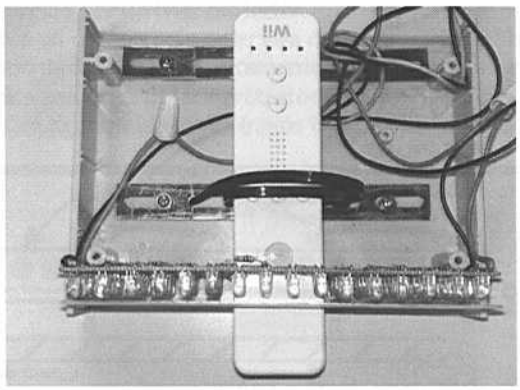


Figura 47. Lámina acrílica incorporada al hardware.
Fuente elaboración propia

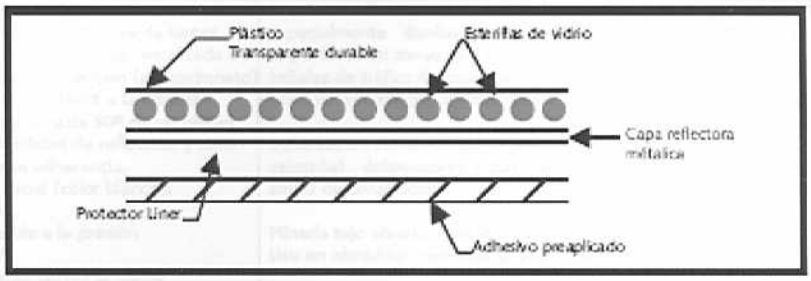
ANEXO A: MATERIAL REFLECTIVO

LÁMINAS REFLECTIVAS

Existen diversos tipos de laminas para la confeccion de los carteles, es importante describir cada una de ellas para saber cuando utilizar una u otra.

LAMINAS REFLECTIVAS GRADO INGENIERIA

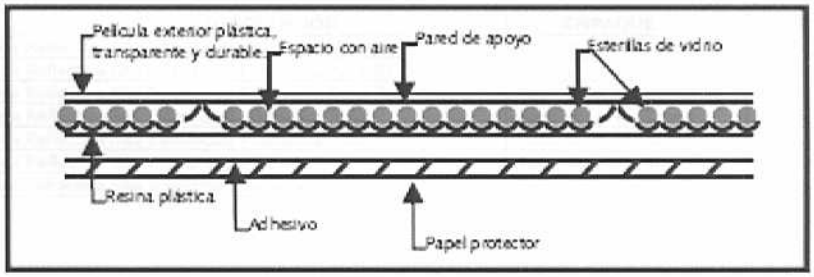
Básicamente las laminas reflectivas están conformadas por un material base en cuya cara inferior se aplica un pegamento protegido por un liner protector.
En la cara superior se aplica un una película reflectora metálica y sobre esta una pelicula de esferas microscópicas de vidrio orientadas electroestaticamente que son las que reflejan la luz.
Sobre esta pelicula se aplica una capa de laca protectora que es la que esta en contacto con el medio ambiente y la protege de los agentes agresivos (rayos UV, viento, lluvia, etc.)



Este tipo de laminas se utilizan para la confección de carteles de señalización vial, sobre todo en caminos secundarios o en lugares donde el transito no es muy intenso. Se presentan en colores blanco, amarillo, rojo, verde, azul, naranja, y cebrado naranja y blanco, en rollos de 1.22 mts. de ancho y 45/50 mts. de largo

LAMINAS REFLECTIVAS ALTA INTENSIDAD.

Las laminas reflectivas de alta intensidad tienen el mismo principio de construcción que las laminas grado ingeniería, la diferencia radica en que se usa la técnica de encapsulamiento.
La pelicula que cubre las esferillas de vidrio no descansa sobre las esferillas, sino sobre paredes construidas de forma hexagonal u otro forma similar. Esto permite la presencia de aire entre las esferillas y la pelicula que la cubre, este tipo de construcción es lo que hace posible tener un brillo superior en 3.5 veces a las laminas grado ingeniería un grado de angularidad muy superior a cualquier tipo de superficie o lamina.
La presentación es similar a las laminas grado ingeniaria.



Este tipo de laminas son utilizadas para la confección de carteles viales en lugares donde la visibilidad debe ser optima. Actualmente es norma el uso este tipo de laminas para carteles en autopistas e identificación de transporte automotor de carga y pasajeros.

Rev. 1.0 24/10/2005

CARP Y ASOCIADOS 3M Material Reflectivo

Lamina Reflectiva
Alta Intensidad Prismática (HIP) 3M

DESCRIPCIÓN	USOS	BENEFICIOS
Lámina Reflectiva autoadhesiva de lentes Micro prismáticos trucados, no metalizada de Buena capacidad de reflección (policarbonato) Uso vertical y no resistente a la abrasión Buena angularidad, hasta 90º de visibilidad Mantiene la estabilidad de reflección y color Durable y de gran adherencia. - Reflectividad inicial (color blanco): 360 cd/ Lux m2. - Adhesivo sensible a la presión **Clase III / IV** **Especificaciones ASTM D-4956**	Especialmente diseñada para reflectorizar dispositivos en zonas de construcción y de señales de tráfico durables, los cuales se exponen verticalmente en servicio. Señalización vial en zonas de alta velocidad , delineadores y barricadas y zonas de construcción. Minería tajo abierto y socavón. Uso en identificar vehículos y publicidad	- Alta reflectividad inicial, estable y retenida. - Mayor visibilidad en cualquier Situación, angularidad hasta 90° - Garantía de 10 años con el 70% de Reflección y color estable no se Decolora.

MÉTODO DE APLICACIÓN	MANTENCIÓN – PRECAUCIÓN
Para instalar lámina **Grado Alta Intensidad (HIP)** se debe acondicionar la superficie, limpio de contaminantes y/o de y sustancias que puedan reducir el adherencia. Una vez instalado pasar una paleta para sacar el aire interior y un trapo para sellar los bordes. Se puede instalar en Metal, cemento muy liso, fibra de vidrio y en superficies totalmente lizas y limpias. Aplicar a un mínimo de 8º C de temperatura ambiente No recomendable en superficies porosas, madera ni triplay.	- La lámina **Grado Alta Intensidad Prismática (HIP)** se debe de limpiar con un paño no abrasivo y enjuagar con abundante agua. No utilizar solventes ni diluyentes, deteriora la cara exterior de la lámina. Utilizar la lámina antes de un año de adquirido. Almacenar en posición horizontal en su empaque original y a una temperatura ambiente, entre 18ºC y 25ºC.

FACTOR DE REFLECCIÓN TÍPICO

Ra	Blanco	Amarillo	Rojo	Azul	Verde	Anaranjado
Angulo de entrada -4.0°	360	270	65	30	50	145
Angulo de Observación 0.2°						

CÓDIGO	DESCRIPCIÓN	EMPAQUE	Stock
3930	Lámina Reflectiva Alta Intensidad Prismática Blanca	Rollo 36 x 45.70 mts.	Inmediato
3931	Lámina Reflectiva Alta Intensidad Prismática Amarilla	Rollo 30 x 45.70 mts	Inmediato
3931	Lámina Reflectiva Alta Intensidad Prismática Amarilla	Rollo 36" x 45.70 mts	Inmediato
3932	Lámina Reflectiva Alta Intensidad Prismática Roja	Rollo 30" x 45.70 mts	Inmediato
3935	Lámina Reflectiva Alta Intensidad Prismática Azul	Rollo 36" x 45.70 mts	45 días
3037	Lámina Reflectiva Alta Intensidad Verde	Rollo 36" x 45.70 mts	45 días
3934	Lámina Reflectiva Alta Intensidad Naranja	Rollo 36" x 45.70 mts	45 días

ANEXO B: EXTRACTOS DE LA DOCUMENTACIÓN DE LA LIBRERÍA WIMOTELIB

Classes

Class	Description
Wiimote	Implementation of Wiimote
WiimoteChangedEventArgs	Argument sent through the WiimoteChangedEvent
WiimoteExtensionChangedEventArgs	Argument sent through the WiimoteExtensionChangedEvent
WiimoteState	Current overall state of the Wiimote and all attachments

Structures

Structure	Description
AccelCalibrationInfo	Accelerometer calibration information
AccelState	Current state of the accelerometers
ButtonState	Current button state
ClassicControllerButtonState	Curernt button state of the Classic Controller
ClassicControllerCalibrationInfo	Calibration information stored on the Classic Controller
ClassicControllerState	Current state of the Classic Controller

IRState	Current state of the IR camera
LEDState	Current state of LEDs
NunchukCalibrationInfo	Calibration information stored on the Nunchuk
NunchukState	Current state of the Nunchuk extension

Delegates

Delegate	Description
WiimoteChangedEventHandler	Event to handle a state change on the Wiimote
WiimoteExtensionChangedEventHandler	Event to handle insertion/removal of an extension (Nunchuk/Classic Controller)

Enumerations

Enumeration	Description
ExtensionType	The extension plugged into the Wiimote
IRMode	The mode of data reported for the IR sensor
Wiimote..::.InputReport	The report format in which the Wiimote should return data

IRMode Enumeration

See Also Send Feedback

The mode of data reported for the IR sensor

Namespace: WiimoteLib
Assembly: WiimoteLib (in WiimoteLib.dll)
Version: 1.2.1.0 (1.2.1.0)

Syntax

C#

```
public enum IRMode
```

Visual Basic (Declaration)

```
Public Enumeration IRMode
```

Visual C++

```
public enum class IRMode
```

Members

Member name	Description
Off	IR sensor off
Basic	Basic mode
Extended	Extended mode
Full	Full mode (unsupported)

IRState Structure

Members See Also Send Feedback

Current state of the IR camera

Namespace: WiimoteLib
Assembly: WiimoteLib (in WiimoteLib.dll)
Version: 1.2.1.0 (1.2.1.0)

Syntax

C#

```
public struct IRState
```

Visual Basic (Declaration)

```
Public Structure IRState
```

Visual C++

```
public value class IRState
```

IRState Members

IRState Structure Methods Fields See Also Send Feedback

The IRState type exposes the following members.

Methods

Name	Description
Equals	Indicates whether this instance and a specified object are equal. (Inherited from ValueType.)
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)
GetHashCode	Returns the hash code for this instance. (Inherited from ValueType.)
GetType	Gets the Type of the current instance. (Inherited from Object.)
MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
ToString	Returns the fully qualified type name of this instance.

	Name	Description
	Found1	IR sensor seen
	Found2	IR sensor seen
	Found3	IR sensor seen
	Found4	IR sensor seen
	MidX	Normalized midpoint of IR sensors 1 and 2 only. Values range between 0.0 - 1.0
	MidY	Normalized midpoint of IR sensors 1 and 2 only. Values range between 0.0 - 1.0
	Mode	Current mode of IR sensor data
	RawMidX	Raw midpoint of IR sensors 1 and 2 only. Values range between 0 - 1023, 0 - 767
	RawMidY	Raw midpoint of IR sensors 1 and 2 only. Values range between 0 - 1023, 0 - 767
	RawX1	Raw value of X-axis on individual sensor. Values

	RawX2	Raw value of X-axis on individual sensor. Values range between 0 - 1023
	RawX3	Raw value of X-axis on individual sensor. Values range between 0 - 1023
	RawX4	Raw value of X-axis on individual sensor. Values range between 0 - 1023
	RawY1	Raw value of Y-axis on individual sensor. Values range between 0 - 767
	RawY2	Raw value of Y-axis on individual sensor. Values range between 0 - 767
	RawY3	Raw value of Y-axis on individual sensor. Values range between 0 - 767
	RawY4	Raw value of Y-axis on individual sensor. Values range between 0 -

	Size1	Size of IR Sensor. Values range from 0 - 15
	Size2	Size of IR Sensor. Values range from 0 - 15
	Size3	Size of IR Sensor. Values range from 0 - 15
	Size4	Size of IR Sensor. Values range from 0 - 15
	X1	Normalized value of X-axis on individual sensor. Values range between 0.0 - 1.0
	X2	Normalized value of X-axis on individual sensor. Values range between 0.0 - 1.0
	X3	Normalized value of X-axis on individual sensor. Values range between 0.0 - 1.0
	X4	Normalized value of X-axis on individual sensor. Values range between 0.0 - 1.0
	Y1	Normalized value of

IRState Fields

IRState Structure See Also Send Feedback

The IRState type exposes the following members.

Fields

Name	Description
Found1	IR sensor seen
Found2	IR sensor seen
Found3	IR sensor seen
Found4	IR sensor seen
MidX	Normalized midpoint of IR sensors 1 and 2 only. Values range between 0.0 - 1.0
MidY	Normalized midpoint of IR sensors 1 and 2 only. Values range between 0.0 - 1.0
Mode	Current mode of IR sensor data
RawMidX	Raw midpoint of IR sensors 1 and 2 only. Values range between 0 - 1023, 0 - 767
RawMidY	Raw midpoint of IR sensors 1 and 2 only. Values range between 0 - 1023, 0 - 767
RawX1	Raw value of X-axis on individual sensor. Values range between 0 - 1023

RawX2	Raw value of X-axis on individual sensor. Values range between 0 - 1023
RawX3	Raw value of X-axis on individual sensor. Values range between 0 - 1023
RawX4	Raw value of X-axis on individual sensor. Values range between 0 - 1023
RawY1	Raw value of Y-axis on individual sensor. Values range between 0 - 767
RawY2	Raw value of Y-axis on individual sensor. Values range between 0 - 767
RawY3	Raw value of Y-axis on individual sensor. Values range between 0 - 767
RawY4	Raw value of Y-axis on individual sensor. Values range between 0 - 767
Size1	Size of IR Sensor. Values range from 0 - 15
Size2	Size of IR Sensor. Values range from 0 - 15
Size3	Size of IR Sensor. Values range from 0 - 15
Size4	Size of IR Sensor. Values range from 0 - 15

X1	Normalized value of X-axis on individual sensor. Values range between 0.0 - 1.0
X2	Normalized value of X-axis on individual sensor. Values range between 0.0 - 1.0
X3	Normalized value of X-axis on individual sensor. Values range between 0.0 - 1.0
X4	Normalized value of X-axis on individual sensor. Values range between 0.0 - 1.0
Y1	Normalized value of Y-axis on individual sensor. Values range between 0.0 - 1.0
Y2	Normalized value of Y-axis on individual sensor. Values range between 0.0 - 1.0
Y3	Normalized value of Y-axis on individual sensor. Values range between 0.0 - 1.0
Y4	Normalized value of Y-axis on individual sensor. Values range between 0.0 - 1.0

Wiimote Class

Members See Also Send Feedback

Implementation of Wiimote

Namespace: WiimoteLib
Assembly: WiimoteLib (in WiimoteLib.dll)
Version: 1.2.1.0 (1.2.1.0)

Syntax

C#

```
public class Wiimote : IDisposable
```

Visual Basic (Declaration)

```
Public Class Wiimote _
    Implements IDisposable
```

Visual C++

```
public ref class Wiimote : IDisposable
```

Inheritance Hierarchy

System..::.Object
WiimoteLib..::.Wiimote

Wiimote Members

Wiimote Class Constructors Methods Properties Events See Also Send Feedback

The Wiimote type exposes the following members.

Constructors

Name	Description
Wiimote	Default constructor

Methods

Name	Description
Connect	Connect to a Wiimote paired to the PC via Bluetooth
Disconnect	Disconnect from the controller and stop reading data from it
Dispose	Overloaded.
Equals	Determines whether the specified Object is equal to the current Object. (Inherited from Object.)
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.

	(Inherited from Object.)
GetHashCode	Serves as a hash function for a particular type. GetHashCode()()() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object.)
GetStatus	Retrieve the current status of the Wiimote and extensions. Replaces GetBatteryLevel() since it was poorly named.
GetType	Gets the Type of the current instance. (Inherited from Object.)
MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
ReadData	Read data or register from Wiimote
SetLEDs	Overloaded.
SetReportType	Set Wiimote reporting mode
SetRumble	Toggle rumble
ToString	Returns a String that represents the current Object. (Inherited from Object.)
WriteData	Overloaded.

Properties

	Name	Description
	AltWriteMethod	**Obsolete.** Force the PC to write to the Wiimote in an alternate way.
		This should no longer be used.
	WiimoteState	Current Wiimote state

Events

	Name	Description
	WiimoteChanged	Event raised when Wiimote state is changed
	WiimoteExtensionChanged	Event raised when an extension is inserted or removed

Wiimote Constructor

Wiimote Class See Also Send Feedback

Default constructor

Namespace: WiimoteLib
Assembly: WiimoteLib (in WiimoteLib.dll)
Version: 1.2.1.0 (1.2.1.0)

Syntax

C#

```
public Wiimote()
```

Visual Basic (Declaration)

```
Public Sub New
```

Visual C++

```
public:
Wiimote()
```

Wiimote Methods

Wiimote Class See Also Send Feedback

The Wiimote type exposes the following members.

Methods

Name	Description
Connect	Connect to a Wiimote paired to the PC via Bluetooth
Disconnect	Disconnect from the controller and stop reading data from it
Dispose	Overloaded.
Equals	Determines whether the specified Object is equal to the current Object. (Inherited from Object.)
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)
GetHashCode	Serves as a hash function for a particular type. GetHashCode()()() is suitable for use in hashing algorithms and data structures like a hash table.

	(Inherited from Object.)
GetStatus	Retrieve the current status of the Wiimote and extensions. Replaces GetBatteryLevel() since it was poorly named.
GetType	Gets the Type of the current instance. (Inherited from Object.)
MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
ReadData	Read data or register from Wiimote
SetLEDs	Overloaded.
SetReportType	Set Wiimote reporting mode
SetRumble	Toggle rumble
ToString	Returns a String that represents the current Object. (Inherited from Object.)
WriteData	Overloaded.

Wiimote Properties

Wiimote Class See Also Send Feedback

The Wiimote type exposes the following members.

Properties

Name	Description
AltWriteMethod	**Obsolete.** Force the PC to write to the Wiimote in an alternate way. This should no longer be used.
WiimoteState	Current Wiimote state

Wiimote Events

Wiimote Class See Also Send Feedback

The Wiimote type exposes the following members.

Events

Name	Description
WiimoteChanged	Event raised when Wiimote state is changed
WiimoteExtensionChanged	Event raised when an extension is inserted or removed

WiimoteState Class

Members See Also Send Feedback

Current overall state of the Wiimote and all attachments

Namespace: WiimoteLib
Assembly: WiimoteLib (in WiimoteLib.dll)
Version: 1.2.1.0 (1.2.1.0)

Syntax

C#

```
public class WiimoteState
```

Visual Basic (Declaration)

```
Public Class WiimoteState
```

Visual C++

```
public ref class WiimoteState
```

WiimoteState Members

WiimoteState Class Constructors Methods Fields See Also Send Feedback

The WiimoteState type exposes the following members.

Constructors

	Name	Description
	WiimoteState	

Methods

	Name	Description
	Equals	Determines whether the specified Object is equal to the current Object. (Inherited from Object.)
	Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as a hash function for a particular type. GetHashCode()()() is suitable for use in hashing algorithms and data structures like a hash table. (Inherited from Object.)

GetType	Gets the Type of the current instance. (Inherited from Object.)
MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
ToString	Returns a String that represents the current Object. (Inherited from Object.)

Fields

	Name	Description
	AccelCalibrationInfo	Current calibration information
	AccelState	Current state of accelerometers
	Battery	Current battery level
	ButtonState	Current state of buttons
	ClassicControllerState	Current state of Classic Controller extension
	Extension	Is an extension controller inserted?
	ExtensionType	Extension controller currently inserted, if any
	IRState	Current state of IR sensors
	LEDState	Current state of LEDs
	NunchukState	Current state of Nunchuk extension
	Rumble	Current state of rumble

ANEXO C: PATENTE DEL WII REMOTE

(19) **United States**
(12) **Patent Application Publication** (10) **Pub. No.: US 2007/0060384 A1**
Dohta (43) **Pub. Date: Mar. 15, 2007**

(54) **STORAGE MEDIUM STORING VIDEO GAME PROGRAM**

(75) Inventor: **Takuhiro Dohta**, Kyoto-shi (JP)

Correspondence Address:
NIXON & VANDERHYE, P.C.
901 NORTH GLEBE ROAD, 11TH FLOOR
ARLINGTON, VA 22203 (US)

(73) Assignee: **Nintendo Co., Ltd.**, Kyoto (JP)

(21) Appl. No.: **11/429,249**

(22) Filed: **May 8, 2006**

(30) **Foreign Application Priority Data**

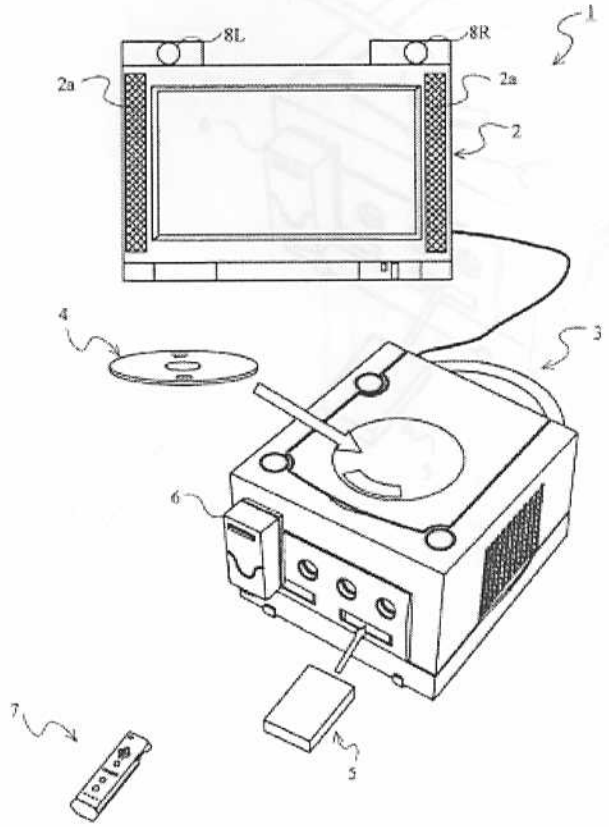
Sep. 14, 2005 (JP) 2005-267180

Publication Classification

(51) **Int. Cl.**
A63F 13/00 (2006.01)
(52) **U.S. Cl.** **463/43**

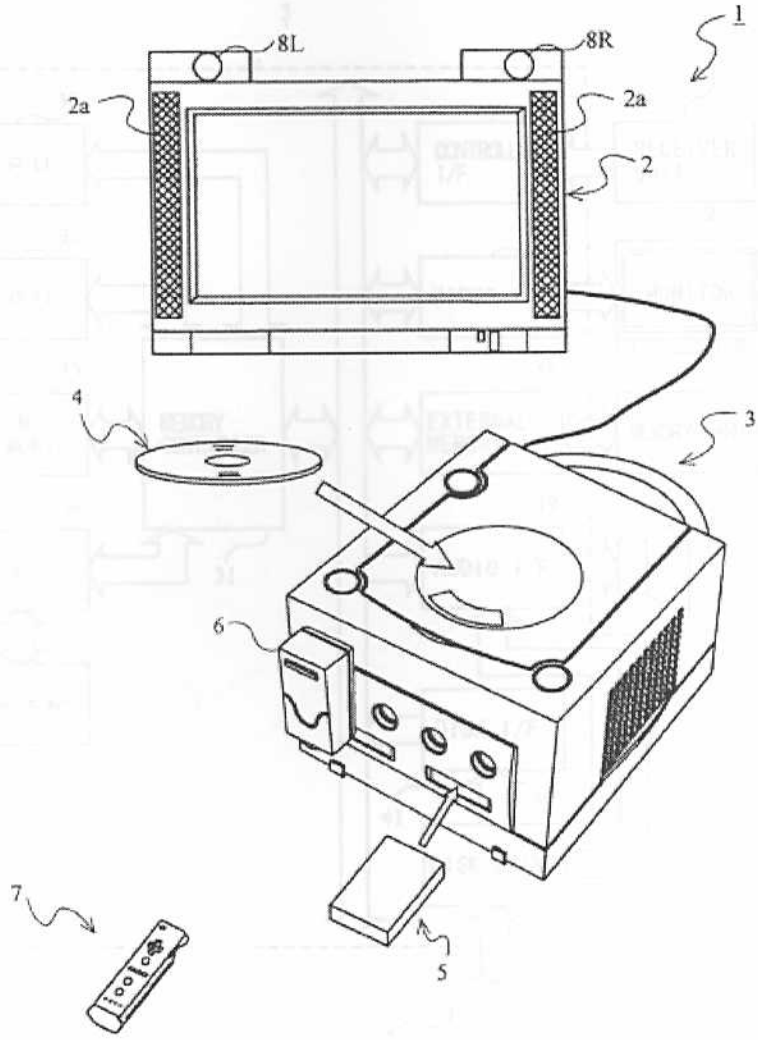
(57) **ABSTRACT**

A game process is performed by obtaining control data from a game controller. Distance data associated with the distance between the game controller and a predetermined measured reference provided in a real space is calculated by using the control data obtained from the game controller. Then, a predetermined game process is performed on a virtual game world through an operation based on the value of the distance data. Then, the virtual game world after the game process is displayed on a display device.



Patent Application Publication Mar. 15, 2007 Sheet 1 of 24 US 2007/0060384 A1

F.I G. 1



Patent Application Publication Mar. 15, 2007 Sheet 2 of 24 US 2007/0060384 A1

FIG. 2

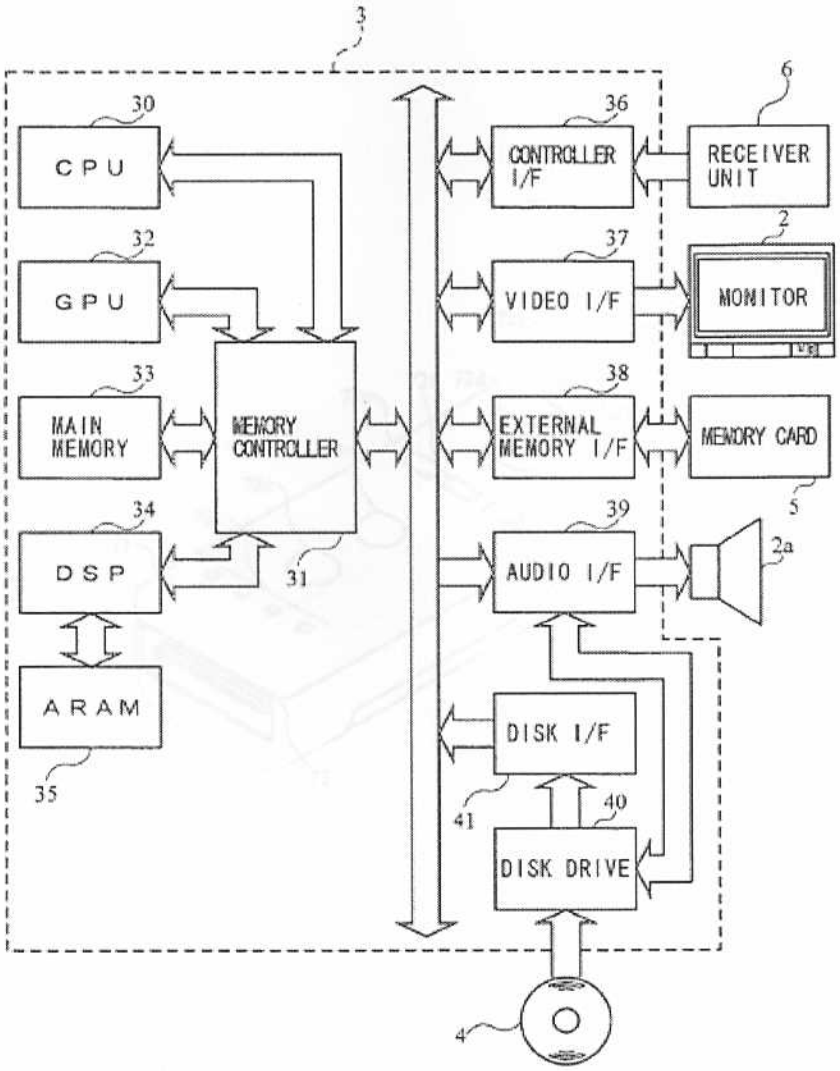
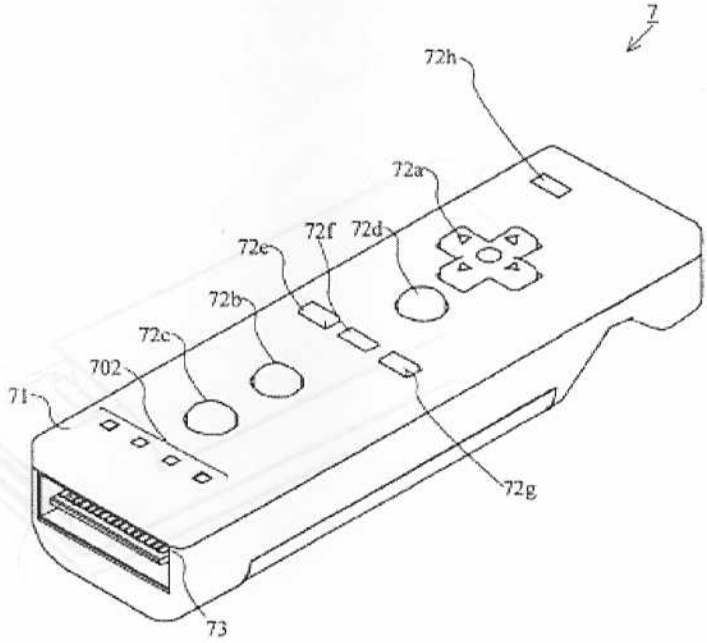


FIG. 3



F I G. 4

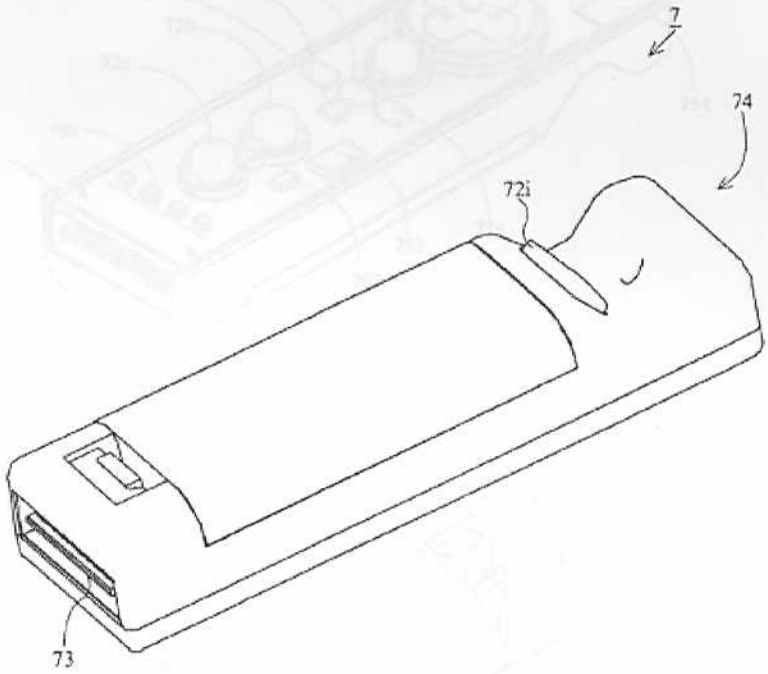


FIG. 5A

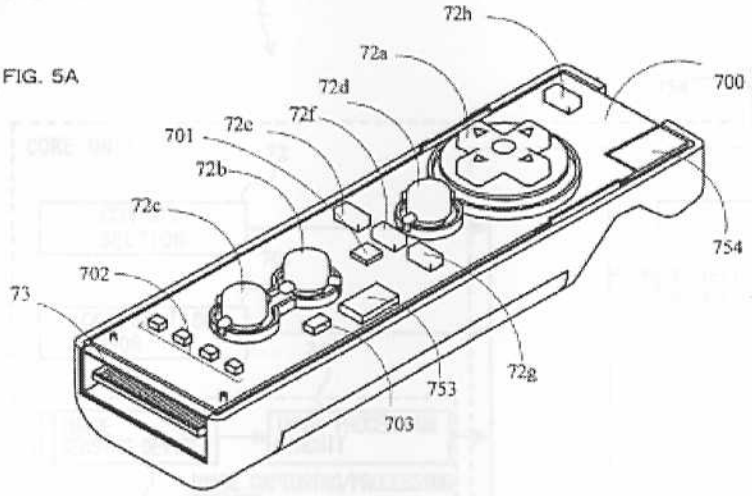

72h
72a
700
72d
72f
72e
701
72b
72c
754
702
73
72g
753
703

FIG. 5B

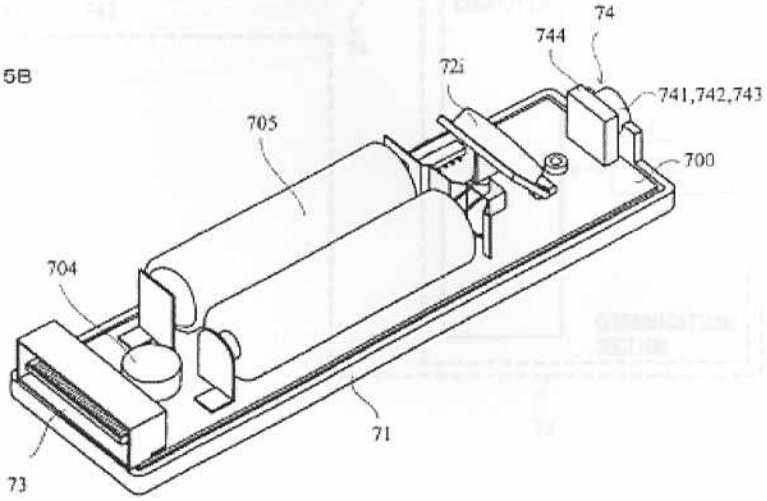
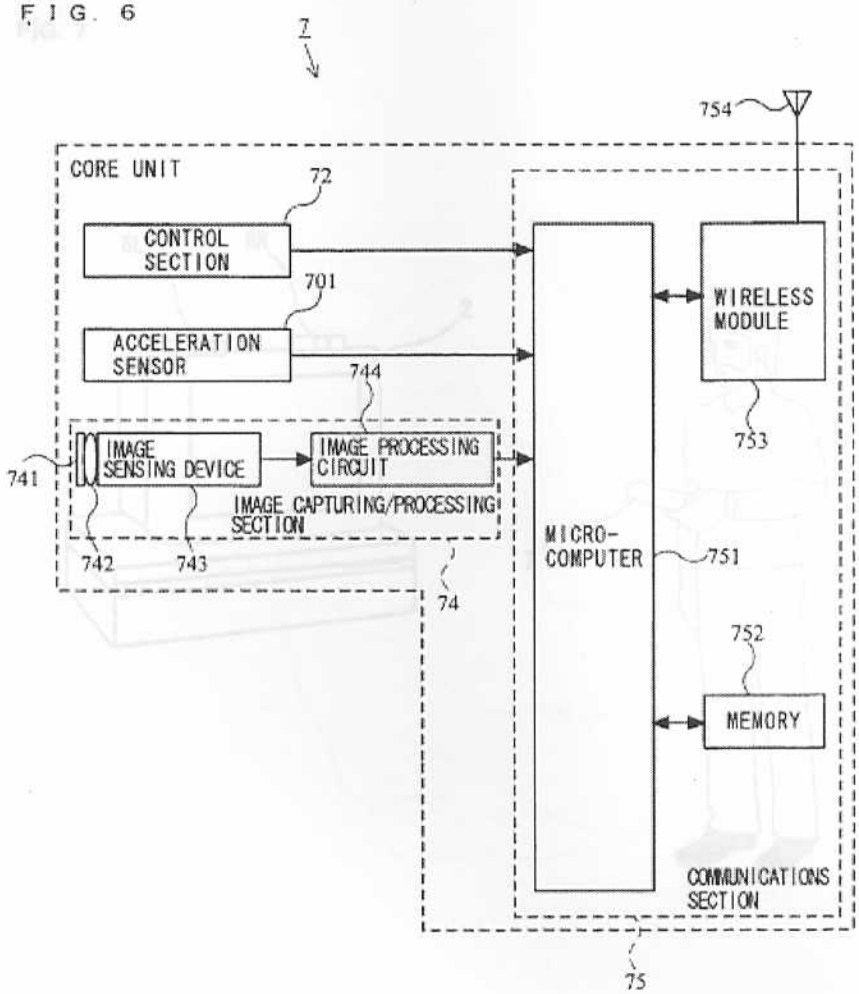
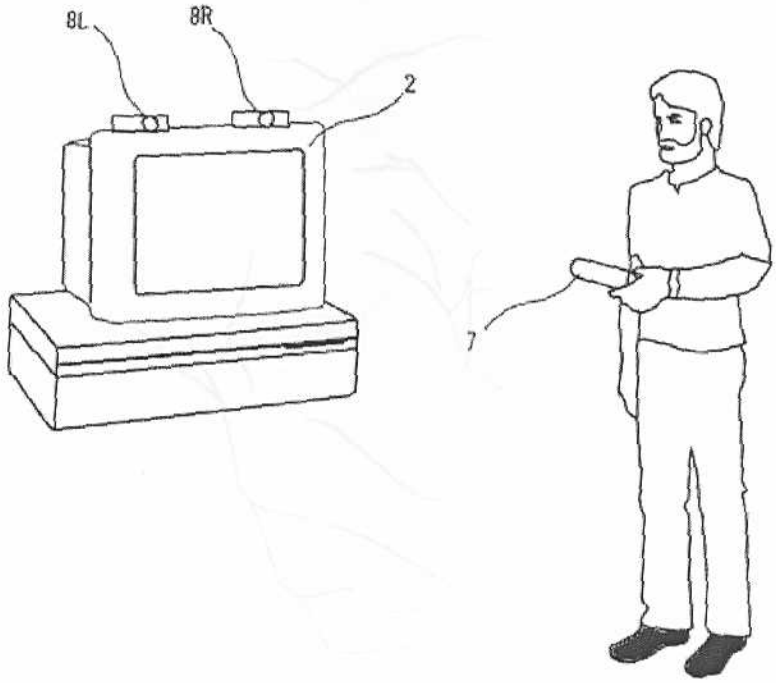

74
744
72i
741,742,743
705
700
704
71
73

FIG. 6


7
754
CORE UNIT
72
CONTROL SECTION
701
ACCELERATION SENSOR
744
IMAGE SENSING DEVICE
IMAGE PROCESSING CIRCUIT
741
IMAGE CAPTURING/PROCESSING SECTION
742
743
74
WIRELESS MODULE
753
MICRO-COMPUTER
751
752
MEMORY
COMMUNICATIONS SECTION
75

Patent Application Publication Mar. 15, 2007 Sheet 7 of 24 US 2007/0060384 A1

FIG. 7



Patent Application Publication Mar. 15, 2007 Sheet 8 of 24 US 2007/0060384 A1

FIG. 8

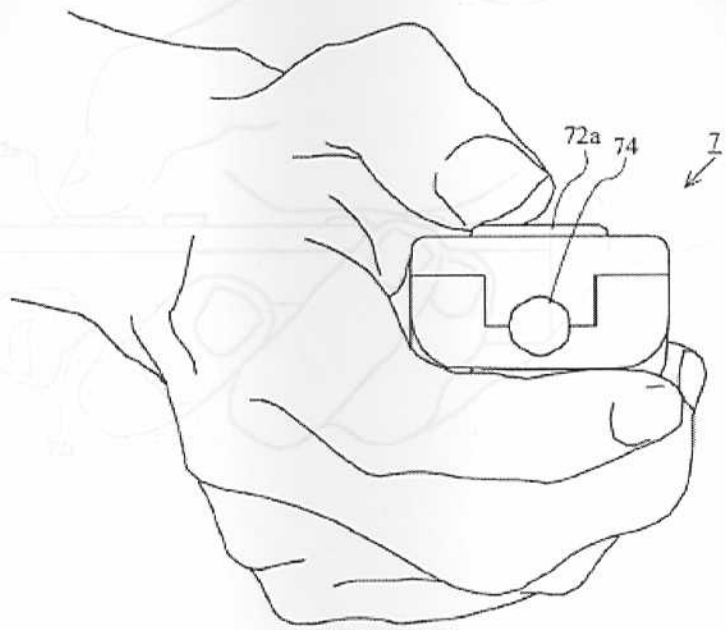


FIG. 9

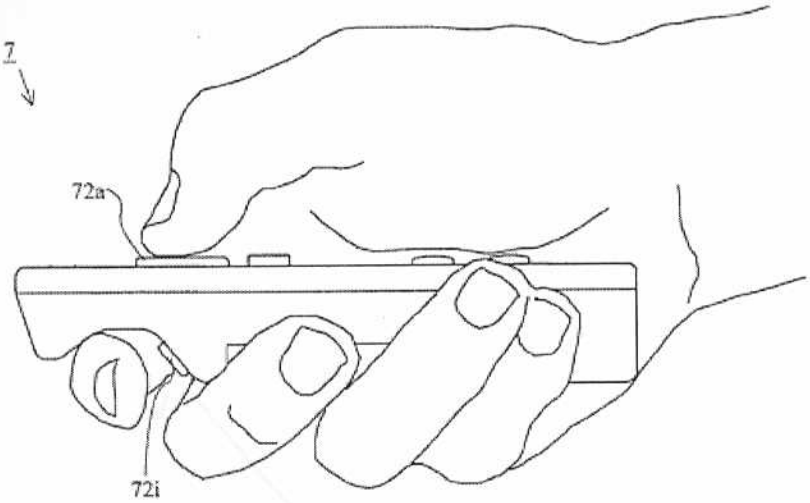
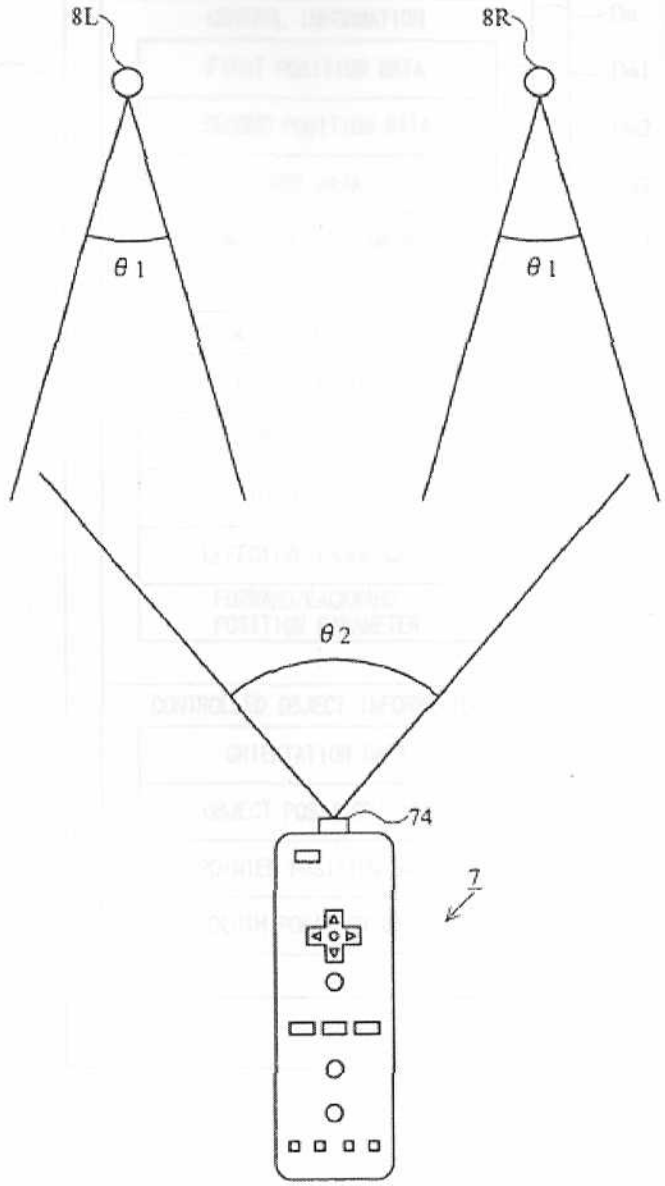


FIG. 10



Patent Application Publication Mar. 15, 2007 Sheet 11 of 24 US 2007/0060384 A1

FIG. 11

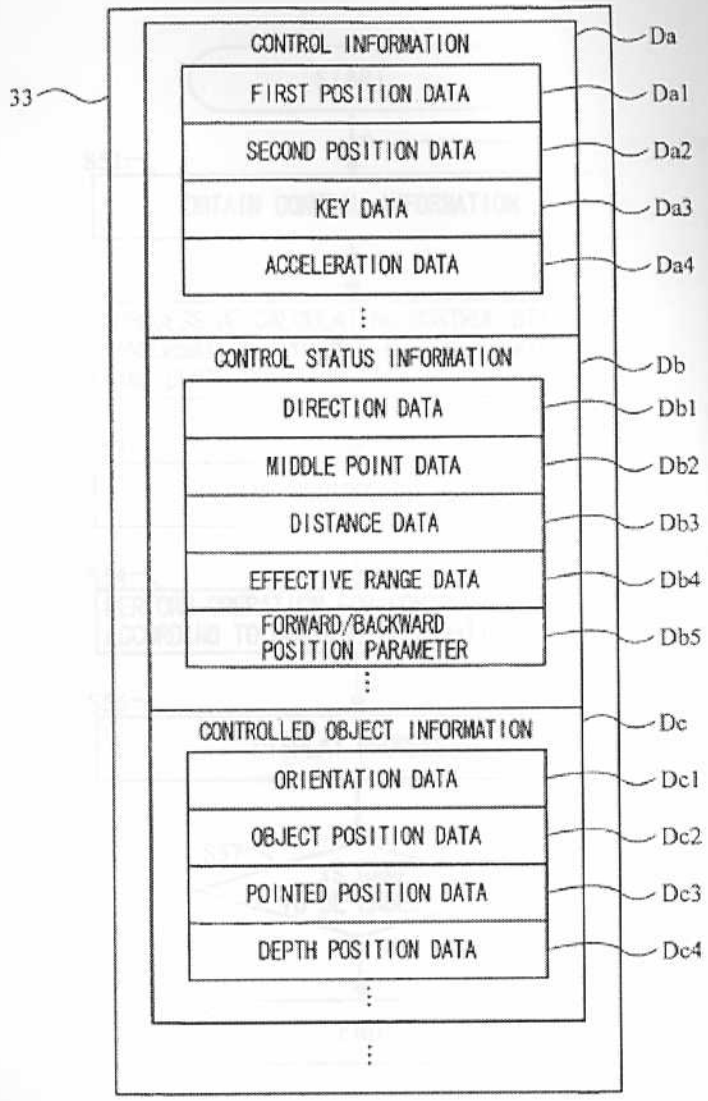
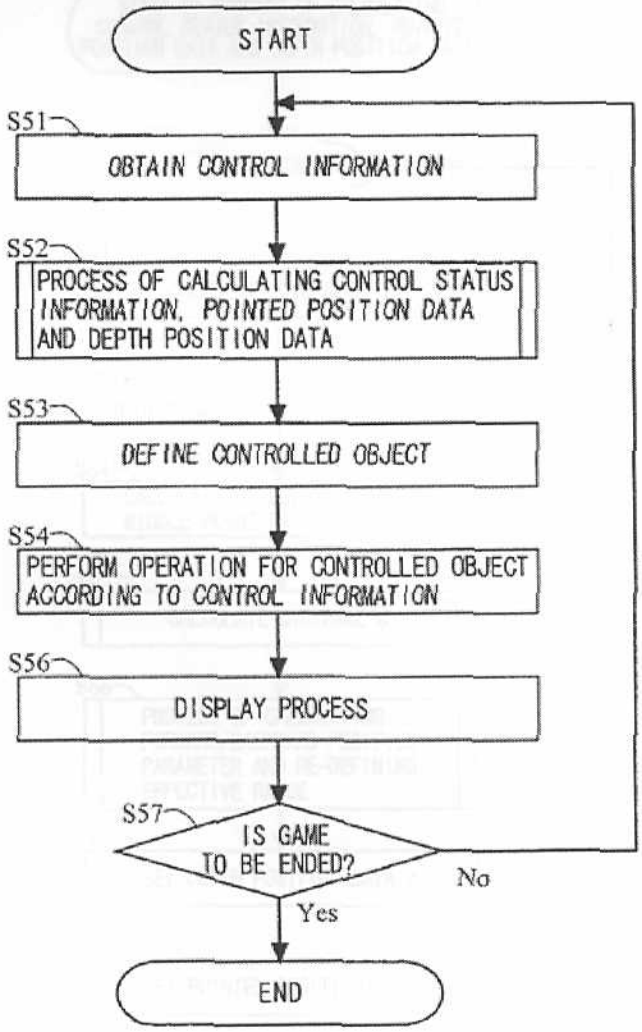
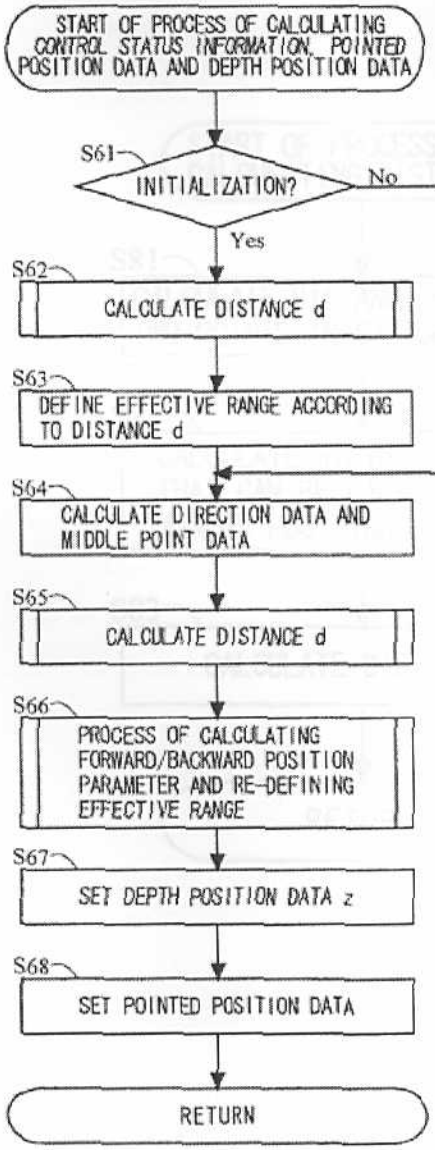


FIG. 12


START
S51
OBTAIN CONTROL INFORMATION
S52
PROCESS OF CALCULATING CONTROL STATUS INFORMATION, POINTED POSITION DATA AND DEPTH POSITION DATA
S53
DEFINE CONTROLLED OBJECT
S54
PERFORM OPERATION FOR CONTROLLED OBJECT ACCORDING TO CONTROL INFORMATION
S56
DISPLAY PROCESS
S57
IS GAME TO BE ENDED?
No
Yes
END

Patent Application Publication Mar. 15, 2007 Sheet 13 of 24 US 2007/0060384 A1

FIG. 13



Patent Application Publication Mar. 15, 2007 Sheet 14 of 24 US 2007/0060384 A1

FIG. 14

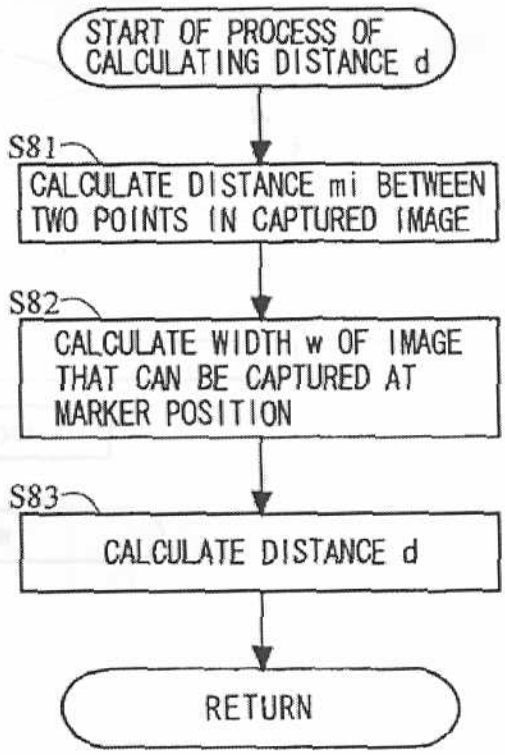


FIG. 15

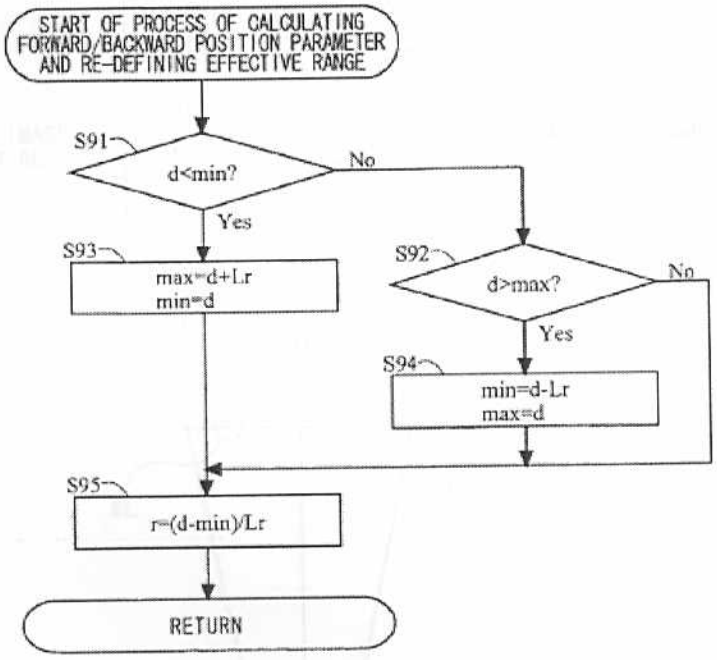

START OF PROCESS OF CALCULATING FORWARD/BACKWARD POSITION PARAMETER AND RE-DEFINING EFFECTIVE RANGE
S91
d<min?
No
Yes
S93
max=d+Lr
min=d
S92
d>max?
No
Yes
S94
min=d-Lr
max=d
S95
r=(d-min)/Lr
RETURN

FIG. 16

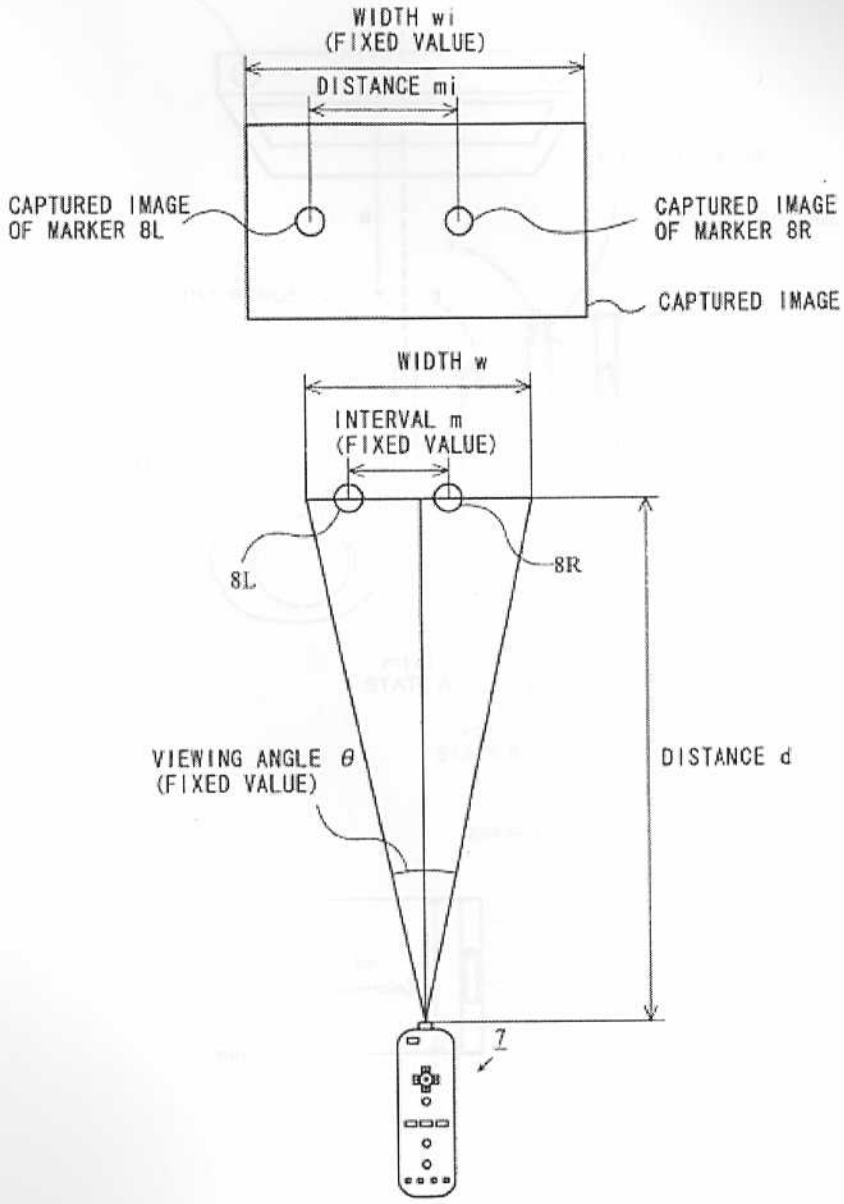

WIDTH wi
(FIXED VALUE)
DISTANCE mi
CAPTURED IMAGE
OF MARKER 8L
CAPTURED IMAGE
OF MARKER 8R
CAPTURED IMAGE
WIDTH w
INTERVAL m
(FIXED VALUE)
8L
8R
VIEWING ANGLE θ
(FIXED VALUE)
DISTANCE d
7

FIG. 17A

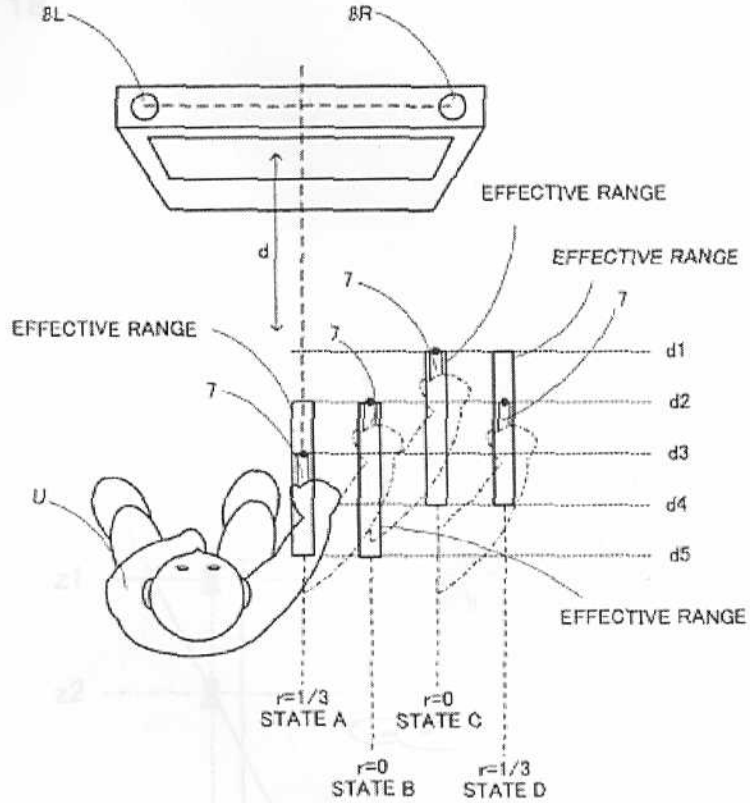
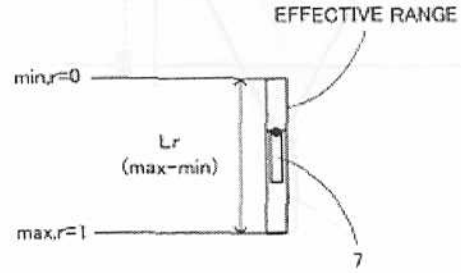
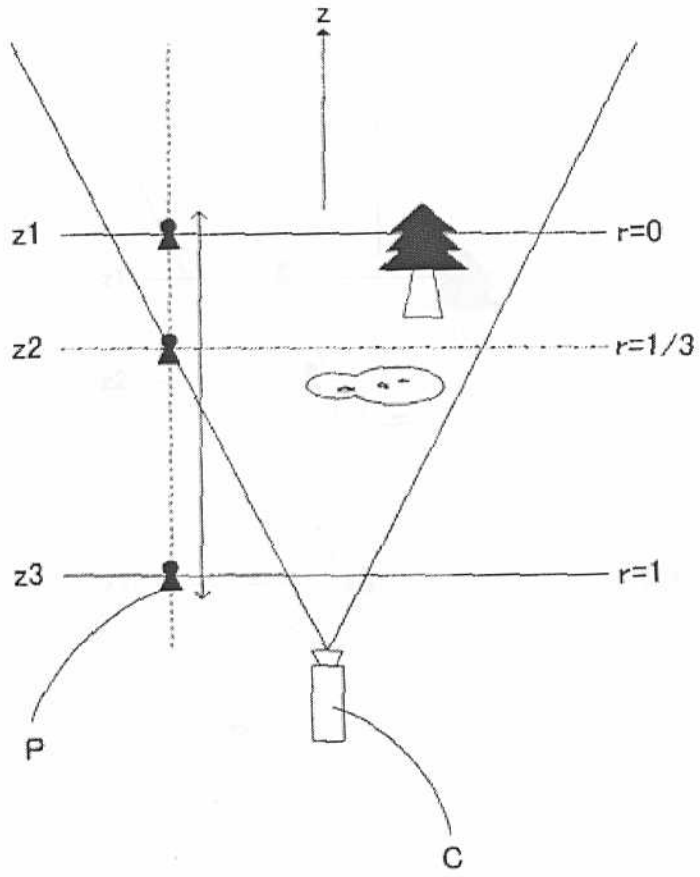


FIG. 17B



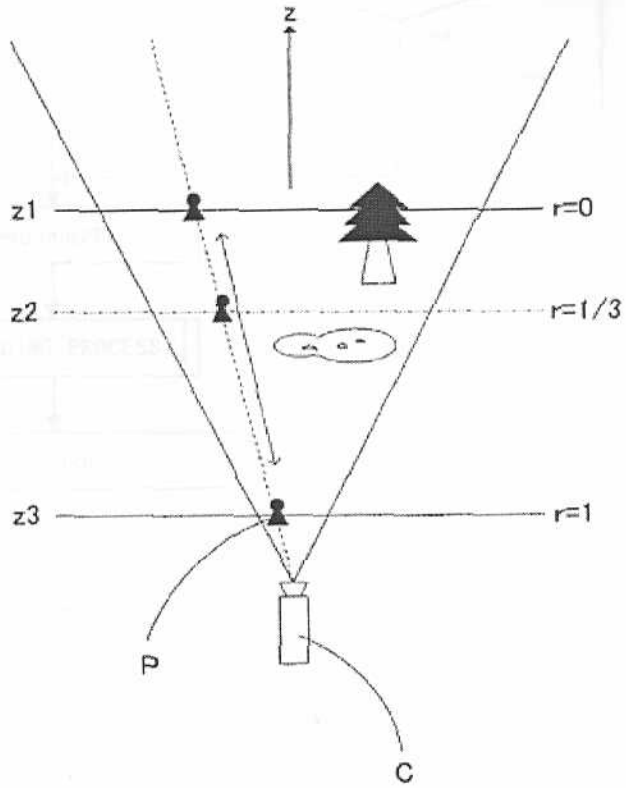
Patent Application Publication Mar. 15, 2007 Sheet 18 of 24 US 2007/0060384 A1

FIG. 18



Patent Application Publication Mar. 15, 2007 Sheet 19 of 24 US 2007/0060384 A1

FIG. 19



Patent Application Publication Mar. 15, 2007 Sheet 20 of 24 US 2007/0060384 A1

FIG. 20

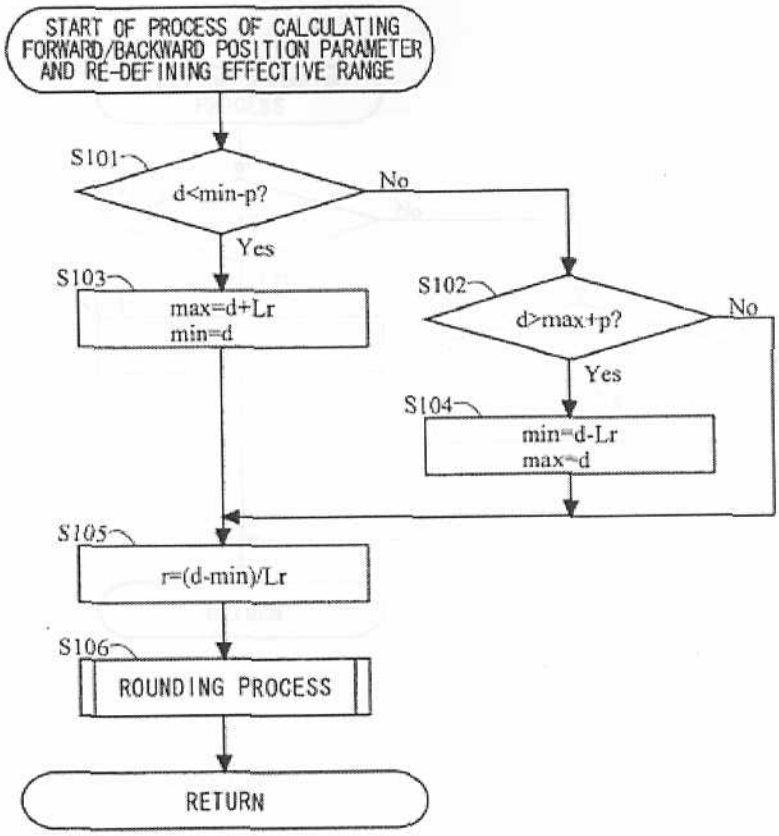


FIG. 21

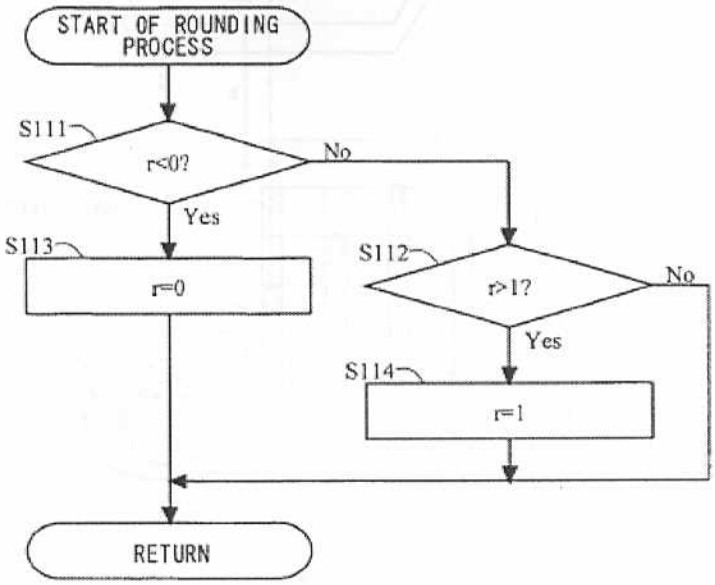

START OF ROUNDING PROCESS
S111
r<0?
No
Yes
S113
r=0
S112
r>1?
No
Yes
S114
r=1
RETURN

FIG. 22A

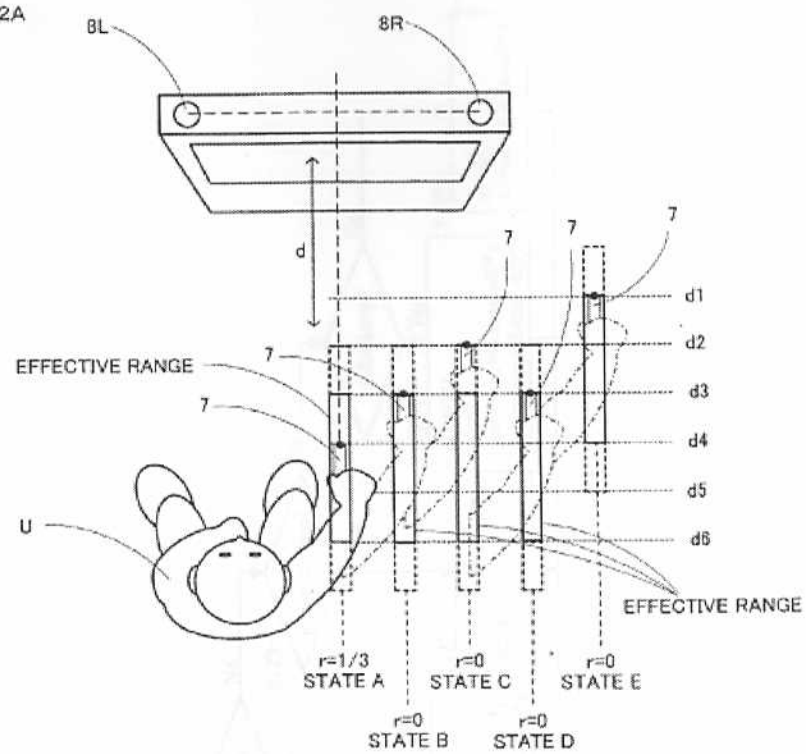


FIG. 22B

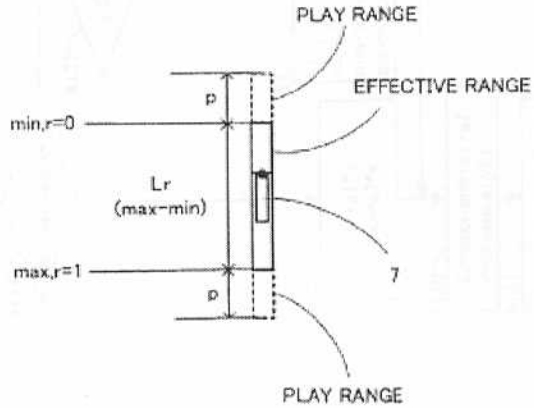


FIG. 23

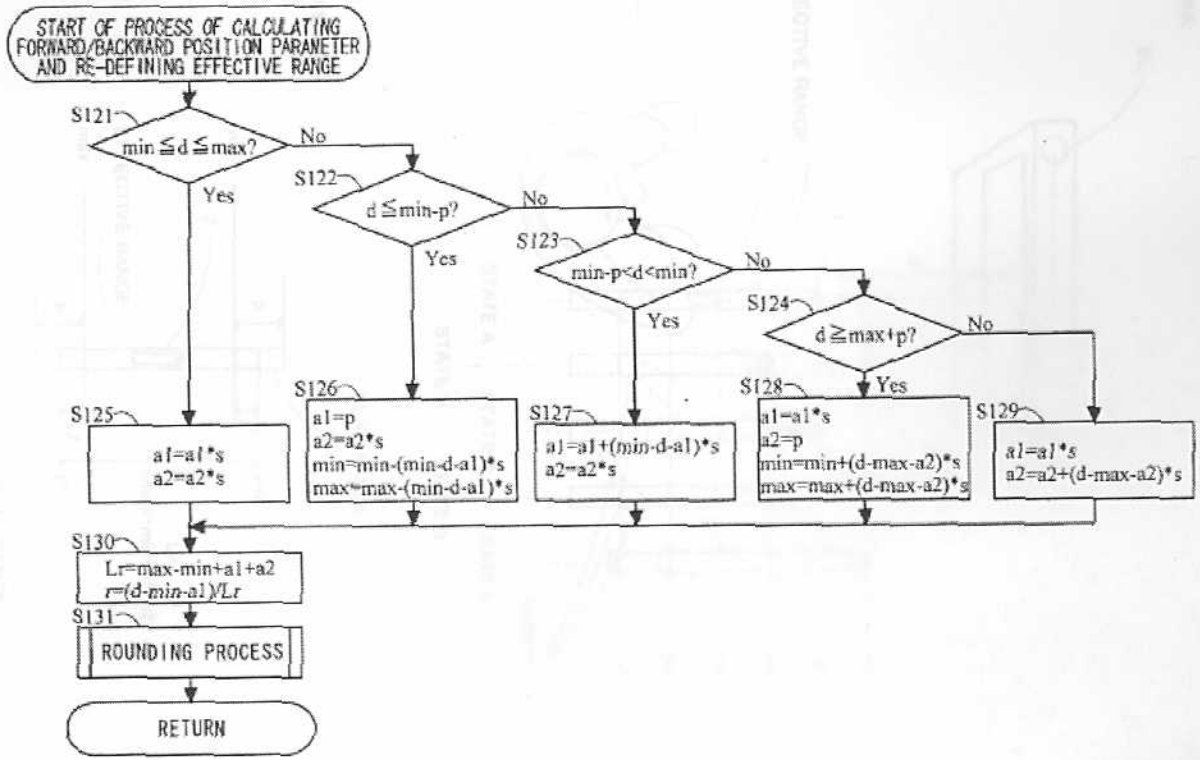

START OF PROCESS OF CALCULATING FORWARD/BACKWARD POSITION PARANETER AND RE-DEFINING EFFECTIVE RANGE
S121
min ≦ d ≦ max?
Yes
No
S122
d ≦ min-p?
Yes
No
S123
min-p<d<min?
Yes
No
S124
d ≧ max+p?
Yes
No
S125
a1=a1*s
a2=a2*s
S126
a1=p
a2=a2*s
min=min-(min-d-a1)*s
max=max-(min-d-a1)*s
S127
a1=a1+(min-d-a1)*s
a2=a2*s
S128
a1=a1*s
a2=p
min=min+(d-max-a2)*s
max=max+(d-max-a2)*s
S129
a1=a1*s
a2=a2+(d-max-a2)*s
S130
Lr=max-min+a1+a2
r=(d-min-a1)/Lr
S131
ROUNDING PROCESS
RETURN

FIG. 24A

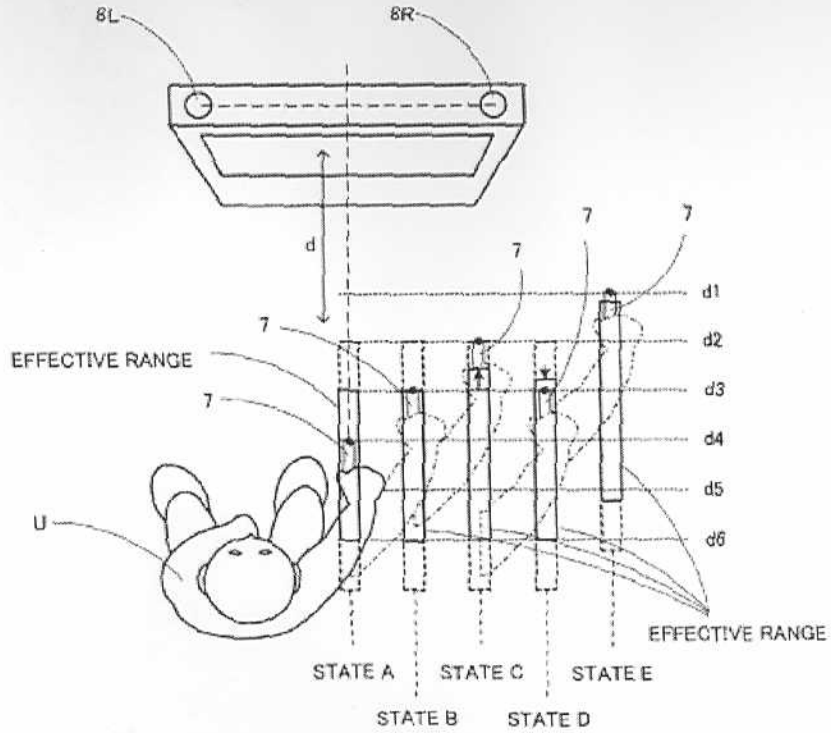
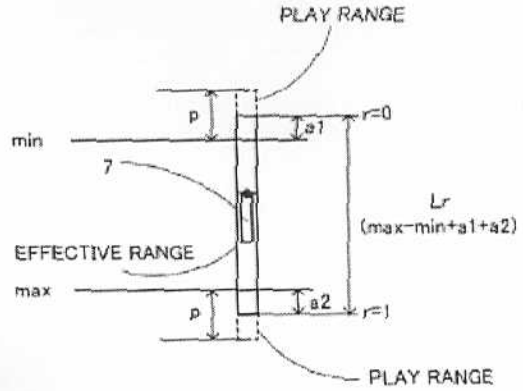



FIG. 24B