



**TECNOLÓGICO
DE MONTERREY®**

Campus Guadalajara

Escuela de Graduados en Ingeniería y Arquitectura – EGIA

Tesis

Visionless Interactions: Plataforma de aplicaciones para invidentes

para la obtención del grado de la

Maestría en Ciencias de la Computación

Autor:

Arturo Jafet Rodríguez Muñoz

Director: Dr. Luis Eduardo Falcón Morales

Sinodales: Dr. Gildardo Sánchez Ante

Dr. Jesús Arámburo Lizárraga

21 de Noviembre de 2012

Dedicatoria

*Quiero dedicar esta tesis a mi esposa quien es la inspiración que me impulsa a diario,
a mi hija por motivarme a ser mejor persona, esposo, padre y profesionalista cada día,
a mis padres que me siguen apoyando incondicionalmente desde mi nacimiento,
a mis hermanos y amigos por asegurarse de hacerme reír.*

Agradecimientos

Quisiera reconocer, antes que nada, el apoyo y paciencia de mi director de tesis, el Dr. Luis Eduardo Falcón. Además, quisiera agradecer a los otros miembros del comité supervisor de tesis por su tiempo y atención. Así también, le agradezco al Dr. Gildardo Sánchez por convencerme y motivarme a realizar esta maestría, al Instituto Jalisciense de la Juventud, a IBM y al ITESM Campus Guadalajara por creer en mí y otorgarme una beca para poder estudiar este posgrado.

Finalmente, le agradezco a mi esposa, hija, padres, hermanos y amigos por toda la paciencia que me han tenido durante este periodo de mi vida, ha sido una experiencia única, he aprendido bastante y me he tenido que esforzar como nunca, y jamás lo había logrado si no hubiera sido por su constante apoyo y cariño.

Resumen

En este documento se presenta el diseño, elaboración, resultados y trabajo futuro de la creación de *Visionless Interactions*. El programa tiene como finalidad el ayudar a personas con discapacidad visual a poder crear un mejor mapa mental de su ambiente a través del sentido del oído.

La aplicación funciona a más de 30 cuadros por segundo utilizando la configuración original de la máquina virtual de Java por lo que no se requiere una gran cantidad de memoria RAM. Los dispositivos que se usaron fueron una *Apple MacBook Pro 2009*, *Kinect* de *Microsoft*, Sonido 5.1 de *Logitech* y una tarjeta de sonido USB 7.1 de *Sabrent*.

Además, para poder confirmar que la plataforma cumple su propósito, se creó un juego sencillo en el cual el usuario tiene que interceptar abejas que van volando de adelante hacia atrás. El juego está dividido en 11 niveles y se realizaron pruebas de campo con un grupo de 13 voluntarios, los participantes tienen desde 4 hasta 63 años, 62% mujeres y 38% hombres, la mitad con discapacidad visual y la otra mitad utilizó un antifaz. En las pruebas que se llevaron a cabo se pudo observar que el usuario se divirtió y con el paso del tiempo fue logrando identificar de mejor manera los objetos 3D de su alrededor. Además, el juego funcionó de manera adecuada al permitir que el jugador mostrara cada vez una mayor familiaridad al navegar tanto entre menús, como al interactuar con los objetos 3D.

En conclusión, fue posible desarrollar una aplicación en *Java* que ayude a las personas visualmente discapacitadas, parcial o totalmente, a navegar mejor en su entorno mientras se divierten. El mercado para este tipo de aplicaciones sigue siendo una gran oportunidad y *Visionless Interactions* permite acortar la brecha tecnológica existente, al proporcionar una plataforma que facilita la creación de videojuegos para múltiples sistemas operativos sin necesidad de una pantalla.

Contenido

Dedicatoria	I
Agradecimientos	II
Resumen	III
Lista de Tablas	VIII
Lista de Figuras	X
1. Introducción	1
1.1. Antecedentes	1
1.2. Definición del problema	2
1.3. Objetivos	3
1.4. Justificación	4
1.5. Hipótesis	5
1.6. Metodología	6
1.7. Trabajos afines	6
2. Marco Teórico	8
2.1. Procesamiento digital de imágenes	8
2.1.1. Eliminación del fondo	8
2.1.2. Operaciones morfológicas	9
2.2. Sonido	11
2.2.1. Binaural	12
2.2.2. Envoltente 5.1	13

2.3. Kinect	13
2.4. Generación de la imagen	14
2.5. Optimización	16
2.5.1. Orientado a objetos	16
2.5.2. Secuencias de datos	16
2.5.3. Hilos	17
3. Desarrollo	18
3.1. Diseño general	18
3.2. Módulo <i>Kinect</i>	20
3.2.1. Inicialización y retroalimentación	20
3.2.2. Traducir los datos del sensor de profundidad	21
3.2.3. Aislar el fondo	23
3.2.4. Restricciones	27
3.2.5. Interacción con el usuario	27
3.3. Módulo Sonido	29
3.3.1. Sistema de sonido y posicionamiento	30
3.3.2. Sonido en 3D	33
3.3.3. Retroalimentación	35
3.4. Módulo Mundo Virtual	36
3.4.1. Escena y abejas	37
3.4.2. Relación coordenadas virtuales 3D a pixeles	38
3.5. Módulo Juego	39
3.5.1. Configuración del entorno físico	39
3.5.2. Menús	41
3.5.3. Niveles	43
3.5.4. Proceso del juego	52
4. Experimentación	54
4.1. Diseño de las pruebas	54
4.1.1. Cuestionario	55

4.1.2. Ambiente	56
4.1.3. Recopilación automática de los datos	57
4.2. Resultados obtenidos	60
4.2.1. Interpretación	64
5. Conclusiones	66
6. Trabajo futuro	69
Bibliografía	71
A. Imágenes	75
B. Sonidos usados	81

Lista de Tablas

2.1. Elemento estructural binario de tamaño 5x5.	10
3.1. Definición y unidad de las variables utilizadas para la traducción de datos del sensor de profundidad.	22
3.2. Elemento estructural binario de tamaño 3x3 en forma de rectángulo.	24
3.3. Elemento estructural binario de tamaño 7x7 en forma de cruz.	26
3.4. Lista de eventos que requieren reproducir audio para dar una retroalimentación adecuada al usuario.	36
3.5. Descripción del nivel 1.	45
3.6. Descripción del nivel 2.	45
3.7. Descripción del nivel 3.	46
3.8. Descripción del nivel 4.	47
3.9. Descripción del nivel 5.	48
3.10. Descripción del nivel 6.	48
3.11. Descripción del nivel 7.	49
3.12. Descripción del nivel 8.	50
3.13. Descripción del nivel 9.	51
3.14. Descripción del nivel 10.	51
3.15. Descripción del nivel 11.	52
3.16. Cantidad mínima de abejas que se deben interceptar para pasar cada nivel.	53
4.1. Especificación técnica del <i>hardware</i> usado en las pruebas.	56
4.2. Especificación técnica del <i>software</i> usado en las pruebas.	57
4.3. Número máximo de abejas interceptadas en la mejor ocasión por cada uno de los 13 participantes en cada uno de los 11 niveles.	60
4.4. Respuestas al cuestionario dadas por cada uno de los 13 participantes del estudio.	61

4.5. Datos estadísticos del número máximo de abejas interceptadas en la mejor ocasión por cada nivel y considerando a todos los usuarios que participaron en el estudio.	62
4.6. Datos estadísticos generales obtenidos durante un tiempo de juego de 15 minutos por parte de cada uno de los participantes.	62
4.7. Datos estadísticos generales dividido en hombres y mujeres.	63
4.8. Datos estadísticos generales de invidentes versus videntes.	63
4.9. Datos estadísticos del cuestionario de las preguntas 1 a 8.	64
4.10. Datos estadísticos del cuestionario de las preguntas 9 a 15.	64

Lista de Figuras

2.1. Demostración de una imagen antes y después de eliminar el fondo y binarizarla.	9
2.2. Demostración de una imagen antes y después de realizar la operación de erosión y dilatación utilizando el elemento estructural de la tabla 2.1. . . .	10
2.3. Ilustración del uso de sonido binaural y envolvente.	12
2.4. Elementos del Kinect.	14
2.5. Mapeo de la profundidad capturada a una imagen en RGB y en escala de grises.	15
3.1. Ilustración de la arquitectura de la aplicación.	20
3.2. Imagen antes y después de ser procesada con las operaciones morfológicas para obtener la máscara.	27
3.3. Ilustración de las regiones de interacción de la aplicación.	28
3.4. Ilustración del sistema de menús empleado en la aplicación.	29
3.5. Imagen de la tarjeta de sonido externa Sabrent USB-SND8 usada para poder conectar la computadora a un sistema 5.1.	31
3.6. Imagen del sistema de sonido Logitech z906 usado en la aplicación.	32
3.7. Imagen de las entradas del sistema de sonido Logitech z906 y como conectar la tarjeta de sonido externa Sabrent USB-SND8 al sistema.	32
3.8. Ilustración del posicionamiento correcto del sistema de sonido y del <i>Kinect</i>	33
3.9. Imagen de la configuración correcta del sonido en Mac OS X.	34
3.10. Ilustración del ajuste al volumen y tono para mejorar la percepción de la profundidad y altura.	35
3.11. Imagen del escenario.	37
3.12. Imagen del escenario con su sistema de coordenadas.	38
3.13. Ilustración de como debe quedar el entorno una vez configurado.	40
3.14. Imagen del escenario real con el entorno ya configurado.	41

3.15. Representación del Menú Principal.	41
3.16. Representación del Menú Pausa.	42
3.17. Representación del Menú Confirmación.	42
3.18. Representación de los tipos de rutas.	44
4.1. Imagen de las pantallas de inicio.	58
4.2. Imágenes de la ventana del juego.	58
4.3. Imagen de las 3 ventanas durante la ejecución del programa.	59
4.4. Imagen de ejemplo de un archivo de log guardado automáticamente por la ventana de la consola.	59

CAPÍTULO 1

Introducción

1.1 Antecedentes

La vista es sin duda uno de los sentidos que más desarrollan los humanos y en el cual se suelen apoyar más para percibir su ambiente. En el estudio realizado por parte del INEGI en México [1] existen un total de 467,040 mexicanos que sufren de discapacidad visual. De este número el 4.8 % es por prevalencia, 11.2 % de nacimiento, 33 % de enfermedad y 49.6 % por otra causa. Además la OMS¹ estima 45 millones de personas ciegas y aproximadamente 135 millones con discapacidad visual parcial [1, p. 97]. Es importante destacar que la OMS considera discapacidad visual cuando la visión es menor a 20/400 [2] tomando en cuenta el ojo con mejor capacidad y con la mejor corrección, esto significa que la persona ve a 20 pies lo que una persona normal puede ver a 400 pies. [1, p. 97].

Es posible que sin saber estos datos no se resalten todos los retos que se tienen a superar debido a esta condición. Las personas que sufren de este tipo de discapacidad aprenden a reemplazar la información obtenida por los ojos usando otros sentidos como el del tacto y el oído. Estudios recientes [3, 4] han demostrado que el humano puede usar ecolocación para navegar en su entorno sin tropezar con obstáculos. Por lo tanto se puede aprovechar la tecnología actual para facilitar este aprendizaje y al mismo tiempo divertirse.

¹Organización Mundial de la Salud

A finales del 2010 la empresa Microsoft comenzó la venta de un dispositivo llamado *Kinect*[5], el cual permite al usuario interactuar con un sistema sin necesidad de un control. Para promover el desarrollo de videojuegos utilizando este dispositivo, Microsoft ha creado varios de ellos, entre los que destacan *Kinect Sports*[6] y *Kinect Adventures*[7] para el *Xbox 360*[8]. *Kinect Sports* permite elegir entre una serie de deportes como fútbol, donde el usuario debe físicamente patear en el aire para poder anotar gol en el juego. Por otro lado, *Kinect Adventures* tiene un set de actividades entre las cuales está el conducir virtualmente una balsa hasta la meta a través de equilibrio y saltos del jugador.

A partir de su lanzamiento, comenzaron a mostrarse vídeos de aplicaciones del *Kinect* en una computadora. En el sitio *Kinect Hacks*[9] tienen una gran cantidad de vídeos de miembros utilizando el *Kinect* para distintos fines. Uno de los más concurridos fue la aplicación *Body Dysmorphic Disorder*[10], el cuál es una alteración en tiempo real de una persona frente a la cámara, transformando su cuerpo en una serie de burbujas. Existen muchos ejemplos de lo que se ha logrado con el *Kinect* y en su mayoría hacen uso de movimientos del cuerpo, reconocimiento de la silueta de la mano y posicionamiento de las extremidades.

1.2 Definición del problema

En la actualidad, la ecolocación es una de las técnicas más usadas para entrenar personas con discapacidad visual a poder navegar en un entorno. El problema con esta técnica es que requiere paciencia y tiempo para poder lograr desplazarse en un ambiente sin lastimarse. Debido a lo anterior nace la necesidad de ayudar a facilitar este proceso haciéndolo divertido, sencillo y sin la necesidad de tener que ir a un Centro de Rehabilitación que cuente con este equipo.

Con la aparición del *Kinect* de Microsoft, una nueva era de aplicaciones comenzó. Existen una gran cantidad de programas con diversos fines realizados por investigadores,

estudiantes y personas interesadas. Microsoft tiene algunos ejemplos de cómo distintos centros de rehabilitación, hospitales y escuelas están usando el *Kinect* como herramienta para acelerar el aprendizaje [11]. En su inicio, Microsoft decidió sólo proporcionar el dispositivo sin acceso a un API². Sin embargo, una comunidad de personas, interesadas en explorar el uso de este dispositivo, creó *OpenKinect*[12].

Por lo tanto, para hacer uso de la ecolocación, se necesita una herramienta que facilite el proceso de identificación de obstáculos a través del sonido. La herramienta no debe necesitar infraestructura especializada como una clínica de rehabilitación. Además la interacción deberá ser sencilla, intuitiva y tener retroalimentación, considerando que el usuario no puede usar la vista. Por último, los componentes a usar deberán ser fáciles de obtener aprovechando los dispositivos que ya existen en el mercado.

1.3 Objetivos

Objetivo General:

Diseñar y desarrollar una plataforma de aplicaciones para invidentes sin el uso de marcadores o controles y utilizando herramientas de *software* y *hardware* de fácil acceso en el mercado.

Objetivos Particulares:

- Crear un programa interactivo el cual facilite el proceso de identificación de obstáculos 3D virtuales a través del sonido.
- El programa deberá ser independiente de la iluminación de la zona de juego.
- Realizar un diseño que permita recabar datos de video y dar retroalimentación en tiempo real mediante el uso de dispositivos fáciles de obtener en el mercado.
- No utilizar controles o marcadores visuales para la retroalimentación de los datos.

²Interfaz de programación de aplicaciones

- La aplicación deberá funcionar en múltiples sistemas operativos y en equipos de cómputo de fácil acceso al público.
- Hacer uso de *software* libre.
- El manejo del menú y la ejecución de los comandos por parte del usuario deberán ser lo más sencilla y efectiva posible.
- Diseñar e implementar un juego que demuestre como se usa la plataforma.

1.4 Justificación

Como se mencionó en las secciones anteriores, existe una necesidad por una herramienta que agilice la curva de aprendizaje de las personas con discapacidad visual para que puedan percibir mejor su ambiente. Es alarmante la cantidad de personas que requieren aprender nuevas maneras de navegar para seguir siendo independientes. Asimismo, existen casos de personas que viven solas y que por ejemplo no pueden dirigirse hacia un centro de rehabilitación por ellos mismos, por lo que siempre están requiriendo la ayuda de terceros. Por lo tanto, sería conveniente contar con un instrumento de fácil acceso y operación dentro del hogar, que sustituya la información proporcionada por la vista utilizando otros sentidos.

En el pasado se han utilizado cámaras de video, cámaras infrarrojas y técnicas de procesamiento de imágenes para poder reconocer a un usuario, pero para ello se requiere tener un ambiente controlado. Uno de los problemas más comunes al usar este enfoque, es la existencia de variaciones de luz en el lugar donde se está capturando la imagen. Debido a lo anterior, existen aplicaciones que solicitan al usuario asegurarse de que el alumbrado sea constante y sin alteraciones, pero difícilmente una persona con discapacidad visual podrá determinar si la iluminación cambia mientras hace uso de la herramienta. También se han creado aparatos con dispositivos de video, infrarrojo y láser para poder tener más información del ambiente, pero son hechos a la medida y difíciles de obtener, además de

tener un alto costo.

Actualmente, con la aparición del *Kinect* en el mercado y el uso de herramientas de procesamiento de imágenes, se pueden realizar aplicaciones de alta precisión que identifiquen una silueta humana o bien, que ayuden a describir el entorno tridimensional del usuario. El *Kinect* es un dispositivo fácil de adquirir por su bajo costo y existen una variedad de SDK que permiten tomar ventaja de ello. Originalmente, Microsoft sólo creó un kit de desarrollo para XBOX, pero varios grupos de personas alrededor del mundo vieron el potencial del *Kinect* en computadoras y comenzaron el desarrollo de APIs no oficiales. A causa del gran éxito de los programas realizados con estas librerías, Microsoft vio la oportunidad de apoyar a programadores con un SDK³ que provee más funcionalidad que las opciones de código abierto, para aumentar la cantidad de software utilizando su tecnología e incluso adquirir nuevas ideas para ellos mismos. De aquí que se propusiera crear una aplicación que se use en casa y la cual requiera solamente de una computadora, un *Kinect* y un sistema de sonido 5.1.

1.5 Hipótesis

Se está suponiendo que se puede crear una aplicación en Java la cual funcione en tiempo real tomando datos del *Kinect* y dando retroalimentación por medio de un sistema de sonido 5.1. También se supone que este programa no necesitará de un poder computacional mayor al que se puede encontrar en computadoras actuales que no son especializadas en alto rendimiento. Asimismo se supone que la mejor forma de entretener y ayudar en su aprendizaje al usuario, es a través de un videojuego.

Por otro lado se busca saber si es posible que una persona pueda reconocer, con un bajo margen de error, el origen de un sonido en 3D. En otras palabras, se busca determinar si el usuario puede localizar y moverse hacia la posición en el espacio en el cual se está simulando la existencia de una fuente de sonido. También se busca conocer si se puede

³Kit para desarrollar software

tener un ambiente no controlado y aislar al usuario del fondo que lo rodea con algoritmos conocidos de procesamiento de imágenes y el uso del *Kinect*.

1.6 Metodología

El primer paso consistirá en investigar las soluciones que han dado otros investigadores en el mundo a este problema. Posteriormente se continuará con el proceso de aprendizaje que permita integrar en Java las librerías que se requieren en el proceso, a saber, *OpenKinect*, *OpenGL* y *OpenCV*. A continuación se investigará qué librerías existentes permiten generar sonido envolvente y que además se puedan comunicar con Java. En este punto se deberán comparar y entender las ventajas y desventajas sobre trabajar con sonido binaural y audífonos o bien, con un sonido envolvente y bocinas 5.1. Asimismo, se hará el diseño y desarrollo de un menú interactivo que le permita al usuario invidente seleccionar las opciones de su preferencia, guiándolo durante todo el proceso de una manera lo más sencilla y amigable posible. Finalmente, se seleccionará a un grupo de personas con diferentes tipos de impedimento visual, reales o simulados, para que interactúen y jueguen con el programa desarrollado para obtener su posterior retroalimentación sobre las distintas etapas y procesos de uso del mismo. Con base a dichos comentarios se llevarán a cabo los cambios y ajustes necesarios que permitan mejorarlo.

1.7 Trabajos afines

Diferentes enfoques han sido explorados por diversos investigadores para resolver problemas parecidos al anteriormente planteado. A continuación se mencionan algunos de los trabajos encontrados durante la realización de esta tesis y que sirvieron para dar dirección.

En el 2008 investigadores de la Universidad de Macedonia propusieron un método de visión computacional el cuál permitía obtener información sobre la posición de los dedos de un pianista[13], y que en lugar de usar un *DataGlove*[14], propusieron usar una cámara de bajo presupuesto, logrando reconocer la posición de los dedos del pianista antes y al

tocar las teclas blancas. Una de las limitaciones de la investigación fue que solo se probó con una persona y en un ambiente altamente controlado.

En el 2009 científicos de la Universidad de Tokio desarrollaron un sistema capaz de dar seguimiento a la mano y sus dedos usando una sola cámara especial que pudiera grabar a una alta cantidad de cuadros por segundo [15]. Al final de su investigación su sistema logra detectar en tiempo real los movimientos realizados con la mano que les permite teclear en el aire. Sin embargo, cabe mencionar que para ello requirieron hacer uso de una cámara especial, además de acelerar el procesamiento mediante el uso de un coprocesador *FPGA*⁴[16].

A finales del 2010 la Universidad de Chile a través de su Centro Avanzado de Investigación para la Educación trabajó en la creación de ambientes virtuales hápticos y sonoros los cuales pudieran ser navegados por personas ciegas en forma de videojuego[17]. Los resultados demostraron que los usuarios pudieron identificar formas geométricas básicas como triángulos y círculos.

En el 2011 investigadores y alumnos de la Universidad Konstanz en Alemania crearon un sistema para facilitar la navegación de personas discapacitadas visualmente a través de un cinturón vibro-táctil, marcadores ópticos y el *Kinect*[18]. Utilizando hardware y software de bajo costo se diseñó un casco que sostiene un *Kinect*, y el cual proporciona información sobre la distancia a la cual se encuentran obstáculos tridimensionales, además de reconocer marcadores visuales que brindan información al usuario sobre su ubicación espacial. Asimismo, diseñaron un cinturón fabricado de placas vibratoras que genera diferentes tipos oscilaciones dependiendo del ángulo y cercanía de los objetos. Los resultados obtenidos fueron bastante positivos, además de ser bien recibido por parte de los usuarios, mostrando que el *Kinect* puede llegar a ser una gran herramienta de usos muy diversos.

⁴Field-Programmable Gate Array

CAPÍTULO 2

Marco Teórico

2.1 Procesamiento digital de imágenes

Para poder obtener información valiosa de una imagen se requieren ciertos ajustes. El procesamiento digital de imágenes[19, p. 15-39] es una herramienta con diversas tácticas para mejorar los datos y hacer uso de ellos. Un ejemplo muy citado es el del conteo de dedos, el problema indica que el usuario en un ambiente controlado mostrará su mano completa y con los dedos apuntando hacia arriba, el programa tiene que decir cuantos dedos se están mostrando. A primera vista podrá parecer que no tiene ninguna aplicación en la vida real, pero este ejemplo puede usarse para interacción de menús u obtener más datos acerca del tamaño, distancia y rotación de los dedos.

2.1.1 Eliminación del fondo

Una de las aplicaciones más comunes del procesamiento digital de imágenes es la eliminación del fondo, el cual consiste en guardar una serie de fotografías para después obtener la imagen promedio[19, p. 112-116] y las nuevas capturas son restadas[19, p. 110-112] contra el promedio, hasta un umbral determinado por la desviación estándar de dichas imágenes. Todos los valores que no quedaron eliminados son los que seguirán manteniendo su valor original, y así es como se aísla el fondo no deseado de los nuevos elementos que se desean analizar. Este proceso ha sido usado en algunos programas para video conferencias al reemplazar el ambiente del usuario y dar una mayor privacidad de su entorno.

Los resultados obtenidos con dicho procedimiento son susceptibles a errores y sus causas pueden tener su origen en las variaciones de luz, en el mismo software interno que utilizan algunas cámaras para ajustar el brillo automáticamente y en otros factores que pueden afectar el brillo y contraste de la imagen capturada. Por lo general se emplea la ecualización de histogramas[19, p. 91] como un primer paso para reducir dichas variaciones indeseadas. El proceso de binarización, es decir, el transformar una imagen a pixeles con valores blancos o negros, es otro método muy utilizado para el análisis de la información obtenida en una imagen. Para ello se hará uso de uno o varios valores umbral, elegidos por cierto generalmente de manera empírica e ilustrado en la figura 2.1.

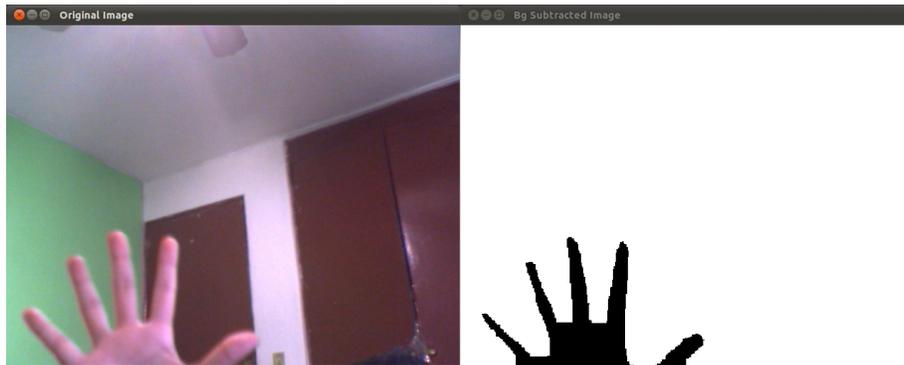


Figura 2.1: Demostración de una imagen antes y después de eliminar el fondo y binarizarla.

2.1.2 Operaciones morfológicas

Como su nombre lo indica, las operaciones morfológicas[19, p. 519-560] están enfocadas en la forma y estructura de las imágenes, o mejor dicho, en extraer componentes útiles de ellas. Estas operaciones funcionan pixel por pixel realizando diversos cálculos en los pixeles vecinos para determinar su valor numérico y determinando así su color o tonalidad final. La información obtenida puede ser usada para determinar qué se está observando y aislar elementos simples que a primera vista son complejos de analizar, por mencionar un ejemplo.

Dilatación y erosión son operaciones primitivas, a partir de las cuales serán definidas otras más. Se comienza creando un elemento estructural binario[19, p. 525] como se ilustra

en la tabla 2.1.

0	0	1	0	0
0	0	1	0	0
1	1	1	1	1
0	0	1	0	0
0	0	1	0	0

Tabla 2.1: Elemento estructural binario de tamaño 5x5.

Después se recorre la imagen pixel por pixel y ahí se coloca el centro del elemento. Se recorren todos los pixeles incluidos en dicho elemento estructural para asignar un valor resultante al pixel central de la estructura, que en el caso de dilatación, es el valor máximo encontrado, en el caso de la erosión será el valor mínimo. Por lo tanto la dilatación ayuda a unir elementos que estaban ligeramente separados, mientras la erosión separa elementos que no estaban fuertemente conectados.

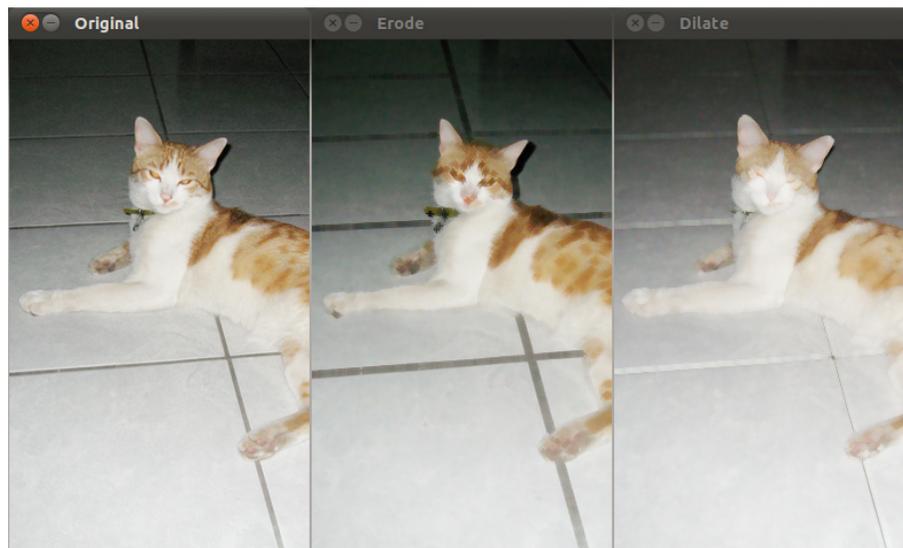


Figura 2.2: Demostración de una imagen antes y después de realizar la operación de erosión y dilatación utilizando el elemento estructural de la tabla 2.1.

Es evidente que el tamaño del elemento estructural utilizado impactará directamen-

te en el número de operaciones a realizar en cada pixel. Por ejemplo, si utilizamos un elemento estructural de 5 filas y 5 columnas en una imagen de 640 por 480, nos darían $(5)(5)(640)(480) = (25)(307,200) = 7'680,000$ cálculos a realizar por cada cuadro capturado y procesado. Es por ello que la elección de un tamaño adecuado del elemento estructural es crítico en el análisis de un proceso. Estos métodos por otro lado, serán de mucha ayuda para remover basura y mejorar la estructura final de la fotografía, en particular cuando se desea combinarlas con otras imágenes.

2.2 Sonido

El sonido en una aplicación de escritorio es común que sea usado para notificaciones, instrucciones, música de fondo o vídeos. En el caso específico de un videojuego el sonido es esencial, ya que ayuda a aumentar la inmersión del jugador. Las dos opciones más utilizadas para la reproducción del sonido es a través de un par de bocinas conectadas a la computadora, o bien, mediante el uso de audífonos. Poco a poco ha aumentado la cantidad de jugadores que desean una experiencia más real al jugar, por lo que constantemente están en la búsqueda de audífonos o bocinas más avanzadas.

Los audífonos y altavoces más básicos constan de dos bocinas, pero con el paso del tiempo se han creado productos más especializados. Los audífonos con cancelación de ruido permiten enfocarse completamente en el audio y no ser distraído por los ruidos del ambiente, emplean un micrófono para recibir el audio del exterior y mandar la onda contraria. Debido a que únicamente se tienen dos fuentes de sonido, al usuario le puede costar mucho trabajo el sentirse dentro del espacio virtual. Para solucionar lo anterior, se han creado audífonos y altavoces con cinco bocinas y un subwoofer. Estos sistemas han tenido un gran auge como entretenimiento en casa, de esta manera un grupo de personas se puede sentar frente al televisor y estar un paso más cerca de ser parte del ambiente virtual.

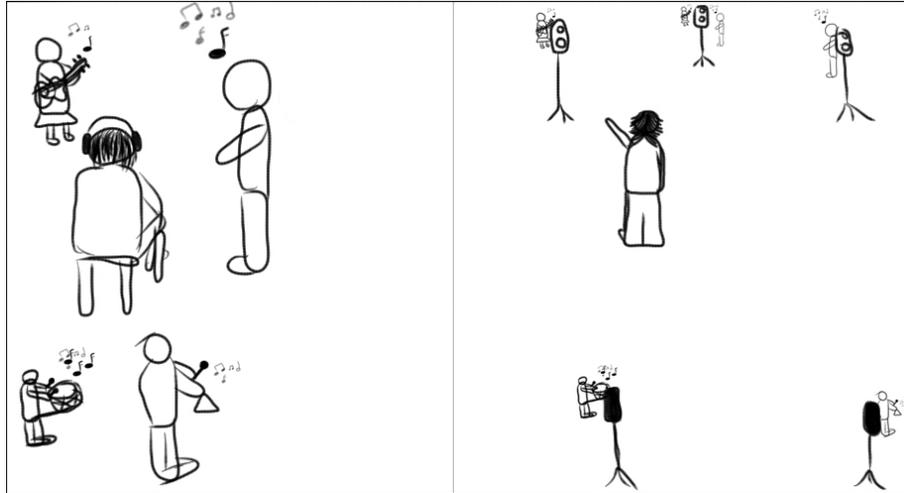


Figura 2.3: Ilustración del uso de sonido binaural y envolvente.

2.2.1 Binaural

Una de las técnicas comunes para la reproducción de audio espacial en audífonos es el sonido binaural. Está basado en la emisión de sonidos independientes a cada oído con base en las variaciones que pueden existir de la fuente al receptor[20, p. 89-93]. El cerebro interpreta las diferencias interaurales y de esta manera logra identificar la posición espacial aproximada del emisor.

Por lo general se requieren varios micrófonos posicionados en distintas partes de la cabeza del receptor para poder grabar sonido, el cuál sea reproducido de manera binaural. Después se aplica la convolución del HRTF¹[20, p. 89], tomando en cuenta la posición virtual deseada al sonido mono capturado y se reproduce en una bocina. Existen varios sitios Web que cuentan con amplias bases de datos con las mediciones de HRTF de un grupo de personas, lamentablemente en su mayoría no tienen los datos para todos los ángulos, por lo que se tiene que hacer una interpolación la cual puede reducir el sentido de inmersión.

El sonido binaural puede ayudar al usuario a identificar con mayor precisión la localización de una fuente de sonido. Además, no requiere de dispositivos avanzados si ya se tiene acceso a una base de datos de HRTF. Por otro lado, este enfoque requiere el uso

¹Head-Related Transfer Function

constante de audifonos, por lo que dificulta el permitir al receptor desplazarse libremente en un espacio y escuchar los cambios del sonido con base a su posición.

2.2.2 Envolverte 5.1

El sonido envolvente difumina en distintas locaciones los diversos sonidos generados para que el receptor crea que el audio viene de su alrededor y no solo de enfrente. Con el transcurso de la evolución de esta tecnología han surgido distintas certificaciones como THX[21] y Dolby Surround Sound[22], con el propósito de que los estudios de cine y videojuegos puedan garantizar la experiencia auditiva.

Existen diversas configuraciones usualmente utilizadas en un sistema de altavoces, mencionando primero la cantidad de bocinas y después la del subwoofer. El más convencional es el 2.0 utilizado en estéreos, computadoras, televisiones y otros. Con el avance de la tecnología y del entretenimiento surgió la necesidad de crear un sistema que hiciera sentir al espectador dentro del mundo virtual. Debido a lo anterior, se crearon sistemas 5.1 y 7.1 capaces de engañar al usuario y hacerlo creer que el sonido esta sucediendo en tiempo real y generado dentro del entorno espacial en que se encuentra.

Una configuración 5.1 permite que el usuario se desplace dentro del rango de las bocinas, percibiendo cómo cambia el sonido sin necesidad de informar al sistema su posición actual. Así, se pueden tener varias personas al mismo tiempo y cada una tendrá una experiencia independiente.

2.3 Kinect

El *Kinect* es un dispositivo creado por Microsoft, el cual une múltiples sensores para lograr reemplazar los controles convencionales usados en videojuegos. Consta de dos sensores infrarrojos para poder determinar la profundidad, una cámara RGB, una serie de micrófonos y un motor para ajustar su inclinación. A causa de lo anterior, Microsoft ha tenido un gran éxito vendiendo el electrónico porque ha permitido a la comunidad académica y de

entretenimiento el explorar nuevos horizontes en interacción con el usuario.



Figura 2.4: Elementos del Kinect.

La cámara de video y los sensores infrarrojos obtienen datos a 30 cuadros por segundo a una resolución de 640x480 píxeles, lo que permite obtener una gran cantidad de datos útiles para poder tomar decisiones. El motor permite una inclinación de hasta 27 grados con un campo de vista horizontal y vertical de 57 y 43 grados respectivamente. Además, debido al distinto posicionamiento de cada sensor, la imagen requiere de ajustes para poder tener una relación uno a uno entre la información RGB y la de profundidad.

Es notable que el *Kinect* es un dispositivo que ha logrado promover una nueva serie de aplicaciones que antes no existían debido a la complejidad de obtener y sincronizar los diversos sensores, pero ahora en un solo aparato se pueden realizar capturas de vídeos y utilizar la profundidad para entender mejor qué es lo que está frente al dispositivo.

2.4 Generación de la imagen

Al trabajar con grandes cantidades de datos, se torna difícil el poder identificar todos los cambios que están sucediendo en ellos cuando únicamente se observan una serie de cadenas de texto. Así, las imágenes son una excelente solución para mostrar y entender mucha información a través de gráficas o indicadores visuales. Por ejemplo, si se quiere

analizar la información de las distancias capturadas por la cámara de profundidad, como cada cuadro que se recibe es de tamaño de 320x240, se deberán estar analizando 76,800 datos de valor flotante por cada imagen. Una solución visual a este problema es generando una imagen del mismo tamaño que el arreglo bidimensional de las distancias proporcionado por la cámara de profundidad. Para ello se deberán normalizar dichas distancias para que queden en un rango de 0 a 255 y poderlas mostrar entonces como una imagen en escala de grises o en RGB. El resultado es una representación visual que permite de una manera mucho más sencilla y a primera vista, entender mejor los miles de datos de las distancias, como muestran claramente las imágenes de la Figura 2.5.

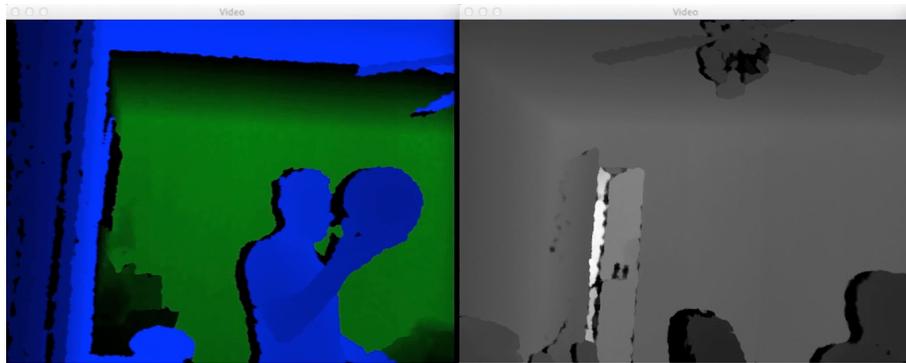


Figura 2.5: Mapeo de la profundidad capturada a una imagen en RGB y en escala de grises.

Muchas aplicaciones y la mayoría de los videojuegos crean mundos virtuales 3D que permiten al usuario tener una mayor sensación de pertenencia. Existen diversas técnicas para transformar la descripción de una serie de objetos 3D a una imagen plana y asignarles iluminación, entre las más comunes se pueden mencionar las basadas en los modelos de iluminación de *Phong*[23, p. 68-86], *Ray Tracing*[23, p. 233-251] y *Photonmapping*[23, p. 252].

OpenGL[24, p. 33-34] es una librería para gráficos 3D y modelado basado principalmente en una variante de *Ray Tracing*. A causa de su gran velocidad y sencillez, es usado para la creación de videojuegos, manejo de ventanas para Sistemas Operativos e incluso dispositivos móviles. Este paquete es para ser utilizado al programar en C o C++ pero existen variantes como *JOGL*[25] para poder desarrollar en Java. La gran flexibilidad y portabilidad de esta librería permiten generar soluciones multiplataforma sin necesidad de recompilar el código.

2.5 Optimización

La optimización del rendimiento de los programas se ha vuelto una realidad debido a los grandes avances en la computación respecto a poder de procesamiento, paralelismo y muchos otros. Existen diversas técnicas que agilizan la ejecución del código logrando una interacción continua, ya que no es usable una aplicación la cual mantiene esperando por respuesta al usuario. Por lo tanto se tiene que trabajar en mantener una comunicación fluida que genere un sentimiento de inmersión y al mismo tiempo evite requerir de sistemas de cómputo con mayor capacidad de procesamiento.

2.5.1 Orientado a objetos

En esta tesis se desarrolló una aplicación aprovechando las bondades del modelo Orientado a Objetos[26, p. 1-179]. Este enfoque se basa en tratar de modelar la vida real en objetos, a diferencia de la programación procedural en la cual se tienen una serie de métodos que se van ejecutando para lograr un objetivo final. Una de las grandes ventajas de OO² es la facilidad de tener relaciones entre objetos como es la herencia.

Java[27] es un lenguaje de programación basado en OO, que en los últimos años ha tenido un gran auge debido a su flexibilidad y ser interpretado en lugar de compilado. Debido a lo anterior el código fuente pasa a ser bytecode, el cual es interpretado por la JVM³ para ser ejecutado. Esta arquitectura permite a Java ser multiplataforma, pero se debe tener en mente que se pueden usar librerías nativas las cuales pueden obligar al programa a sólo correr en un Sistema Operativo.

2.5.2 Secuencias de datos

Los *data streams* son secuencias de datos[28, p. 752] que suelen ser una forma muy veloz de obtener información, sin embargo, son poco amigables. La información se lee directamente desde la fuente en binario y por eso es rápido, pero confuso. En lugar de poder recorrer un

²Orientado a Objetos

³Maquina Virtual de Java

arreglo de enteros y realizar alguna operación con ellos, se debe saber cuántos bytes ocupa nuestro tipo e ir transformando los datos. Se pueden usar *data streams* principalmente en el manejo de recepción y salida de datos a dispositivos, por lo tanto si se tiene un aparato como el *Kinect*, el cual manda una gran cantidad de datos a los que se tiene que reaccionar en tiempo real, esta optimización puede ayudar aumentando el rendimiento.

2.5.3 Hilos

Cuando se desean tener varias actividades ejecutándose al mismo tiempo, es común el uso de Hilos en Java. La programación concurrente[28, p. 1046-1110] tiene una gran cantidad de retos, ya que debe existir un punto de salida y uno de encuentro. En el pasado la concurrencia era un espejismo, ya que sucedía en un solo procesador, el cuál en realidad dejaba de hacer una actividad para continuar haciendo otra y así intercambiaba de manera veloz para simular paralelismo. Hoy en día es una realidad debido a que las computadoras actuales tienen un procesador o más, con varios núcleos encargados de realizar actividades independientes.

Se pueden utilizar hilos si se logran identificar áreas independientes que puedan correr al mismo tiempo y utilizar recursos en común de manera serializada para evitar caer en un *deadlock*[28, p. 1088]. El recurso en común puede no ser serializado si el programa únicamente lee datos, pero si se permite además la lectura y escritura, entonces se deberá tener una lista de espera de los hilos que desean acceder la información y atenderlos así uno a uno. Lo anterior evita que existan problemas de sincronización, logrando que los datos sean consistentes.

El rediseñar áreas de un programa para emplear hilos es una excelente forma de mejorar el rendimiento de una aplicación. La optimización obtenida con base en lo anterior se refleja en un código capaz de correr secciones independientes y reducir los cuellos de botella dependiendo de recursos comunes.

CAPÍTULO 3

Desarrollo

3.1 Diseño general

Teniendo en mente los problemas planteados en el Capítulo 1 se diseñó una arquitectura que permite funcionar en tiempo real. En los primeros intentos se buscó el tener una arquitectura lineal donde un solo hilo estuviera encargado de recabar la información del *Kinect*, procesar la imagen, actualizar el mundo virtual y reproducir el sonido acorde a lo que estaba sucediendo en la aplicación. Después de realizar varias pruebas se encontró que la librería del *Kinect* abortaba la ejecución debido a un problema con la información de profundidad. Al analizar la estructura de la librería se encontró que en realidad lo que estaba sucediendo es que la memoria del *Kinect* se estaba desbordando ya que la aplicación recolectaba los datos de profundidad a menos de 30 cuadros por segundo, por lo tanto la aplicación fallaba y no se podía capturar la excepción a través de los mecanismos existentes en Java. Debido a lo anterior, se rediseñó la arquitectura para aprovechar la capacidad actual del hardware en procesamiento paralelo y el cual cuenta con 5 módulos independientes, los cuales corren al mismo tiempo, logrando mantener 30 cuadros por segundo y una interacción completamente inmersiva.

- ***Kinect***: Está encargado de manejar la conectividad con el dispositivo, realizar los cambios de ángulo, dar retroalimentación utilizando el LED y mantener mínimo a 30 cuadros por segundo la recolección de la información de profundidad. De esta manera se agiliza el proceso de recabado de datos y se puede evitar el error antes

mencionado; pero cabe mencionar que si este hilo se pone a dormir, es altamente probable que se reproduzca el error.

- **Sonido:** Está encargado de la reproducción del audio en 3D a través de varias librerías como se describe en la sección 3.3.2. Tener un hilo independiente para el sonido ayuda a evitar confusiones en el desarrollo de la aplicación además de centralizar los accesos a librerías externas logrando evitar duplicidad en las operaciones y agilizar el proceso de investigación y corrección de posibles errores.
- **Mundo Virtual:** Está encargado de generar la representación tridimensional de la aplicación y mostrar en pantalla lo que está sucediendo para poder visualizar errores como administrador. Este hilo trabaja directamente con la librería de *OpenGL* por lo que muestra una gran variación en rendimiento de la aplicación al hacerlo independiente y realizar los accesos a información de manera asíncrona ya que se manejan a diversas velocidades el procesamiento de imágenes, el *Kinect* y el Mundo Virtual.
- **Juego:** Está encargado del sistema de menús y sus niveles mediante el uso de los hilos antes descritos. Este bloque está diseñado para crear de una manera más fácil nuevos niveles del juego, definiendo la cantidad, posición y rutas de las abejas. Además tiene como ventaja el poder definir nuevas reglas para la aplicación sin necesidad de realizar cambios a los demás hilos.
- **Control Principal:** Está encargado de crear e inicializar los demás hilos y mantener la conexión con el sistema de ventanas del sistema operativo donde se está ejecutando la aplicación.

La arquitectura está basada en el modelo MVC¹[29, p. 89-90] para poder separar de la mejor manera los bloques que se corren como hilos independientes y también facilitar la comunicación entre dispositivos como se muestra en la Figura 3.1. Además, la aplicación se diseñó para ser creada como una plataforma, para de ahí crear la solución específica de

¹Modelo Vista Controlador

esta tesis y facilitar futuros desarrollos semejantes. El resultado es una arquitectura multiplataforma, parcialmente flexible y de alto rendimiento, la cual une diversas tecnologías de software y hardware en una aplicación multisensorial.

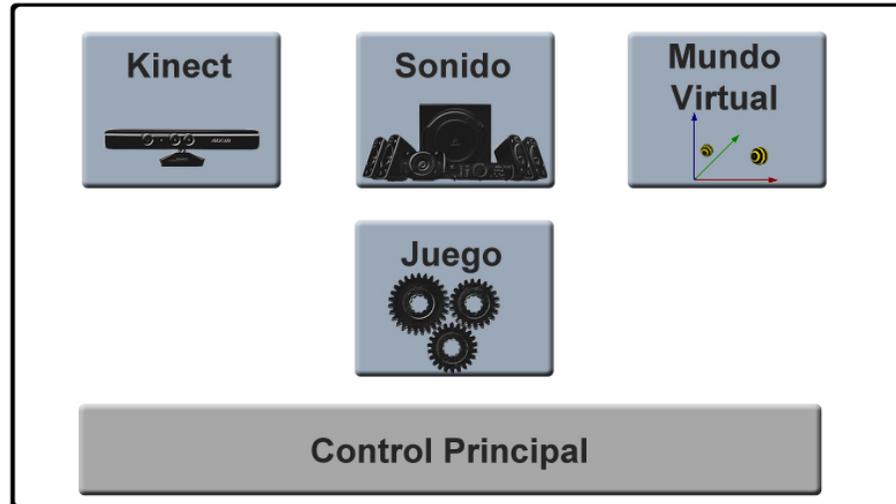


Figura 3.1: Ilustración de la arquitectura de la aplicación.

3.2 Módulo *Kinect*

El principal motivo por el cual se decidió utilizar el *Kinect* en lugar de una cámara RGB o un láser para medir la profundidad es que es un dispositivo fácil de encontrar y además a un costo bajo comparado con soluciones más complejas. Es importante destacar que el SDK de Microsoft incluye un mayor rango de funciones que las librerías *open source*, entre ellas destacan la auto generación del esqueleto del usuario y el uso de los micrófonos. Debido a lo anterior, se desarrollaron algoritmos para poder aislar al usuario y poder manipular el menú.

3.2.1 Inicialización y retroalimentación

A través del *OpenKinect* se tiene acceso a la cámara de RGB, al sensor infrarrojo, al LED y al motor de inclinación del dispositivo. El primer paso consta de la aplicación ajustando la inclinación alrededor de los 0 grados y el usuario debe asegurarse que el dispositivo esté aproximadamente a 1.20 metros de altura sobre el suelo. El siguiente paso es crear un

objeto, el cual recibirá la imagen capturada por el sensor de profundidad y se mantendrá en un ciclo constante hasta que se le indique que tiene que terminar. Finalmente, se tiene que liberar la memoria y cerrar la conexión con el dispositivo para poder volver a usarlo.

El *Kinect* puede obtener imágenes de una resolución de 640x480 tanto en la cámara de RGB como del sensor infrarrojo, debido a lo anterior, se tiene suficiente información para poder tener mayor precisión en la posición del usuario y los movimientos que son realizados. En la inicialización de la conexión se debe de especificar qué resolución se desea obtener, teniendo como opción 320x240 o 640x480. Asimismo, versiones previas de *JOGL* no permitían obtener imágenes simultáneamente de la cámara RGB y del sensor infrarrojo por lo que se tenía que especificar cual de los 2 se deseaba usar.

3.2.2 Traducir los datos del sensor de profundidad

El sensor de profundidad captura las distancias en una imagen de capacidad de 11 bits de tamaño 320x240 ó 640x480 dependiendo de la inicialización. La mayoría de librerías utilizan imágenes de 8 o 16 bits por lo que al intentar obtener las distancias en centímetros como tipo *short* los valores pueden no hacer sentido. Lo anterior sucede ya que el Sistema Operativo esta obteniendo los datos en *Big Endian* y se necesita transformar cada elemento a *Little Endian*.

Después, se debe realizar un ajuste a cada elemento ya que existe una disparidad entre las cámaras infrarrojas. Cuando se tiene un sistema estéreo las cámaras se encuentran calibradas, pero en este caso la relación entre disparidad y profundidad es dada por $z = \frac{bf}{d}$ donde z es la profundidad en metros, b es la distancia horizontal en metros entre las cámaras, f es la distancia focal entre las cámaras en pixeles y d es la disparidad en pixeles[30]. Por lo tanto una disparidad de 0 deberá de resultar en una profundidad infinita, el problema es que la disparidad que se obtiene del *Kinect* no sigue esa norma, por lo que vemos que la disparidad tiene que estar dada por $d = \frac{1}{8}(doff - kd)$ donde el factor de 1/8 se debe a que los valores de kd están en unidades de 1/8 de pixel, d es la disparidad normalizada del *Kinect*, kd es la disparidad del *Kinect* sin normalizar y $doff$ es un offset

que depende de cada *Kinect*. El Dr. Xiuwen Liu de la Universidad del Estado de Florida usa un valor de 10.915 metros[31][32] mientras que Kurt Konolige y Patrick Mihelich[30] aproximan que el promedio es de 10.90 metros, por lo que en nuestra aplicación el offset tiene un valor de 10.915 metros después de realizar varias pruebas y comprobar que este valor nos da la precisión necesaria. Con base a lo anterior, la profundidad está dada por $z = \frac{bf}{\frac{1}{8}(doff - kd)} = \frac{8bf}{doff - kd}$, en la Tabla 3.1 se encuentra la definición y unidad de cada variable, donde la distancia horizontal promedio entre las cámaras infrarrojas es 0.075 metros, la distancia focal de la cámara infrarroja es de 580 unidades de pixel, el offset es de 10.915 metros y la disparidad del *Kinect* es el valor en centímetros que tenemos en la matriz de 11 bits que nos da el *Kinect*, por lo que la profundidad está dada por $z = \frac{(8)(7.5)(580)}{(1091.5) - kd} = \frac{34800}{(1091.5) - kd} cm$.

Variable	Definición	Unidad
z	Profundidad	Metros
b	Distancia horizontal entre las cámaras	Metros
f	Distancia focal entre las cámaras	Pixeles
d	Disparidad	Pixeles
$doff$	Offset	Metros
kd	Disparidad del Kinect	1/8 de pixel

Tabla 3.1: Definición y unidad de las variables utilizadas para la traducción de datos del sensor de profundidad.

A causa de lo anterior se deben realizar una gran cantidad de cálculos solo para obtener la profundidad, ya que el *Kinect* captura 30 cuadros por segundo lo que significa que se tiene que resolver la ecuación de profundidad un total de 9'216,000 ocasiones cada segundo. Al intentar usar las estructuras más comunes como arreglos y listas para mantener esta información, se encontró que es muy lento y el *Kinect* debe consistentemente obtener los datos generados, de lo contrario se desborda su alojamiento en el dispositivo causando que la aplicación termine sin poder manejar la excepción. Para evitar lo anterior se utiliza una

secuencia de datos de tipo *ByteBuffer* la cual funciona más rápido que otras estructuras, logrando mantener la recepción de datos de manera consistente.

Uno de los problemas más comunes al utilizar *ByteBuffer* es que se debe de tener cuidado en qué momento se lee la información que contiene, ya que este tipo cambia de manera dinámica por lo que fácilmente se pueden obtener datos desactualizados, incongruentes o una excepción. Para el caso de nuestra aplicación desarrollada, se accesa a la variable realizando una previa verificación de que los datos sean congruentes a través de una serie de banderas que se activan y desactivan antes y después de realizar cambios a la variable, de esta manera los datos obtenidos son consistentes y confiables.

3.2.3 Aislar el fondo

Como se explica en la sección 2.1.1, el uso de una cámara RGB tiene diversos inconvenientes, los cuales difícilmente pueden ser evitados por personas con discapacidad visual ya que no se darán cuenta cuando haya cambios de iluminación, haciendo frustrante la interacción de este tipo de usuarios con la aplicación. Para evitar este problema, se trabajó en realizar el aislamiento del fondo de las imágenes utilizando la cámara de profundidad, bajo la premisa de que dicho fondo es estático y siempre está atrás del jugador, por lo que si la profundidad es menor significa que existe algo nuevo en ese pixel.

Para identificar el fondo de las imágenes primeramente se dejan pasar 100 cuadros para permitir al software interno del *Kinect* que haga los ajustes que requiera. Después, durante los siguientes 100 cuadros se obtiene su profundidad y se suma a los que ya fueron capturados en una sola variable de tipo *ByteBuffer*, esto se hace para realizarlo velozmente y sin necesitar espacio extra en la memoria. Al terminar la captura de los 100 cuadros se obtiene el promedio y se guardan las distancias en una imagen de 16 bits. Ahora todo cuadro que sea obtenido por el *Kinect* será comparado con la imagen recién generada para obtener una nueva imagen la cual tenga únicamente la profundidad de los pixeles que no pertenecen al fondo.

Para eliminar el fondo en cada cuadro es recomendable tener un rango el cual especifique qué distancias sí deben tomarse en cuenta y reducir la cantidad de operaciones que se sabe que no son de utilidad. Por lo tanto se recibe cada cuadro nuevo y se crea una imagen binaria temporal inicializada en todos sus pixeles en 0. Pixel por pixel se siguen los siguientes pasos para determinar si pertenecen o no al fondo de las imágenes. A los pixeles identificados que no pertenecen al fondo se les asigna un valor de 255:

1. Obtener la profundidad real del cuadro capturado con la ecuación de profundidad.
2. Determinar si el valor obtenido en el Paso 1 está dentro del rango especificado por el método. En caso contrario terminar con este proceso, ignorando los pasos 3 y 4.
3. Restar el valor de la profundidad promedio del valor del pixel analizado y si dicha diferencia en valor absoluto no es menor de 7 centímetros, valor obtenido empíricamente, entonces terminar el proceso, ignorando el Paso 4.
4. Asignar al pixel analizado el valor de 255, en una imagen binaria temporal.

Al terminar el recorrido por todos los pixeles se tiene una imagen binaria del mismo tamaño que el cuadro capturado, en dicha imagen se le realizan las siguientes operaciones:

1. 3 erosiones con el elemento estructural de la Tabla 3.2.
2. 4 dilataciones con el elemento estructural de la Tabla 3.2.
3. 1 dilatación con el elemento estructural de la Tabla 3.3.
4. 2 erosiones con el elemento estructural de la Tabla 3.2.

1	1	1
1	0	1
1	1	1

Tabla 3.2: Elemento estructural binario de tamaño 3x3 en forma de rectángulo.

El número y orden de las operaciones antes descritas, así como los elementos estructurales usados, se seleccionaron empíricamente después de realizar pruebas con varias personas de distintas características físicas y en diversas posiciones, con el fin de eliminar la sombra que existe al ver los datos de profundidad del sensor como se muestra en la Figura 3.2. Esta se debe a que la cámara infrarroja y el proyector infrarrojo no captaron la misma información, esto suele suceder cuando existe un obstáculo que únicamente es visible por uno de estos dos dispositivos, causando que los datos de esa región se consideren inválidos y asignándoles un valor de 0, es decir, de color negro.

En las pruebas se encontró que 3 erosiones iniciales con el elemento estructural de la Tabla 3.2 logran eliminar el ruido, ya que el elemento estructural seleccionado cambia al valor 0 los pixeles que no están inmediatamente rodeados por pixeles con valor 1, sin embargo, se pierde información relevante cerca del centro del jugador y en zonas de poca conexión como suele ser entre los dedos de la mano. Para enmendar lo anterior se realizan a continuación 4 dilataciones con el mismo elemento estructural, las cuales reconectan los dedos de la mano, seguida de 1 dilatación con el elemento estructural de la Tabla 3.3 para volver a conectar espacios que pudieron quedar desconexos al centro del usuario. Finalmente solo queda refinar la silueta del jugador volviendo a realizar 2 erosiones para eliminar excesos de pixeles.

0	0	0	1	0	0	0
0	0	0	1	0	0	0
0	0	0	1	0	0	0
1	1	1	1	1	1	1
0	0	0	1	0	0	0
0	0	0	1	0	0	0
0	0	0	1	0	0	0

Tabla 3.3: Elemento estructural binario de tamaño 7x7 en forma de cruz.

El siguiente paso consta de recorrer pixel por pixel la imagen binaria y si el valor del pixel es 0 entonces al pixel equivalente del cuadro capturado se le pone un valor de -1 y si es 1 entonces se guarda en una variable temporal si es la profundidad más pequeña encontrada y además se guarda en otra variable cuál es la posición del pixel. Al finalizar este proceso se tiene una imagen con las profundidades que no pertenecen al fondo, la profundidad más pequeña encontrada y el punto en el que se encontró, estos últimos dos valores son usados para el manejo de las opciones en los menús y detectar cuando el usuario desea pausar el juego.

La imagen obtenida es independiente de la iluminación ya que proceden de un dispositivo infrarrojo, por lo que no importa si es de día o de noche, el usuario puede estar en un cuarto completamente oscuro y el sensor infrarrojo leerá de manera correcta la profundidad. Por lo tanto esta manera de aislar el fondo permite al jugador estar en interiores o en exteriores, siempre y cuando los rayos del sol no estén pegando directamente al *Kinect* y afecten la información proporcionada por las cámaras infrarrojas. Si esto último puede evitarse, el proceso que acabamos de describir podrá identificar con gran precisión las siluetas de los usuarios como se ilustra en la Figura 3.2.

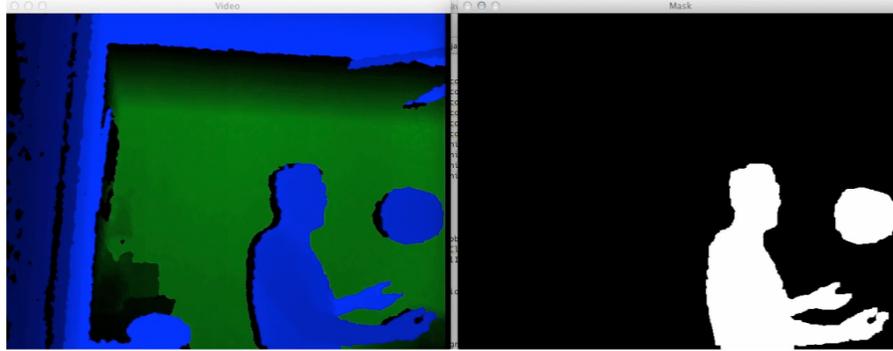


Figura 3.2: Imagen antes y después de ser procesada con las operaciones morfológicas para obtener la máscara.

3.2.4 Restricciones

El *Kinect* tiene una serie de limitaciones que son importantes conocer para evitar dañar el equipo y saber hasta donde llegan sus límites.

- No se debe mover el ángulo de la cámara más de una vez por segundo.
- No se debe mover el ángulo de la cámara más de 15 veces en un periodo de 20 segundos.
- El rango de inclinación posible es de -27 a 27 grados.
- Al tener una de las cámaras capturando imágenes, se deben sacar los datos del *Kinect* a 30 cuadros por segundo mínimo, de lo contrario se desborda la memoria interna del dispositivo causando un error que no se puede atrapar.
- El rango de distancia al cual puede detectar objetos el *Kinect* es de 0.7 metros hasta 6 metros.

3.2.5 Interacción con el usuario

A primera vista no parece evidente que al crear una aplicación sin una pantalla se requiere de una nueva manera de interacción para el jugador, ya que los humanos estamos acostumbrados a usar la computadora y las consolas a través de algún dispositivo físico, como lo son un teclado o un control, por ejemplo. Desde la aparición del *Kinect* han surgido varios

juegos que permiten la interacción sin necesidad de estos dispositivos físicos, sin embargo aún se requiere de una pantalla por parte del usuario poder entender qué está haciendo.

Recordemos que la solución que se está proponiendo en esta tesis consiste en recurrir a conceptos básicos que no estén fundamentados en el uso de la vista por parte del usuario. En este sentido, considerando que el usuario entiende el concepto de adelante, atrás, arriba, abajo, izquierda y derecha, se puede diseñar el sistema de menús con base a este conocimiento. Además, se hará uso de retroalimentación sonora para permitir al jugador saber en que etapa o nivel de juego está y qué se espera que haga a continuación, como se explica en la sección 3.3.3.

Primero se debe decidir cómo se podrá invocar el sistema de menús. Aprovechando las capacidades del *Kinect*, la zona de juego se divide en 5 secciones las cuales se muestran en la Figura 3.3. En otras palabras dependiendo del punto más cercano del usuario hacia el *Kinect*, la aplicación sabrá si se desea entrar a pausa, si se desea controlar el menú o continuar jugando.

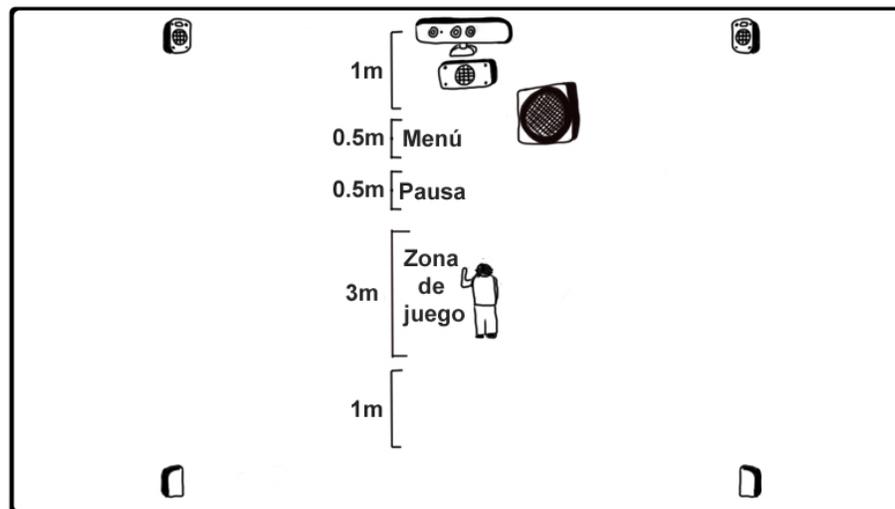


Figura 3.3: Ilustración de las regiones de interacción de la aplicación.

A continuación se debe solucionar la selección de opciones en el menú, por lo que la pantalla se puede dividir en 4 o en 2 opciones. Como en la mayoría de los menús, dividiremos la pantalla en 4 como se muestra en la Figura 3.4, por lo que cada menú tendrá

como máximo 4 opciones a elegir, sin embargo, se puede hacer que una de ellas sirva para cambiar de página de opciones logrando un número ilimitado de opciones a través de un sencillo sistema de paginado. Cada que se cambia de opción se notifica la acción a través de un sonido y para confirmar que dicha selección es la opción deseada se debe mantener la selección por más de 3 segundos aproximadamente.

Esta solución que se está proponiendo permite que el usuario pueda interactuar con las opciones del menú sin requerir del uso de varias o todas sus extremidades, en otras palabras, el jugador puede estar por ejemplo en una silla de ruedas utilizando un bastón para seleccionar las opciones del menú y estar jugando, ya que se requiere solamente del punto más cercano al *Kinect* para el manejo de opciones, como se describió en la sección 3.2.3. Asimismo, significa que un usuario puede usar su cabeza o sus pies para la interacción con la aplicación, permitiendo a un mayor número de personas con diferentes tipos de discapacidades el disfrutar de la experiencia de esta aplicación.

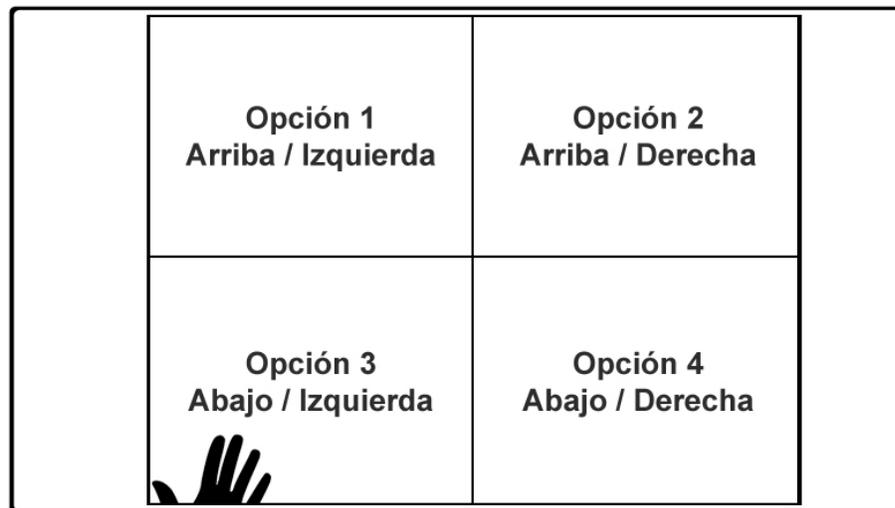


Figura 3.4: Ilustración del sistema de menús empleado en la aplicación.

3.3 Módulo Sonido

En esta aplicación los oídos son los ojos que permiten al usuario poder interactuar con el juego, debido a lo anterior es muy importante el lograr la inmersión del jugador con sonidos

únicos y reconocibles. Inicialmente se exploró la posibilidad de usar sonido binaural para dar una mayor sensación de realismo pero se encontraron diversos problemas, entre ellos destacan:

- El usuario está en constante movimiento por lo que se necesitaría ajustar los sonidos con base en la posición, altura y rotación del jugador.
- El sonido binaural requiere del uso de audífonos por lo que sería necesario que fueran inalámbricos para evitar la posibilidad de que se tropiece el jugador.
- Las bases de datos gratuitas no incluyen todos los datos requeridos para poder generar el sonido con diversas alturas y en 360 grados.

Por lo tanto se decidió utilizar sonido envolvente el cuál depende únicamente del posicionamiento de las bocinas, por lo que el usuario puede moverse con libertad dentro del área seleccionada.

3.3.1 Sistema de sonido y posicionamiento

Una gran cantidad de personas están acostumbradas a escuchar acerca de teatros en casa y sonido envolvente, por lo general estos sistemas se conectan por cable óptico o HDMI a las consolas y televisiones, pero es muy diferente cuando se requiere que una computadora logre mandar de manera correcta el sonido a uno de estos sistemas. Al explorar la computadora es evidente que en su mayoría solo tienen 1 salida de sonido de 2 canales, por lo que se requiere un dispositivo el cual permita conectar un sistema envolvente.

Por lo tanto, para esta tesis se adquirió una tarjeta de sonido externa marca *Sabrent* con el modelo *USB-SND8*, ya que este dispositivo es fácil de obtener, se conecta a la computadora por medio de USB y permite conectar sistemas de sonido de entrada y salida a través de cable óptico y conexión análoga.

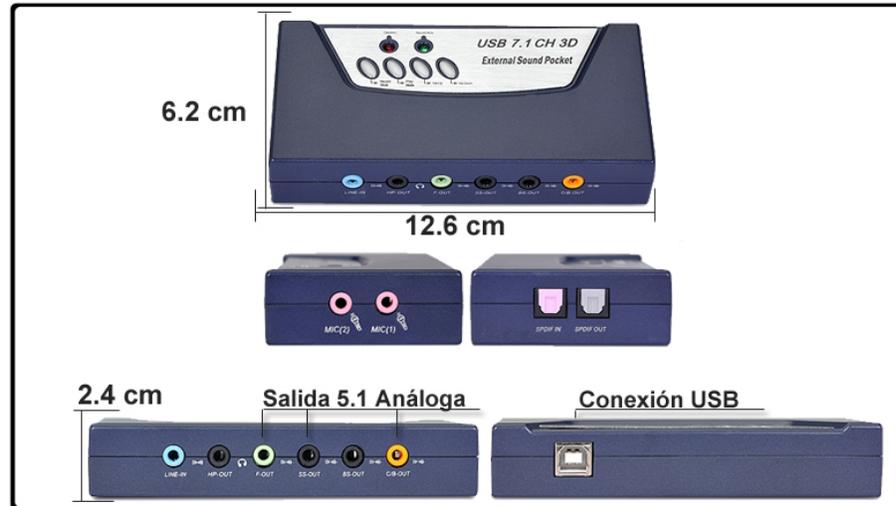


Figura 3.5: Imagen de la tarjeta de sonido externa Sabrent USB-SND8 usada para poder conectar la computadora a un sistema 5.1.

El siguiente paso fue encontrar un sistema de sonido el cuál tuviera una excelente calidad, entrada por conexión analógica 5.1 ó cable óptico y con control remoto para mantener la libertad del usuario al interactuar con el juego. Para esta tesis se adquirió un sistema de sonido *Logitech z906 5.1* con certificación *THX* ya que cumple con los requisitos antes mencionados y la certificación asegura la calidad del sonido. En la figura 3.7 se muestra la gran variedad de entradas que existen para este sistema de sonido y cómo se debe conectar la tarjeta de sonido externa al sistema de sonido para emitir correctamente el audio a cada bocina. Lo antes mencionado es de suma importancia ya que la tarjeta externa soporta sonido 7.1 y 5.1, por lo que al conectar mal los cables, el audio puede no escucharse o sonar en distinta posición de la esperada.



Figura 3.6: Imagen del sistema de sonido Logitech z906 usado en la aplicación.



Figura 3.7: Imagen de las entradas del sistema de sonido Logitech z906 y como conectar la tarjeta de sonido externa Sabrent USB-SND8 al sistema.

Además, la sensación de inmersión está basada en la precisión del sonido, es por eso que el posicionamiento de las bocinas y el usuario juegan un papel de gran importancia. Con base en las sugerencias de posicionamiento de *Dolby Surround*[22] y la zona especificada por la aplicación para que el usuario pueda interactuar, se recomienda colocar los parlantes como se muestra en la Figura 3.8. Teniendo en mente lo antes mencionado, se requiere aproximadamente de un área de $12.6m^2$ sin obstáculos para poder interactuar con la aplicación y poner los dispositivos requeridos.

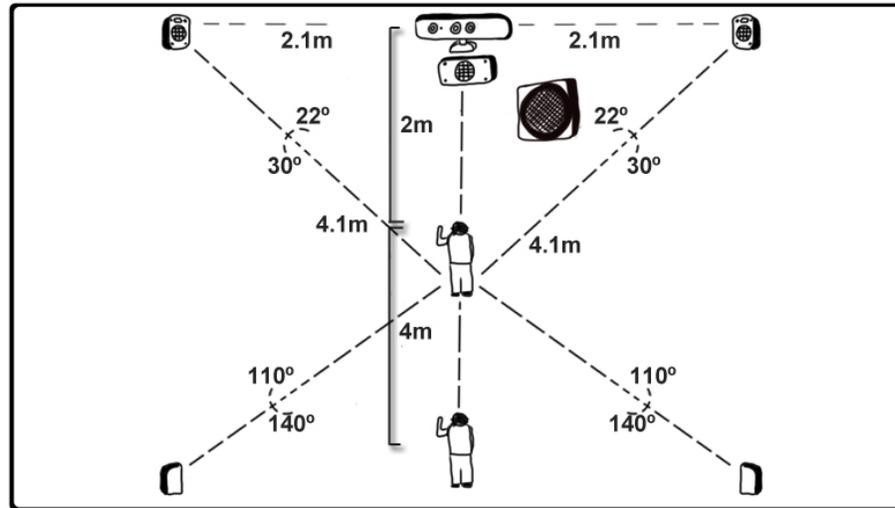


Figura 3.8: Ilustración del posicionamiento correcto del sistema de sonido y del *Kinect*.

3.3.2 Sonido en 3D

Además de tener los dispositivos adecuados para poder generar sonido envolvente, se requieren librerías las cuales permitan reproducir el sonido deseado en una posición 3D. En esta tesis se usa la librería de Paul Lamb[33] la cuál emplea como base la *LWJGL*²[34], juntas proveen de valiosas herramientas para poder reproducir archivos de extensión WAV³ en una determinada coordenada x,y,z. Algunas de las funciones incluidas son para modificar en tiempo real el volumen, tono, posición y otras propiedades. Estas librerías pueden utilizar su propio hilo por lo que no es necesario detener la ejecución de la aplicación y también pueden tocar múltiples audios de manera simultánea.

Ambos paquetes son multiplataforma y la configuración para reproducir el audio en las bocinas correctas depende del sistema operativo, en el caso de esta tesis se usó *Mac OS X* versión 10.7 por lo tanto existe un menú llamado *Audio MIDI Setup*, en la sección de preferencias del sistema, el cual permite hacer los ajustes requeridos como se muestra en la Figura 3.9. Si se ajustan las propiedades para utilizar sonido 7.1 pero en un sistema 5.1 existirán problemas al tratar de mandar el sonido ya que el sistema operativo encontrará que existe un desborde de información, por lo que después de un par de minutos dejará

²Lightweight Java Game Library

³Waveform audio format

de escucharse el audio.

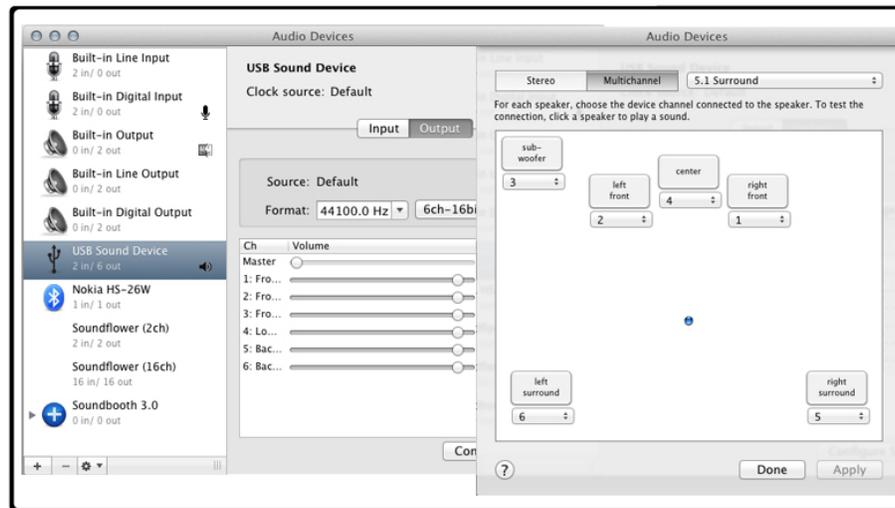


Figura 3.9: Imagen de la configuración correcta del sonido en Mac OS X.

Al realizar las primeras pruebas se hizo evidente que modificar la altura de los objetos virtuales no causaba ningún cambio en el audio, por lo tanto el usuario no podía distinguir la altura del sonido. Además, al correr simulaciones de objetos pasando del frente hacia atrás no se escuchaba de manera real, por lo que el jugador podía confundirse al querer ubicar e interactuar con los objetos virtuales. Para resolver el primer punto se ajustó el valor del tono tomando en cuenta la altura del objeto, de forma tal que el valor del tono estará dado de acuerdo a la operación $\frac{1}{|altura|}$. El tono está normalizado por lo que sus posibles valores son de 0 a 1, de esta forma se da un punto de referencia al usuario acerca de la altura en la que se encuentra el objeto como se muestra en la Figura 3.10. En el caso del ajuste de la profundidad del sonido se modifica el volumen normalizado con la fórmula $\frac{1}{|(3)(profundidad)|}$, el valor de 3 fue obtenido empíricamente, con este ajuste se logra una sensación más realista de la profundidad del sonido de los objetos virtuales.

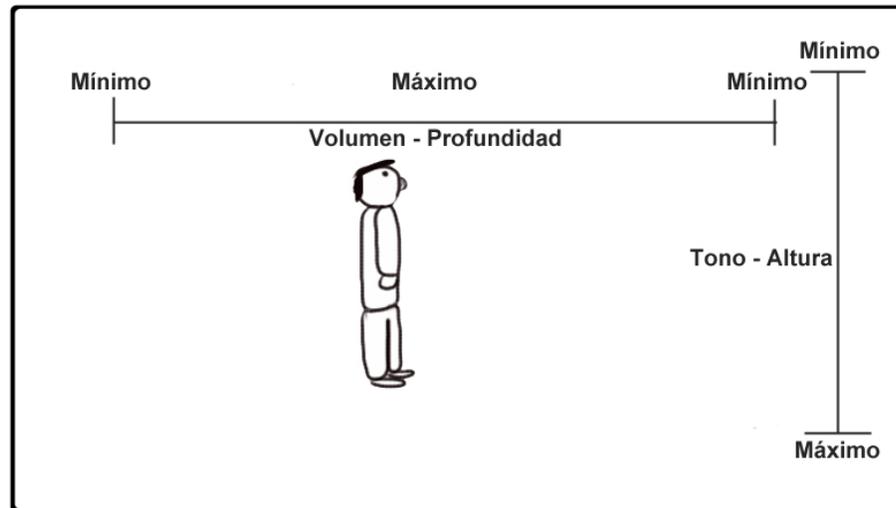


Figura 3.10: Ilustración del ajuste al volumen y tono para mejorar la percepción de la profundidad y altura.

3.3.3 Retroalimentación

Debido a que el usuario no puede recibir retroalimentación visual por su condición, se decidió reemplazar con sonidos dicha retroalimentación, en lugar de utilizar alguna solución vibrotáctil para promover la navegación a través del oído. Con base a lo anterior se diseñó la aplicación para dar retroalimentación al jugador a través de audio en español en las distintas secciones y eventos como se muestra en la Tabla 3.4. La retroalimentación incluye sonidos instrumentales para indicar notificaciones y voz para dar instrucciones, de esta manera es más veloz la interacción con la aplicación ya que no se requiere de voz para todas las indicaciones. Por ejemplo, cuando el usuario está decidiendo entre que opción del menú tomar, se escucha un corto sonido metálico que permite al jugador decidir de manera inmediata, evitando que llegue a hartarse por el exceso de sonidos.

Sección	Evento
Inicio aplicación	Bienvenida al usuario
Menú principal	Opciones del menú Alerta de cambio de opción Opción seleccionada Verificación de la opción seleccionada Confirmación de la opción seleccionada
Nivel	Bienvenida al nivel Abejas Abeja interceptada Pausa Opciones menú pausa Alerta de cambio de opción Opción seleccionada Verificación de la opción seleccionada Confirmación de la opción seleccionada Nivel completado Nivel fallado Juego terminado

Tabla 3.4: Lista de eventos que requieren reproducir audio para dar una retroalimentación adecuada al usuario.

3.4 Módulo Mundo Virtual

La aplicación tiene como base un mundo virtual en el cuál el usuario interactúa con los dispositivos. El mundo virtual tiene una representación visual para poder asegurarse de que no existen incongruencias al correr el programa, este modo se puede considerar de prueba, ya que el usuario final no podrá apreciar la pantalla y no es necesario. Sin

embargo, dicha representación visual también puede ser útil para que en un principio personas videntes puedan asistir en cualquier momento al jugador con discapacidad visual mientras se familiariza con el sistema, o bien, para que los videntes puedan también seguir el desarrollo del juego y divertirse junto con el usuario. A través de *OpenGL* se describe el mundo virtual para obtener la visualización y además tener una relación entre las coordenadas del usuario y las de la aplicación.

3.4.1 Escena y abejas

El escenario es el área de interacción del usuario y tiene 4 paredes que abarcan del punto $(-1.33, -1.0, -1.0)$ al punto $(1.33, 1.0, 4.0)$ dejando solamente sin pared al frente y atrás para permitir el paso de las abejas. En la siguiente sección, 3.5, se explica el objetivo del juego y el papel que juegan las abejas. Para poder utilizar las paredes como referencia del avance de las abejas se utiliza una rejilla de color blanca sobre fondo negro como se muestra en la Figura 3.11.

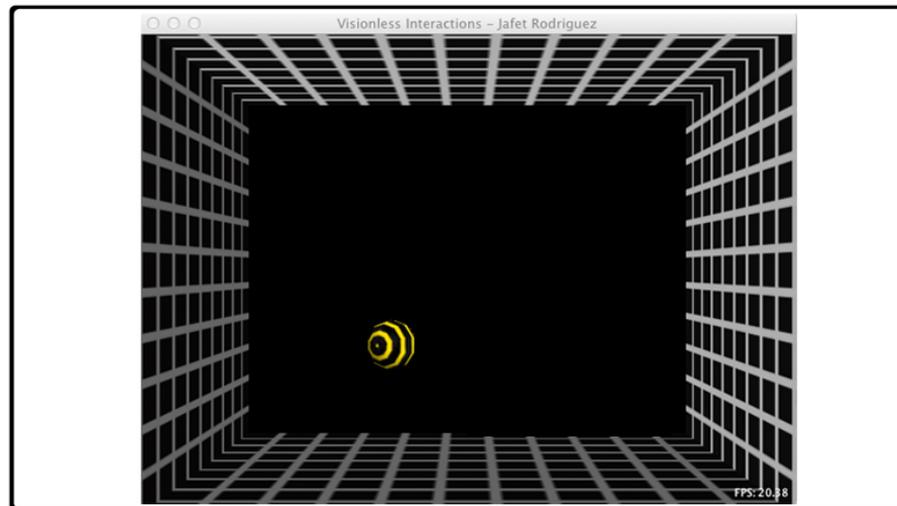


Figura 3.11: Imagen del escenario.

Los objetos que podrán ser interceptados son esferas de radio 0.1 con una textura amarilla a rayas negras para aparentar una abeja, estas abejas pueden únicamente desplazarse, no giran ni cambian de tamaño ya que sería complicado dar a entender tantos comportamientos a través de sonido. El área del escenario por el cual se estarán desplazando las

abejas, abarca desde el punto $(-1.22, -0.9, -1.0)$ al punto $(1.22, 0.9, 4.1)$ como se muestra en la Figura 3.10.

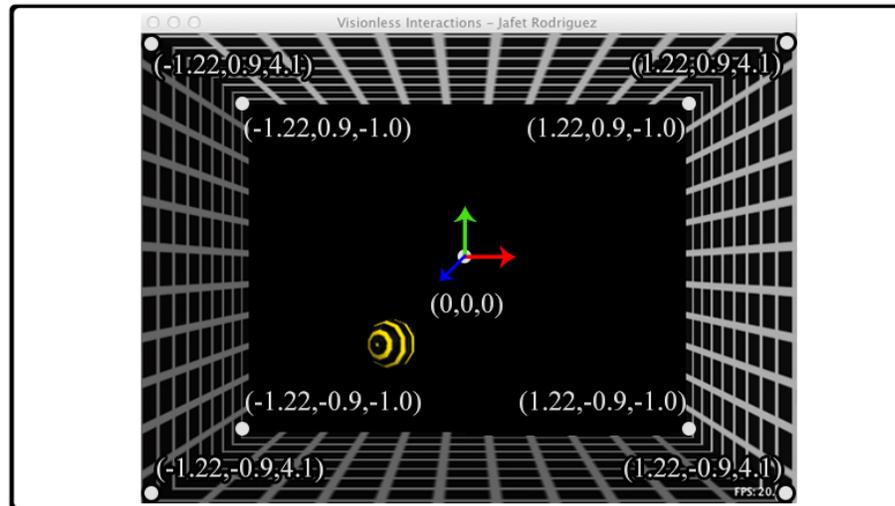


Figura 3.12: Imagen del escenario con su sistema de coordenadas.

3.4.2 Relación coordenadas virtuales 3D a pixeles

Para poder calcular la intersección del usuario con cada abeja se requiere poder obtener el rectángulo de pixeles que cubre a la abeja para comparar esa zona con los datos del *Kinect*. Después de realizar pruebas con 100 posiciones distintas se determinó empíricamente que se deben seguir las siguientes operaciones:

1. Obtener el origen de la esfera (x, y, z) según el mundo virtual
2. Obtener el factor de ajuste de posición, $ajuste_{pos} = 27z + 150$
3. Obtener el factor de ajuste de tamaño, $ajuste_{tam} = 40z + 150$
4. Obtener el centro de la pantalla en $pantallaMedio_X$ y $pantallaMedio_Y$
5. El inicio en x del rectángulo de pixeles es dado por $init_X = 640 - |((((x)(ajuste_{pos})) + pantallaMedio_X) - ((radio)(ajuste_{tam}))) + 2|$
6. El inicio en y del rectángulo de pixeles es dado por $init_Y = |(((((-y)(ajuste_{pos})) + pantallaMedio_Y) - ((radio)(ajuste_{tam}))) - 15|$

7. El fin en x del rectángulo de pixeles es dado por $fin_x = |init_x - ((2radio)(ajuste_{tam}))|$
8. El fin en y del rectángulo de pixeles es dado por $fin_y = |init_y + ((2.4radio)(ajuste_{tam}))|$
9. Al tener el rectángulo de pixeles se recorre uno por uno verificando si la profundidad del pixel dada por $profundidad = 100z$ tiene una diferencia máxima de 5cm
10. Si el punto anterior es verdadero, entonces se determina que existe colisión del usuario con la abeja

De esta manera se puede calcular velozmente si alguna abeja fue interceptada, ya que solamente se debe calcular una parte de la pantalla por cada abeja en lugar de comparar toda la imagen.

3.5 Módulo Juego

El objetivo del juego es interceptar la mayor cantidad de abejas virtuales, con cualquier parte del cuerpo, cuando estas se desplazan dentro de un escenario a diferentes velocidades y de las cuales el usuario solamente percibe el sonido de su vuelo. En ningún momento el sentido de la vista podrá ser usado como recurso. Se tienen 11 niveles de dificultad y 3 tipos distintos de trayectoria, al final de cada nivel el total de abejas virtuales interceptadas tiene que ser mayor o igual a 80 % para poder pasar al siguiente nivel.

El juego fue diseñado para ayudar a personas con discapacidad visual a familiarizarse más con la ubicación exacta de sonidos a través de la intercepción de la mayor cantidad de abejas posible. Las abejas vuelan del frente hacia el usuario en diversas velocidades y rutas forzando al usuario a prestar mayor atención a los sonidos para poder ganar.

3.5.1 Configuración del entorno físico

Ya que el usuario no puede usar su vista para identificar cuál es la zona permitida para jugar y además no colisionar con otros objetos, es necesario delimitar esta área de manera física para que el jugador disfrute al máximo la experiencia y no tenga que preocuparse

por chocar o dañar parte del ambiente. Existen una variedad de opciones para resolver el problema planteado previamente, algunas de ellas son acordonar el área, poner paredes falsas, mandar un sonido o poner indicaciones en el suelo.

Para esta tesis se decidió usar cinta aislante, la decisión de usar cinta se tomó con base a la portabilidad del juego ya que es más sencillo cargar las bocinas, el *Kinect*, la laptop, cinta métrica y la cinta aislante, que tener que cargar otros objetos para delimitar el área de juego. La cinta se coloca en el suelo para permitir que el usuario al pisarla o arrastrar el pie sobre ella se de cuenta del borde. Además, para asegurar que ciertas áreas fueran fáciles de reconocer se hace un doblez en la cinta, por lo que es más sencillo identificarlas, ya que tiene una textura diferente. En la Figura 3.13 se ilustra en donde se debe poner cinta negra y en la Figura 3.14 se muestra como se ve el entorno físico ya configurado.

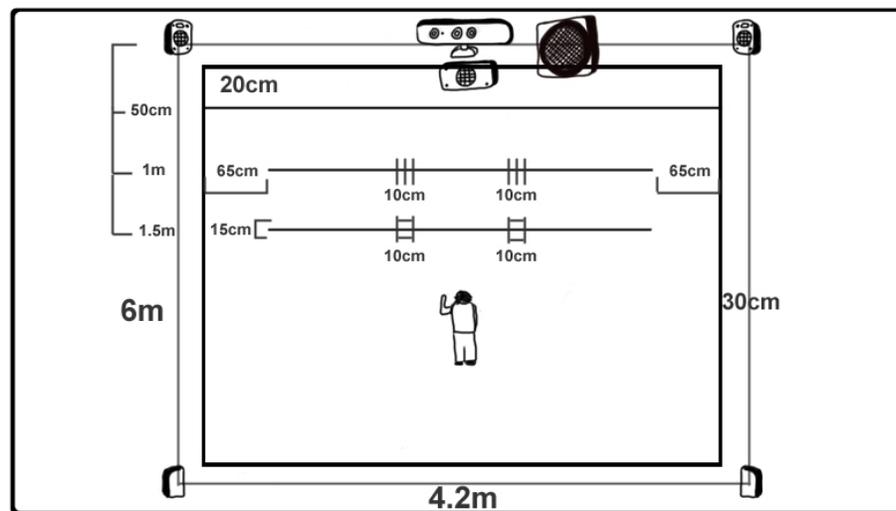


Figura 3.13: Ilustración de como debe quedar el entorno una vez configurado.

El tiempo total aproximado en colocar las bocinas, el *Kinect*, la laptop y la cinta aislante es de 20 minutos. Este tiempo puede parecer demasiado, pero es importante recordar que en la mayoría de los casos se espera tener un área de juego fija destinada para usar con esta aplicación, por lo que en teoría sólo se hará la configuración total del entorno físico en una ocasión y no cada que se desee jugar.



Figura 3.14: Imagen del escenario real con el entorno ya configurado.

3.5.2 Menús

El juego consta de 3 menús:

- **Menú Principal:** Está encargado de manejar la pantalla inicial en el cuál existen 4 opciones: Nuevo Juego, Continuar Juego, Opciones y Tutorial. En la Figura 3.15 se ilustra su acomodo.

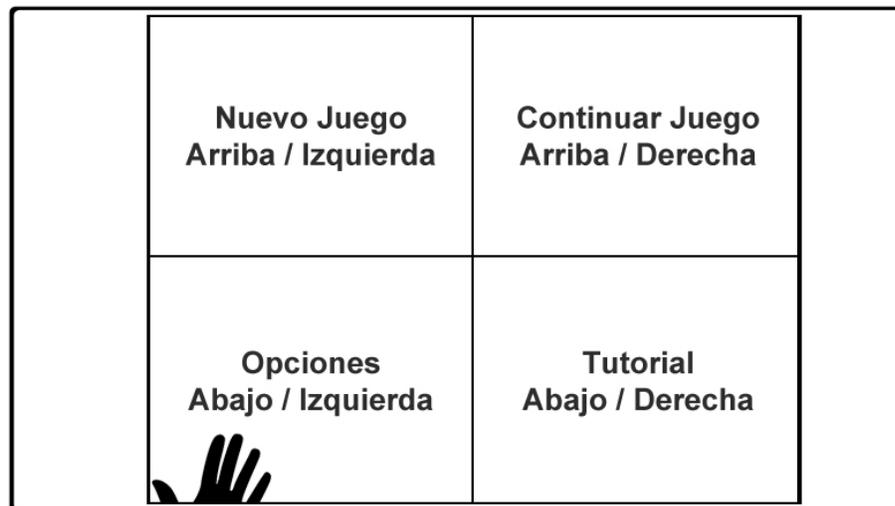


Figura 3.15: Representación del Menú Principal.

- **Menú Pausa:** Está encargado de manejar la pantalla cuando el usuario decide pausar el juego, existen 4 opciones: Reanudar Juego, Reiniciar Nivel, Opciones y

Menú Principal. En la Figura 3.16 se ilustra su acomodo.



Figura 3.16: Representación del Menú Pausa.

- **Menú Confirmación:** Está encargado de manejar la pantalla cuando el usuario ha elegido una opción y la tiene que confirmar, existen 2 opciones: Aceptar y Cancelar. En la Figura 3.17 se ilustra su acomodo.



Figura 3.17: Representación del Menú Confirmación.

Las acciones que realiza cada opción son las siguientes:

- **Nuevo Juego:** Reescribe el archivo de preferencias para que el nivel actual sea el número 1 e inicia el juego.

- **Continuar Juego:** Lee el archivo de preferencias para determinar el nivel actual e inicia el juego en el nivel encontrado.
- **Opciones:** Actualmente no existen opciones, regresa al Menú anterior.
- **Tutorial:** Se reproduce el audio que explica las reglas del juego.
- **Reanudar Juego:** Continúa el juego en la misma parte que se quedó antes de entrar a pausa.
- **Reiniciar Nivel:** Detiene la ejecución del juego y vuelve a cargar el mismo nivel desde el inicio.
- **Menú Principal:** Detiene la ejecución del juego y regresa al menú principal.

Para interactuar con los menús el usuario debe mover una extremidad o algún objeto entre los 1.5 y 2.0 metros de distancia del *Kinect* como se explica en la sección 3.2.5.

3.5.3 Niveles

El juego está dividido en 11 niveles, se comienza con una dificultad sencilla y cada nivel aumenta su dificultad cambiando la cantidad de abejas, su posición original, su posición destino y el tipo de ruta que se sigue. En los primeros niveles las abejas se mueven únicamente en la profundidad para permitir que el jugador poco a poco asimile el juego y configure mentalmente la relación entre sonido y posición. El programa está diseñado para poder tener múltiples abejas simultáneamente, pero después de realizar algunas pruebas fue evidente que la combinación de sonidos evita el poder identificar de donde viene cada uno, por lo que las abejas vuelan una después de la otra para que puedan ser identificadas.

El rango de posiciones válidas para las abejas es en x de -1.0 a 1.0, en y de -0.9 a 0.9 y en z de -1 a 4.1. Los valores aleatorios que se generan para varios niveles utilizan estos rangos para no generar abejas fuera del área de juego, los demás niveles también usan este rango en x,z pero en y se utiliza el rango de -0.9 a 0.4 para que el usuario no tenga que

brincar para interceptar a las abejas.

Uno de los elementos que ayudan a aumentar la dificultad del juego, es la ruta o trayectoria que seguirán las abejas. El programa está diseñado e implementado para poder incluir múltiples tipos de caminos y velocidades. En nuestro caso se implementaron 3 tipos de rutas diferentes con la misma velocidad y que se ilustran en la Figura 3.18 y que describimos a continuación:

1. Línea Recta: Es la ruta más sencilla ya que la abeja viaja en línea recta del origen hacia el destino.
2. En V: En esta ruta la abeja comienza descendiendo del origen hasta la mitad de la distancia y después asciende hasta el destino .
3. En Λ : En esta ruta la abeja comienza ascendiendo del origen hasta la mitad de la distancia y después desciende hasta el destino.

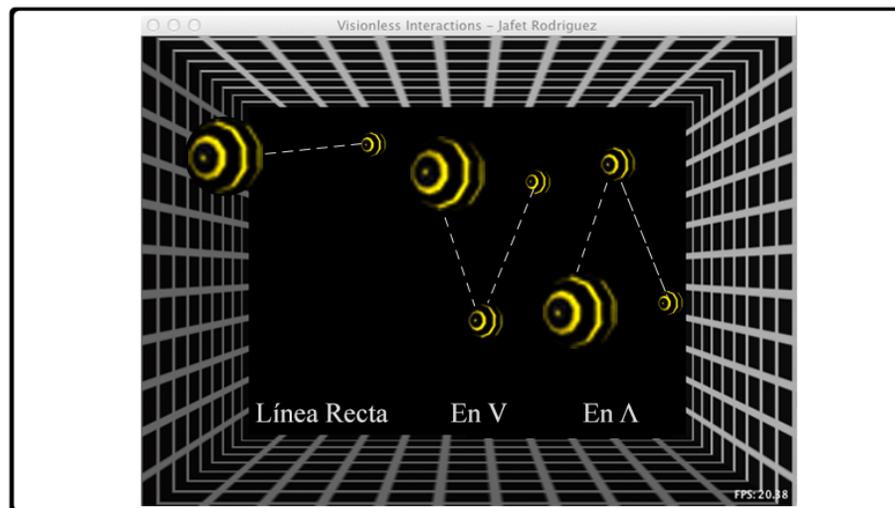


Figura 3.18: Representación de los tipos de rutas.

A continuación, en las Tablas 3.5 a 3.15 se describen las características de las rutas a seguir por las abejas virtuales en cada uno de los 11 niveles del juego.

Nivel 1			
#	Origen	Destino	Ruta
1	(-1.0,-0.9,-4)	(-1.0,-0.9,4.6)	Recta
2	(1.0,-0.9,-4)	(1.0,-0.9,4.6)	Recta
3	(-1.0,0.4,-4)	(-1.0,0.4,4.6)	Recta
4	(1.0,0.4,-4)	(1.0,0.4,4.6)	Recta
5	(0.3,0.0,-4)	(0.3,0.0,4.6)	Recta
2 series - Total 10 abejas			

Tabla 3.5: Descripción del nivel 1.

Nivel 2			
#	Origen	Destino	Ruta
1	(-1.00,-0.9,-4)	(-1.00,-0.9,4.6)	Recta
2	(-0.3,0.0,-4)	(-0.3,0.0,4.6)	Recta
3	(1.00,0.4,-4)	(1.00,0.4,4.6)	Recta
4	(-1.00,0.4,-4)	(-1.00,0.4,4.6)	Recta
5	(0.5,0.0,-4)	(0.5,0.0,4.6)	Recta
6	(1.00,-0.9,-4)	(1.00,-0.9,4.6)	Recta
2 series - Total 12 abejas			

Tabla 3.6: Descripción del nivel 2.

Nivel 3			
#	Origen	Destino	Ruta
1	(-0.22,-0.7,-4)	(-0.22,-0.7,4.6)	Línea Recta
2	(0.22,-0.4,-4)	(0.22,-0.4,4.6)	Línea Recta
3	(-0.22,0.4,-4)	(-0.22,0.4,4.6)	Línea Recta
4	(0.22,0.4,-4)	(0.22,0.4,4.6)	Línea Recta
5	(-0.22,0.4,-4)	(-0.22,0.4,4.6)	Línea Recta
6	(0.22,-0.7,-4)	(0.22,-0.7,4.6)	Línea Recta
7	(-0.22,-0.4,-4)	(-0.22,-0.4,4.6)	Línea Recta
8	(0.22,0.4,-4)	(0.22,0.4,4.6)	Línea Recta
2 series - Total 16 abejas			

Tabla 3.7: Descripción del nivel 3.

Nivel 4			
#	Origen	Destino	Ruta
1	(Aleatorio, Aleatorio, -4)	(Origen, Origen, 4.6)	Línea Recta
2	(Aleatorio, Aleatorio, -4)	(Origen, Origen, 4.6)	Línea Recta
3	(Aleatorio, Aleatorio, -4)	(Origen, Origen, 4.6)	Línea Recta
4	(Aleatorio, Aleatorio, -4)	(Origen, Origen, 4.6)	Línea Recta
5	(Aleatorio, Aleatorio, -4)	(Origen, Origen, 4.6)	Línea Recta
6	(Aleatorio, Aleatorio, -4)	(Origen, Origen, 4.6)	Línea Recta
7	(Aleatorio, Aleatorio, -4)	(Origen, Origen, 4.6)	Línea Recta
8	(Aleatorio, Aleatorio, -4)	(Origen, Origen, 4.6)	Línea Recta
9	(Aleatorio, Aleatorio, -4)	(Origen, Origen, 4.6)	Línea Recta
2 series - Total 18 abejas			

Tabla 3.8: Descripción del nivel 4.

Nivel 5			
#	Origen	Destino	Ruta
1	(-1.00,-0.9,-4)	(0.6,0.0,4.6)	Línea Recta
2	(1.00,-0.9,-4)	(-1.00,0.4,4.6)	Línea Recta
3	(-1.00,0.4,-4)	(1.00,-0.9,4.6)	Línea Recta
4	(1.00,0.4,-4)	(-1.00,-0.9,4.6)	Línea Recta
5	(-0.6,0.0,-4)	(1.00,0.4,4.6)	Línea Recta
2 series - Total 10 abejas			

Tabla 3.9: Descripción del nivel 5.

Nivel 6			
#	Origen	Destino	Ruta
1	(-1.00,-0.9,-4)	(0.4,0.0,4.6)	Línea Recta
2	(0.4,0.0,-4)	(-1.00,0.4,4.6)	Línea Recta
3	(1.00,0.4,-4)	(0.4,0.0,4.6)	Línea Recta
4	(-1.00,0.4,-4)	(1.00,-0.9,4.6)	Línea Recta
5	(0.5,0.0,-4)	(-1.00,-0.9,4.6)	Línea Recta
6	(1.00,-0.9,-4)	(1.00,0.4,4.6)	Línea Recta
2 series - Total 12 abejas			

Tabla 3.10: Descripción del nivel 6.

Nivel 7			
#	Origen	Destino	Ruta
1	(-0.22,-0.7,-4)	(-0.66,0.4,4.6)	Línea Recta
2	(0.22,-0.4,-4)	(0.22,-0.7,4.6)	Línea Recta
3	(-0.22,0.4,-4)	(-0.66,0.4,4.6)	Línea Recta
4	(0.22,0.4,-4)	(0.22,-0.4,4.6)	Línea Recta
5	(-0.22,0.4,-4)	(-0.22,-0.7,4.6)	Línea Recta
6	(0.22,-0.7,-4)	(0.66,0.4,4.6)	Línea Recta
7	(-0.22,-0.4,-4)	(0.22,0.4,4.6)	Línea Recta
8	(0.22,0.6,-4)	(-0.66,-0.4,4.6)	Línea Recta
2 series - Total 16 abejas			

Tabla 3.11: Descripción del nivel 7.

Nivel 8			
#	Origen	Destino	Ruta
1	(Aleatorio, Aleatorio, -4)	(Aleatorio, Aleatorio, 4.6)	Línea Recta
2	(Aleatorio, Aleatorio, -4)	(Aleatorio, Aleatorio, 4.6)	Línea Recta
3	(Aleatorio, Aleatorio, -4)	(Aleatorio, Aleatorio, 4.6)	Línea Recta
4	(Aleatorio, Aleatorio, -4)	(Aleatorio, Aleatorio, 4.6)	Línea Recta
5	(Aleatorio, Aleatorio, -4)	(Aleatorio, Aleatorio, 4.6)	Línea Recta
6	(Aleatorio, Aleatorio, -4)	(Aleatorio, Aleatorio, 4.6)	Línea Recta
7	(Aleatorio, Aleatorio, -4)	(Aleatorio, Aleatorio, 4.6)	Línea Recta
8	(Aleatorio, Aleatorio, -4)	(Aleatorio, Aleatorio, 4.6)	Línea Recta
9	(Aleatorio, Aleatorio, -4)	(Aleatorio, Aleatorio, 4.6)	Línea Recta
2 series - Total 18 abejas			

Tabla 3.12: Descripción del nivel 8.

Nivel 9			
#	Origen	Destino	Ruta
1	(-1.00,-0.9,-4)	(-1.00,-0.9,4.6)	En V
2	(1.00,-0.9,-4)	(1.00,-0.9,4.6)	En V
3	(-1.00,0.4,-4)	(-1.00,0.4,4.6)	En V
4	(1.00,0.4,-4)	(1.00,0.4,4.6)	En V
5	(0.3,0.0,-4)	(0.3,0.0,4.6)	En V
2 series - Total 10 abejas			

Tabla 3.13: Descripción del nivel 9.

Nivel 10			
#	Origen	Destino	Ruta
1	(-1.00,-0.9,-4)	(-1.00,-0.9,4.6)	En Λ
2	(1.00,-0.9,-4)	(1.00,-0.9,4.6)	En Λ
3	(-1.00,0.4,-4)	(-1.00,0.4,4.6)	En Λ
4	(1.00,0.4,-4)	(1.00,0.4,4.6)	En Λ
5	(0.6,0.0,-4)	(0.6,0.0,4.6)	En Λ
2 series - Total 10 abejas			

Tabla 3.14: Descripción del nivel 10.

Nivel 11			
#	Origen	Destino	Ruta
1	(Aleatorio, Aleatorio, -4)	(Aleatorio, Aleatorio, 4.6)	Aleatorio
2	(Aleatorio, Aleatorio, -4)	(Aleatorio, Aleatorio, 4.6)	Aleatorio
3	(Aleatorio, Aleatorio, -4)	(Aleatorio, Aleatorio, 4.6)	Aleatorio
4	(Aleatorio, Aleatorio, -4)	(Aleatorio, Aleatorio, 4.6)	Aleatorio
5	(Aleatorio, Aleatorio, -4)	(Aleatorio, Aleatorio, 4.6)	Aleatorio
6	(Aleatorio, Aleatorio, -4)	(Aleatorio, Aleatorio, 4.6)	Aleatorio
7	(Aleatorio, Aleatorio, -4)	(Aleatorio, Aleatorio, 4.6)	Aleatorio
8	(Aleatorio, Aleatorio, -4)	(Aleatorio, Aleatorio, 4.6)	Aleatorio
9	(Aleatorio, Aleatorio, -4)	(Aleatorio, Aleatorio, 4.6)	Aleatorio
10	(Aleatorio, Aleatorio, -4)	(Aleatorio, Aleatorio, 4.6)	Aleatorio
2 series - Total 20 abejas			

Tabla 3.15: Descripción del nivel 11.

3.5.4 Proceso del juego

El juego comienza con audio describiendo que se trata de un proyecto de tesis, nombra al autor, la institución en la que se realiza la tesis y la maestría, después le solicita al usuario que salga del área para poder identificar el fondo como se explica en la sección 3.2.3. Dependiendo del equipo de cómputo en el que se este corriendo el programa este proceso inicial puede tardar de 15 segundos hasta 1 minuto y medio. Al terminar el reconocimiento del fondo se le hace saber al usuario que la detección fue exitosa y prosigue a describir el menú principal y sus opciones.

Cada nivel comienza especificando el número de nivel en que se encuentran y después comienzan a volar las abejas hasta la última, entonces el sistema determina si se interceptó el mínimo del 80 % de abejas del nivel requeridas para pasar al siguiente nivel, en caso contrario se le informa al usuario que no tuvo éxito y se reinicia el nivel. Estando en el último nivel, el nivel 11 y una vez alcanzando el mínimo de abejas interceptadas, se le informa al usuario que logró con éxito terminar el juego. En la Tabla 3.16 se especifica cuantas abejas existen en cada nivel y cuantas deben de ser interceptadas para pasar al siguiente nivel.

Nivel	# Abejas en el nivel	# Abejas a interceptar
1	10	8
2	12	10
3	16	13
4	18	15
5	10	8
6	12	10
7	16	13
8	18	15
9	10	8
10	10	8
11	20	16
Total	152	124

Tabla 3.16: Cantidad mínima de abejas que se deben interceptar para pasar cada nivel.

CAPÍTULO 4

Experimentación

4.1 Diseño de las pruebas

Para poder realizar una verificación inicial de que el programa cumplió con los objetivos planteados inicialmente, se llevaron a cabo una serie de pruebas con varias personas de distintas edades. Debido al acuerdo al que se llegó con la institución de donde se seleccionaron la mayoría de los participantes, no se hará mención del nombre de la institución ni de las personas que participaron en las pruebas, por lo mismo no se tiene ninguna grabación de video o fotografías con los participantes. Simplemente podemos mencionar que dicha institución está dedicada a la educación de invidentes y débiles visuales.

Buscando tener una variedad en las edades, sexo y capacidad visual se reclutó un grupo de 13 voluntarios para realizar las pruebas, los participantes tienen desde 4 hasta 63 años, 62 % mujeres y 38 % hombres, la mitad con discapacidad visual y la otra mitad utilizó un antifaz para simular la falta de visión del resto de participantes.

A cada participante antes de iniciar se le explicó cómo era el entorno físico, la existencia de la cinta negra para delimitar las áreas, cómo funciona el juego, cuál es el objetivo de cada nivel para poderlo pasar, cómo funcionan los menús y se le ayudó a recorrer el área. Lo que no se les mencionó fue cuántas abejas existen ni en qué ruta vuelan. El tiempo que se asignó para cada participante fue de 15 minutos debido a los horarios que se manejan en la institución y la disponibilidad de ambas partes, después de esos 15 minutos a cada

participante se le pidió que jugara en el nivel 11 una ocasión para determinar su dificultad.

4.1.1 Cuestionario

Después de jugar por 15 minutos y al terminar de jugar el último nivel, a cada participante se le aplicó un cuestionario para poder analizar su experiencia obtenida con el juego y obtener así retroalimentación que nos permita posteriormente irlo mejorando. Las primeras preguntas del cuestionario son del tipo de escala nominal y las últimas de escala ordinal para poder obtener algunas de sus medidas de tendencia central.

1. ¿Consideras el juego divertido?
2. ¿Consideras que el juego te podría ayudar a identificar mejor la posición de un objeto cuando hace un sonido?
3. ¿Consideras que fueron útiles las guías del suelo?
4. ¿Volverías a jugarlo?
5. ¿Te gustaría que existieran otros juegos parecidos a este por diversión?
6. ¿Te gustaría que existieran otros juegos parecidos a este para mejorar tu percepción de la posición de una fuente de sonido?
7. ¿Te sentiste restringido por el espacio que se tiene para jugar?
8. ¿Te sentiste en riesgo de salirte del área de juego y lastimarte?

En una escala del 1 al 5, siendo 1 muy fácil y 5 muy difícil

9. ¿Qué tan difícil fue usar el juego?
10. ¿Qué tan difícil fueron los primeros 4 niveles?
11. ¿Qué tan difícil fueron los niveles después del 4?
12. ¿Qué tan difícil fue interceptar las abejas en los primeros minutos?

13. ¿Qué tan difícil fue interceptar las abejas en los últimos minutos?
14. ¿Qué tan difícil fue usar los menús?
15. ¿Qué tan difícil fue hacer pausa durante el juego?

4.1.2 Ambiente

Todas las pruebas se realizaron en una computadora portátil para poder correr el programa en las distintas locaciones en las que se encontraban los participantes, además para poder demostrar que el sistema puede correr en una maquina que no fue adquirida recientemente y por lo tanto se puede correr el juego en computadoras de capacidades similares sin necesidad de adquirir un equipo nuevo. En la Tabla 4.1 se muestra la especificación técnica del *hardware* usado en las pruebas y en la Tabla 4.2 el *software*.

<i>Hardware</i>	
Marca	<i>Apple</i>
Modelo	<i>MacBook Pro - Mid 2009(5,5)</i>
Procesador	<i>Intel Core 2 Duo - 2.53GHz</i>
Arquitectura	<i>x86_64 - 2 núcleos</i>
RAM	<i>8GB - DDR3 1333MHz</i>
Video	<i>nVidia GeForce 9400M - 256MB</i>

Tabla 4.1: Especificación técnica del *hardware* usado en las pruebas.

<i>Software</i>	
Sistema Operativo	<i>Mac OS X - 10.7.4</i>
<i>Java</i>	1.6.0_33 - 64 Bits
<i>OpenCV</i>	v2.4.2
<i>JavaCV</i>	v0.2
<i>JOGAMP</i>	v2.0rc10
<i>LWJGL</i>	v2.8.4
<i>JNA</i>	v3.4.1
<i>Eclipse</i>	<i>Juno v3.8</i>

Tabla 4.2: Especificación técnica del *software* usado en las pruebas.

4.1.3 Recopilación automática de los datos

Al realizar las pruebas es complicado poder asegurar que las mediciones sean correctas si los datos son recopilados por una persona, ya que existen errores humanos y se puede registrar de manera incorrecta lo que se está observando. Para evitar la situación anterior el programa cuenta con una interfaz gráfica para poder verificar que todo funciona en orden y también para recabar los datos.

La interfaz cuenta con las siguientes 3 ventanas:

1. **Ventana de juego:** Esta ventana está encargada de mostrar las imágenes de inicio, la pantalla durante reconocimiento del fondo, la pantalla del menú principal y la escena de *OpenGL* mientras se está jugando. En la Figura 4.1 se muestran las pantallas al iniciar la aplicación y en la Figura 4.2 la ventana del juego.



Figura 4.1: Imagen de las pantallas de inicio.



Figura 4.2: Imágenes de la ventana del juego.

2. **Ventana del *Kinect*:** Esta ventana sólo se activa cuando se desea ver la imagen binarizada final que utiliza el juego para determinar las coordenadas del usuario, esto debido al hecho de que mostrar esta imagen de manera continua reduce la velocidad del juego en casi 20fps.
3. **Ventana de la consola:** Esta ventana está encargada de registrar toda la información importante y mantener actualizado el archivo que tendrá dichos datos. En la Figura 4.3 se muestran las 3 ventanas durante la ejecución.

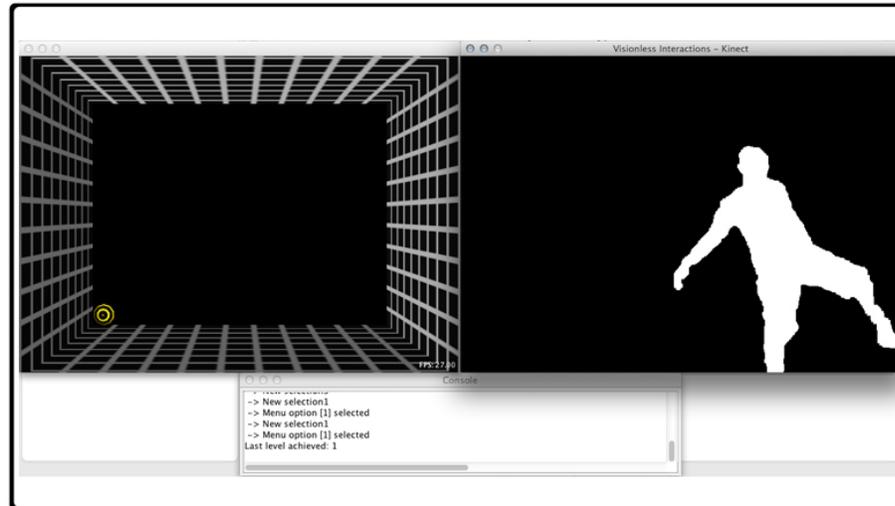


Figura 4.3: Imagen de las 3 ventanas durante la ejecución del programa.

Como se mencionó previamente, la ventana de la consola guarda toda la información a un archivo para después poder evaluar lo ocurrido durante la ejecución. El archivo incluye información como las decisiones tomadas en cada menú, si se logró interceptar o no a la abeja, entre otros datos. En la Figura 4.4 se muestra un ejemplo de un archivo de log.

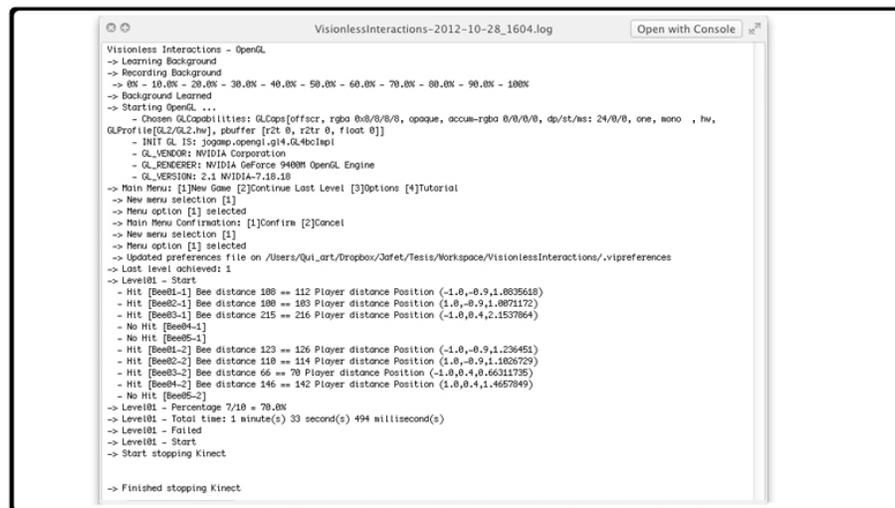


Figura 4.4: Imagen de ejemplo de un archivo de log guardado automáticamente por la ventana de la consola.

4.2 Resultados obtenidos

#	N-01	N-02	N-03	N-04	N-05	N-06	N-07	N-08	N-09	N-10	N-11
1	8	9	0	0	0	0	0	0	0	0	4
2	8	9	0	0	0	0	0	0	0	0	5
3	8	10	10	0	0	0	0	0	0	0	8
4	8	11	13	14	0	0	0	0	0	0	10
5	8	10	13	15	8	8	0	0	0	0	10
6	8	10	14	16	9	8	0	0	0	0	10
7	8	10	13	15	8	8	0	0	0	0	9
8	8	11	14	16	9	10	10	0	0	0	11
9	8	10	13	11	0	0	0	0	0	0	9
10	8	10	13	12	0	0	0	0	0	0	9
11	8	10	10	0	0	0	0	0	0	0	9
12	8	10	10	0	0	0	0	0	0	0	7
13	8	10	10	0	0	0	0	0	0	0	5

Tabla 4.3: Número máximo de abejas interceptadas en la mejor ocasión por cada uno de los 13 participantes en cada uno de los 11 niveles.

#	Edad	Sexo	Antifaz	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	4	F	S	S	N	S	S	S	N	N	S	3	3	4	3	1	2	2
2	7	M	N	S	N	S	S	S	S	N	S	2	3	4	4	2	2	2
3	9	F	N	S	S	S	S	S	S	N	S	2	3	5	3	1	2	2
4	14	M	N	S	S	S	S	S	S	N	N	2	4	5	5	2	2	2
5	14	F	N	S	S	S	S	S	S	N	N	3	3	5	3	2	2	2
6	19	F	S	S	S	S	S	S	S	N	N	2	5	5	2	1	2	2
7	20	M	S	S	S	S	S	S	S	S	S	2	2	4	3	1	2	2
8	20	F	N	S	S	S	S	S	S	N	N	1	3	4	4	2	2	2
9	25	F	S	S	S	S	S	S	S	N	S	3	3	4	4	2	2	1
10	27	M	S	S	S	S	S	S	S	N	N	2	4	4	3	1	2	1
11	55	M	N	S	S	S	S	S	S	S	S	4	3	5	4	2	2	2
12	56	F	S	S	S	S	S	S	S	S	S	4	4	4	3	1	2	2
13	63	F	N	S	S	S	S	S	S	S	S	3	3	5	5	2	2	2

Tabla 4.4: Respuestas al cuestionario dadas por cada uno de los 13 participantes del estudio.

Nivel	Promedio	Moda	Mínimo	Máximo	Máximo posible
1	8.00	8	8	8	10
2	10.00	10	9	11	12
3	10.23	13	0	14	16
4	7.62	0	0	16	18
5	2.62	0	0	9	10
6	2.62	0	0	10	12
7	0.77	0	0	10	16
8	0.00	0	0	0	18
9	0.00	0	0	0	10
10	0.00	0	0	0	10
11	8.15	9	4	11	20

Tabla 4.5: Datos estadísticos del número máximo de abejas interceptadas en la mejor ocasión por cada nivel y considerando a todos los usuarios que participaron en el estudio.

Parámetro	Valor
Edad promedio	25.62
Edad mínima	4
Edad máxima	63
Invidente	54 %
Vidente con antifaz	46 %
Promedio nivel alcanzado	4.08
Mínimo nivel alcanzado	2
Máximo nivel alcanzado	7
Hombres	38 %
Mujeres	62 %

Tabla 4.6: Datos estadísticos generales obtenidos durante un tiempo de juego de 15 minutos por parte de cada uno de los participantes.

Parámetro	Hombres	Mujeres
Edad promedio	24.6	26.25
Invidente	60 %	50 %
Vidente	40 %	50 %
Promedio nivel	3.8	4.25
Mínimo nivel alcanzado	2	2
Máximo nivel alcanzado	6	7

Tabla 4.7: Datos estadísticos generales dividido en hombres y mujeres.

Parámetro	Invidentes	Videntes
Edad promedio	26	25.16
Edad mínima	7	4
Edad máxima	63	56
Hombres	60 %	40 %
Mujeres	50 %	50 %
Promedio nivel	4	4.17
Mínimo nivel alcanzado	2	2
Máximo nivel alcanzado	7	6

Tabla 4.8: Datos estadísticos generales de invidentes versus videntes.

Parámetro	1	2	3	4	5	6	7	8
Sí	100 %	85 %	100 %	100 %	100 %	92 %	31 %	62 %
No	0 %	15 %	0 %	0 %	0 %	8 %	69 %	38 %
Hombres Sí	100 %	80 %	100 %	100 %	100 %	100 %	40 %	60 %
Hombre No	0 %	20 %	0 %	0 %	0 %	0 %	60 %	40 %
Mujeres Sí	100 %	88 %	100 %	100 %	100 %	88 %	25 %	63 %
Mujeres No	0 %	13 %	0 %	0 %	0 %	13 %	75 %	38 %

Tabla 4.9: Datos estadísticos del cuestionario de las preguntas 1 a 8.

Pregunta	Promedio	Moda	Mínimo	Máximo
9	2.54	2.00	1.00	4.00
10	3.54	3.00	2.00	5.00
11	4.46	4.00	4.00	5.00
12	3.54	3.00	2.00	5.00
13	1.54	2.00	1.00	2.00
14	2.00	2.00	2.00	2.00
15	1.85	2.00	1.00	2.00

Tabla 4.10: Datos estadísticos del cuestionario de las preguntas 9 a 15.

4.2.1 Interpretación

Al revisar los resultados obtenidos que se muestran de la Tabla 4.3 a la 4.10, se puede decir que el 100 % de los participantes disfrutaron del juego a pesar de que el 31 % se sintió restringido por el espacio y el 62 % en riesgo de lastimarse. Al 100 % les fueron útiles las guías del suelo, volverían a jugarlo y les gustaría que existieran juegos parecidos por diversión. El 85 % considera que el juego le puede ayudar a identificar mejor su posición y el 92 % le gustaría que existieran más juegos parecidos para mejorar su percepción del

ambiente.

Los participantes consideraron retador el juego ya que en promedio ponderaron la dificultad de los primeros niveles en 3.54 y los siguientes niveles con 4.46, además de que confirmaron que entre más tiempo pasaban jugando más se familiarizaban con el juego, al identificar los primeros minutos con una dificultad de 4.46 y en los últimos minutos con 3.54, una diferencia de casi el 20 %. Asimismo se demostró que el sistema de menús y la pausa que se diseñaron cumplieron su propósito al obtener un promedio de 2.0 de dificultad en la navegación en los menús y 1.85 para efectuar una pausa.

Resalta a la vista que en promedio lograron llegar a un nivel más alto las personas con antifaz, pero al ver con detenimiento los datos se identifica que en realidad el grupo de personas que no puede ver tiene más participantes de edad avanzada, lo que les dificulta el poderse mover con facilidad para interceptar las abejas que vuelan con una ruta más agresiva, por lo tanto las personas jóvenes de ese grupo lograron mejores resultados que su contraparte que puede ver pero que usaron antifaz.

CAPÍTULO 5

Conclusiones

En la actualidad existe una gran cantidad de juegos de mesa y videojuegos, estos suelen ser una manera entretenida de pasar el tiempo o incluso una forma eficaz y divertida de aprender, ya que desde una perspectiva positiva motivan al jugador a lograr metas a través de retos que se proponen para ir avanzando en los diferentes niveles del juego, además de que van aprendiendo a no dejar una tarea inconcluso, sino hasta que esta esté completamente terminada. Lamentablemente una gran mayoría de estos juegos no están orientados a personas que no pueden usar su vista y que se pudieran beneficiar de las ventajas de estos.

Uno de los retos más grandes que tienen las personas discapacitadas visualmente es la navegación, cuando caminamos estamos evadiendo obstáculos sin darnos cuenta de que hacemos uso de nuestra vista, oído, olfato y tacto. La vista es el sentido que nos provee con la información más exacta sobre donde podríamos colisionar y así poder planear una ruta para evitarlo.

Con base a lo anterior se decidió crear *Visionless Interactions*, un plataforma que permite crear juegos para personas que no pueden ver. Para demostrar que este sistema ha funcionado exitosamente en esta primera etapa y que puede ayudar a un invidente o débil visual a mejorar su proceso de ubicación de objetos en su entorno tridimensional mediante el reconocimiento de la posición espacial del sonido producido por dichos objetos, se creó un juego orientado a reconocer la ubicación de un objeto virtual para poder interceptarlo y por ende lograr que la persona pueda navegar con mayor facilidad haciendo uso de los

sonidos del ambiente.

El presente trabajo mostró que es posible realizar una aplicación que utilice como entrada video en tiempo real, que sea multiplataforma y utilice dispositivos de acceso al público utilizando el lenguaje de programación *Java*. La aplicación funciona a más de 30 cuadros por segundo utilizando la configuración original de la máquina virtual de Java, por lo que no se requiere una gran cantidad de memoria RAM. Los dispositivos que se usaron fueron una *Apple MacBook Pro 2009*, *Kinect* de *Microsoft*, Sonido 5.1 de *Logitech* y una tarjeta de sonido USB 7.1 de *Sabrent*.

Durante la creación de la plataforma se decidió usar bocinas en lugar de audífonos con sonido binaural, debido a que el usuario está en constante movimiento y el uso de audífonos implicaría el tener además que alinear el sonido generado en el mundo virtual con la posición exacta del jugador en el mundo real. Por otro lado, también se tomó la decisión de usar el *Kinect* para no depender de la calibración de cámaras ni las variaciones de luz, de esta manera el usuario tiene una mejor experiencia.

En los cuestionarios aplicados se muestra que el usuario se divirtió y con el paso del tiempo fue logrando identificar de mejor manera los objetos 3D de su alrededor. Asimismo el juego funcionó de manera adecuada al permitir que el jugador navegara entre menús y pudiera interactuar con los objetos 3D. Además, se logró que con un sonido 5.1 el usuario pudiera crear un mapa mental de la posición de objetos en el espacio, aunque para no tener que simular la altura de los objetos modificando el tono del sonido, se debe usar una mayor cantidad de bocinas en la siguiente fase.

Finalmente, resaltamos que el mercado de aplicaciones para gente discapacitada visualmente sigue siendo tierra virgen, *Visionless Interactions* permite acortar la brecha tecnológica existente al proporcionar una plataforma que facilita la creación de videojuegos para múltiples sistemas operativos sin necesidad de una pantalla. El desarrollo de nuevos juegos utilizando esta plataforma es sencillo ya que su arquitectura divide adecua-

damente el manejo de los distintos dispositivos, el procesamiento de imágenes y el control del juego; así, el programador únicamente se tiene que enfocar a crear los diferentes niveles y su interacción.

CAPÍTULO 6

Trabajo futuro

Visionless Interactions es una plataforma funcional y se puede seguir mejorando al expandir su funcionalidad y facilitar más su uso. Para comenzar, se requiere realizar dos series de pruebas de campo con una muestra significativa. La primera es para conocer lo que el usuario final experimenta y opina del juego de ejemplo. La segunda está enfocada a los desarrolladores que usarán la plataforma, es indispensable el conocer cómo la usarían y qué puntos de mejora reconocen.

Además, es importante tener datos de desarrolladores y usuarios finales de diversos países, el Internet permite el jugar y hacer negocios con personas de distintas culturas, edades y géneros sin importar su ubicación en el mundo. Con esta información se puede tropicalizar la plataforma y los juegos resultantes, por lo que se puede incentivar a los usuarios y programadores a probar esta plataforma al mostrarles la variedad de juegos que se pueden crear y el tamaño del mercado que les espera.

Con base a los resultados obtenidos en el punto anterior, se puede trabajar en las siguientes mejoras:

- **Sonido:** Realizar una investigación formal acerca de la mejor manera de reproducir el sonido y que ayude a mejorar la identificación de la posición de un sonido. Por lo que se debe de indagar el uso de una mayor cantidad de bocinas contra el desarrollo de un sistema que permita usar audífonos con sonido binaural de manera inalámbrica para no limitar los movimientos del usuario.

- **Múltiples jugadores:** Crear juegos que involucren varios jugadores para analizar si es posible que varias personas puedan replicar el aprendizaje del entorno como en las aplicaciones de un jugador. Se comenzaría creando juegos sencillos como *pong* y se aumentaría la dificultad hasta crear juegos complejos de deportes o de peleas.
- **Juego en línea:** Crear juegos con múltiples jugadores en línea, parecido al punto anterior, la razón detrás de esta mejora es el permitir que los usuarios se motiven entre ellos al poder divertirse sin la necesidad de estar físicamente en el mismo lugar. Esta mejora abre la posibilidad de interactuar con personas de todo el mundo teniendo como único requisito una conexión de Internet.
- **Editor gráfico:** Crear un ambiente de desarrollo gráfico que permita a desarrolladores no experimentados, el crear un juego al arrastrar elementos visuales y tener que agregar pequeñas porciones de código. La meta es aumentar la cantidad de juegos para personas invidentes y además motivar a las generaciones jóvenes a elegir carreras en ingeniería.
- **Compatibilidad con otros motores:** Existen aplicaciones como *Unity 3D*[35] que facilitan el desarrollo de videojuegos al tener una interfaz gráfica fácil de usar y soporte para modelos 3D creados con programas de la industria. La plataforma *Visionless Interactions* se podría adaptar para funcionar como *plugin* para este tipo de motores de juegos. Por lo que se combinaría el poder de creación de juegos del programa y la capacidad de tener como entrada de datos la información procesada del *Kinect*. Por lo que usuarios actuales de *Software* como *Unity 3D* podrían comenzar de inmediato a tomar provecho de esta plataforma.

Bibliografía

- [1] G. e. I. M. Instituto Nacional de Estadística, *Las personas con discapacidad en México: una visión censal*. Aguascalientes, Aguascalientes, México: Instituto Nacional de Estadística, Geografía e Informática, 2004.
- [2] W. Strouse Watt, O.D., “How visual acuity is measured,” Oct. 2003. <http://www.mdsupport.org/library/acuity.html>
- [3] S. D. Pinder and T. C. Davies, “Exploring direct downconversion of ultrasound for human echolocation,” in *CHINZ '07 Proceedings of the 7th ACM SIGCHI New Zealand chapter's international conference on Computer-human interaction: design centered HCI*. New York, N.Y.: Association for Computing Machinery, 2007.
- [4] T. Papadopoulos, D. S. Edwards, D. Rowan, and R. Allen, “Identification of auditory cues utilized in human echolocation - objective measurement results,” in *Information Technology and Applications in Biomedicine, 2009. ITAB 2009. 9th International Conference on*. IEEE, Nov. 2009, pp. 1–4. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5394427>
- [5] Microsoft, “What is kinect?” <http://www.xbox.com/en-US/Kinect?xr=shellnav>
- [6] “Kinect sports,” USA, Nov. 2010. <http://xbox360.ign.com/objects/077/077448.html>
- [7] “Kinect adventures,” USA, Nov. 2010. <http://xbox360.ign.com/objects/077/077456.html>
- [8] “XBOX 360.” <http://www.xbox.com/en-US/Xbox360?xr=shellnav>
- [9] “Kinect hacks.” <http://www.kinecthacks.net/>

- [10] “Body dysmorphic disorder.” <http://www.kinecthacks.net/body-dysmorphic-disorder/>
- [11] Microsoft, “The kinect effect: How the world is using kinect,” 2012. <http://www.xbox.com/en-US/Kinect/Kinect-Effect>
- [12] J. Blake, “OpenKinect.” <http://openkinect.org/>
- [13] M. Sotirios and P. Georgios, “Computer vision method for pianist’s fingers information retrieval,” in *iiWAS ’08 Proceedings of the 10th International Conference on Information Integration and Web-based Applications Services*. ACM Press, 2008, p. 604. <http://bit.ly/FPuc9q>
- [14] D. Quam, “Gesture recognition with a dataglove,” in *Aerospace and Electronics Conference, 1990. NAECON 1990., Proceedings of the IEEE 1990 National*, may 1990, pp. 755 –760 vol.2. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=112862>
- [15] K. Terajima, T. Komuro, and M. Ishikawa, “Fast finger tracking system for in-air typing interface,” in *CHI EA ’09 Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*. ACM Press, 2009, p. 3739. <http://bit.ly/wljTLE>
- [16] A. Corp, “FPGAs.” <http://www.altera.com/products/fpga.html>
- [17] J. Sánchez and A. Tadres, “Audio and haptic based virtual environments for orientation and mobility in people who are blind,” in *ASSETS ’10 Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility*. ACM Press, 2010, p. 237. <http://bit.ly/x49gMq>
- [18] M. Zöllner, S. Huber, H.-C. Jetter, and H. Reiterer, “Navi: a proof-of-concept of a mobile navigational aid for visually impaired based on the microsoft kinect,” in *Proceedings of the 13th IFIP TC 13 international conference on Human-computer*

- interaction - Volume Part IV*, ser. INTERACT'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 584–587. <http://0-dl.acm.org.millennium.itesm.mx/citation.cfm?id=2042283.2042382>
- [19] R. C. Gonzalez and R. E. Woods, *Digital image processing*. Upper Saddle River, N.J.: Prentice Hall, 2002.
- [20] J. Villegas and M. Cohen, “HRIR : modulating range in headphone-reproduced spatial audio,” in *Proceedings of the 9th ACM SIGGRAPH Conference on Virtual-Reality Continuum and its Applications in Industry*, ser. VRCAI '10. New York, NY, USA: ACM, 2010, pp. 89–94. <http://bit.ly/GzUh6J>
- [21] T. Ltd, “THX technology.” <http://www.thx.com/consumer/thx-technology/>
- [22] D. Laboratories, “Dolby surround sound.” <http://www.dolby.com/us/en/consumer/technology/surround-sound.html>
- [23] S. R. Buss, *3D computer graphics a mathematical introduction with OpenGL*. Cambridge: Cambridge University Press, 2003.
- [24] R. S. Wright, B. Lipchak, and N. Haemel, *OpenGL superbible : comprehensive tutorial and reference*. Upper Saddle River, NJ; Munich [u.a.]: Addison-Wesley, 2007.
- [25] JogAmp.org, “JOGAMP.” <http://jogamp.org/jogl/www/>
- [26] M. A. Weisfeld, *The object-oriented thought process*. Upper Saddle River, NJ: Addison-Wesley, 2009.
- [27] S. Microsystems, “Java.” <http://www.java.com/en/>
- [28] P. J. Deitel, P. J. Deitel, and H. M. Deitel, *Java : how to program*. Upper Saddle River, N.J.: Prentice Hall, 2012.
- [29] M. A. Weisfeld, *The object-oriented thought process*. Upper Saddle River, NJ: Addison-Wesley, 2009.
- [30] ROS.org, “Technical kinect.” http://www.ros.org/wiki/kinect_calibration/technical

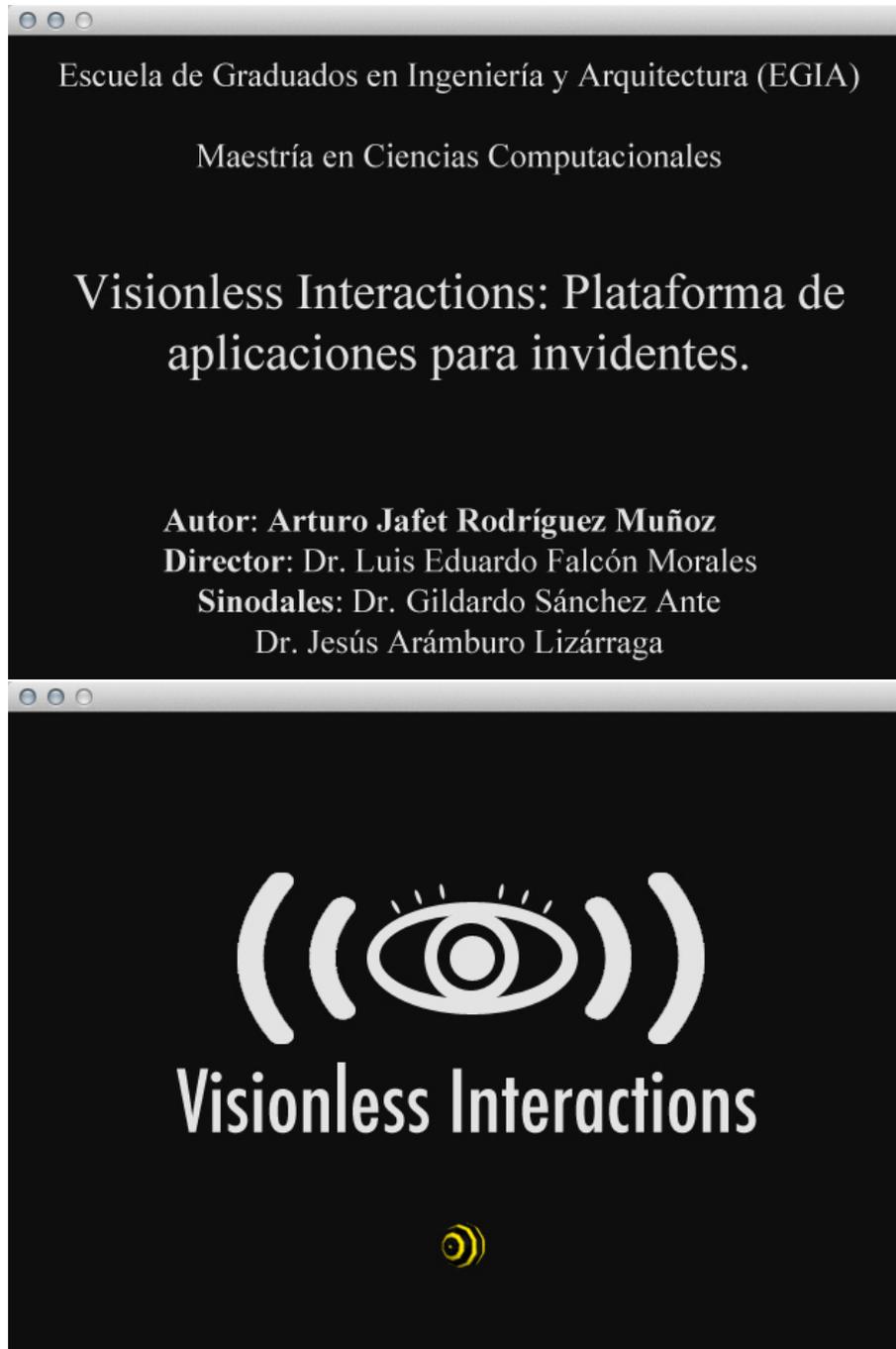
-
- [31] (2011, January). <http://goo.gl/7G1Ui>
- [32] (2011, February). <http://mathnathan.com/2011/02/depthvsdistance/>
- [33] P. Lamb, “Paul 3d sound system.” <http://www.paulscode.com/>
- [34] C. Rychlik Prince, “Lightweight java game library,” November 2011. <http://lwjgl.org/>
- [35] U. Technologies. Unity 3d. <http://unity3d.com/>

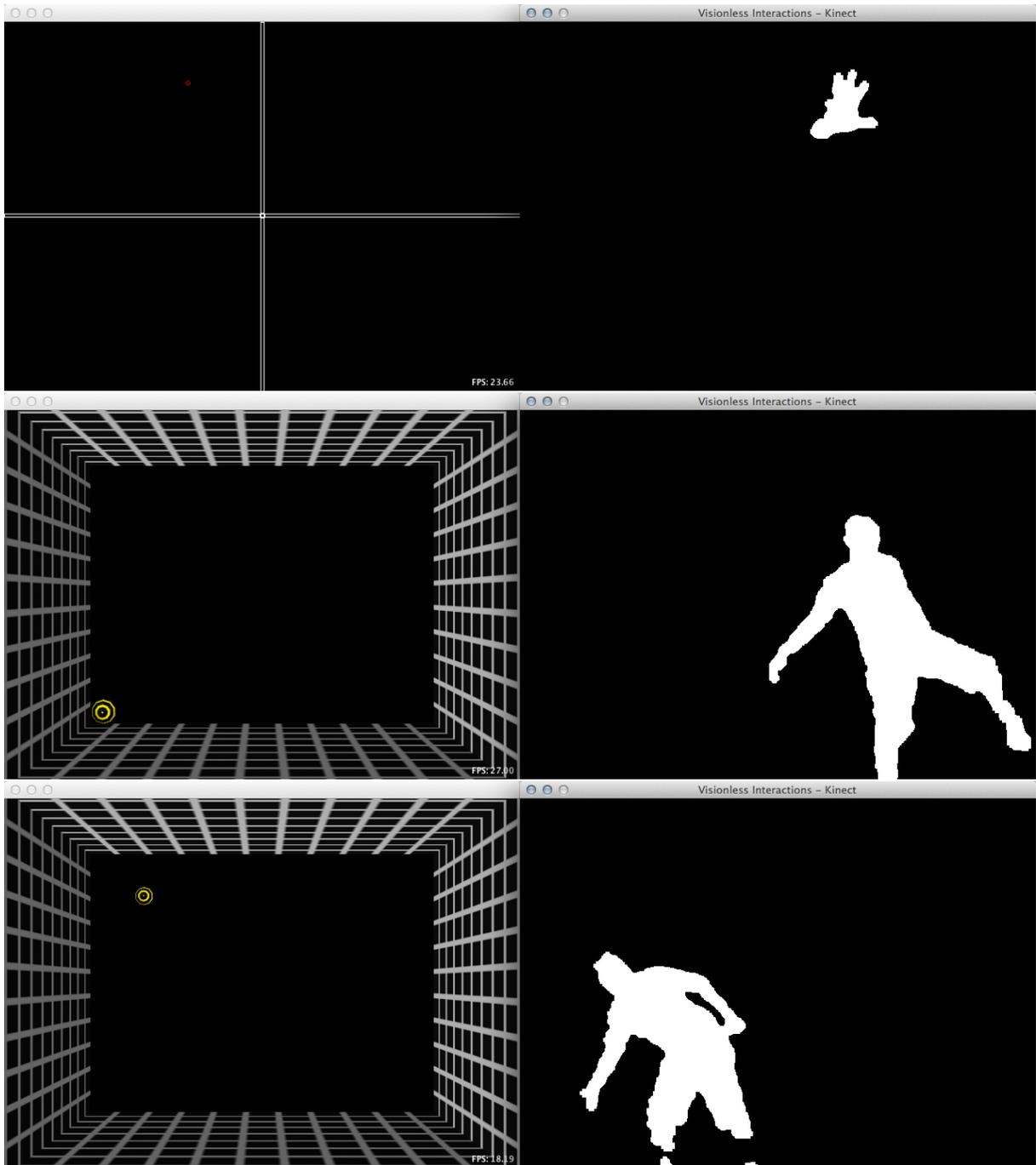
APÉNDICE A

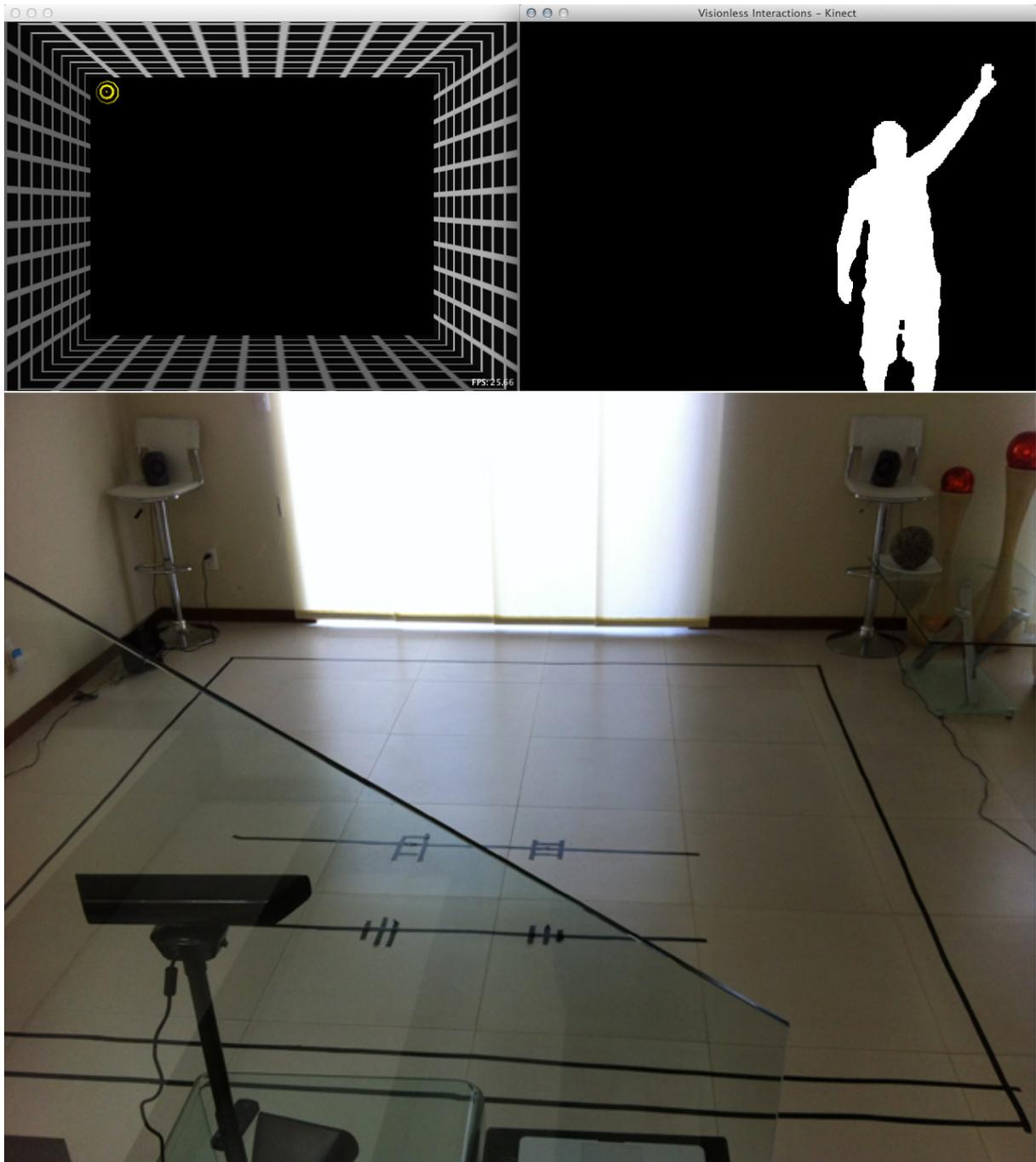
Imágenes



Instituto Tecnológico y de Estudios
Superiores de Monterrey
Campus Guadalajara











APÉNDICE B

Sonidos usados

Sonido	URL
Abeja	http://www.freesound.org/samplesViewSingle.php?id=73372
Abeja aplastada	http://www.freesound.org/people/Flasher21/sounds/87535/
Fondo Menú Tutorial	http://www.freesound.org/people/kerri/sounds/42932/
Cambio Opción Menú	http://www.freesound.org/people/reinsamba/sounds/35631/
Nuevo Nivel	http://www.freesound.org/people/reinsamba/sounds/35631/
Fondo Fin del Juego	http://www.freesound.org/people/reinsamba/sounds/35631/
Fondo Fin del Juego	http://www.freesound.org/people/HerbertBoland/sounds/30167/
Fondo Fin del Juego	http://www.freesound.org/people/dobroide/sounds/5560/
Pausa	http://www.freesound.org/people/HardPCM/sounds/30668/
Fondo Pausa	http://www.freesound.org/people/kerri/sounds/15778/
Fondo Pausa	http://www.freesound.org/people/kerri/sounds/35730/
Sonido Aceptar Menú	http://www.freesound.org/people/nkuitse/sounds/18654/
Sonido Cancelar Menú	http://www.freesound.org/people/gezortenplotz/sounds/18522/