



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

UTILIZACIÓN DE REALIDAD AUMENTADA E INTERFACES BASADAS EN AUDIO PARA FACILITAR
LA MOVILIDAD Y ORIENTACIÓN DE PERSONAS CIEGAS

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

ANGELO RODRIGO TADRES BUSTAMANTE

PROFESOR GUÍA
JAIME SÁNCHEZ ILABACA

MIEMBROS DE LA COMISIÓN
BENJAMÍN BUSTOS CARDENAS
LUIS MATEU BRULE

SANTIAGO DE CHILE
SEPTIEMBRE 2010

Resumen

Hoy en día la ciencia de la computación se encuentra presente en nuestras vidas de forma cotidiana, siendo muchas veces transparente para nosotros. Un área que se ha visto potenciada por esta ciencia es la que tiene relación con apoyar el desarrollo de habilidades de Movilidad y Orientación por medio de audio. Otra área es la asociada a la Realidad Aumentada, variación de lo que se conoce como Realidad Virtual, donde en lugar de introducir completamente a un usuario en un entorno ficticio se busca introducir elementos virtuales en el entorno real.

Esta memoria tuvo como objetivo diseñar, desarrollar y evaluar un sistema que, utilizando Realidad Aumentada e interfaces basadas en audio, pueda asistir a un usuario ciego en tópicos de Movilidad y Orientación. Como resultado se obtuvo ARTAB, un sistema capaz de identificar ciertos objetos de interés de un entorno real mediante un dispositivo de captura de video, y entregar así información, basada en audio, que permite conocer e identificar su entorno como si lo estuvieran viendo.

Para identificar la utilidad del sistema, se realizaron evaluaciones de funcionalidad y usabilidad. La funcionalidad del sistema fue constantemente evaluada durante el desarrollo para asegurar el correcto desempeño final. La usabilidad se aplicó durante y al finalizar el proceso de desarrollo, de manera cualitativa y cuantitativa con usuarios finales, utilizando los métodos de observación y evaluación de usuario final. Estas evaluaciones permitieron detectar y solucionar tempranamente problemas de representación de la información que entrega el sistema, así como también de interacción.

Los resultados obtenidos muestran que ARTAB es una aplicación útil que podrá cumplir con ayudar a usuarios ciegos en tareas de Movilidad y Orientación. Además, la percepción de los usuarios sobre ARTAB es buena, debido a que todos los usuarios participantes de las evaluaciones quedaron motivados y entusiasmados con el sistema.

Agradecimientos

A mis padres Patty y Oscar, quienes junto a Lilian y toda mi familia me apoyaron incondicionalmente.

A mis amigos Jorge y Nicolás, por su constante motivación.

A todo el equipo de C5, en especial a Mauricio, Héctor, Catalina, Javiera, Juan Pablo, Marina, Matías, Natalia y mi profesor guía JS.

Índice

Capítulo 1: Introducción.....	1
1.1 Descripción general del problema.....	1
1.2 Motivación	3
1.3 Objetivos	3
1.3.1 Objetivo general	3
1.3.2 Objetivos específicos.....	3
Capítulo 2: Trabajo relacionado	4
2.1 Realidad Aumentada	4
2.1.1 Técnicas de visualización.....	6
2.1.2 Handheld displays	7
2.1.3 Spatial Augmented Reality displays.....	7
2.1.4 Aplicaciones	8
2.2 Tecnologías de apoyo para reforzar habilidades de M&O.....	15
2.2.1 Aplicaciones de entrenamiento	18
2.2.2 Aplicaciones de asistencia.....	20
2.3 Metodologías de usabilidad.....	21
Capítulo 3: Diseño de ARTAB	23
3.1 Consideraciones preliminares	23
3.2 Descripción general.....	24
3.3 Modelo	24
3.4 Decisiones de diseño	26
Capítulo 4: Desarrollo de ARTAB.....	31
4.1 Herramientas de desarrollo.....	31

4.1.1	Hardware	31
4.1.2	Software	32
4.1.3	ARToolKit.....	33
4.2	Diagramas de clase.....	35
4.2.1	Componente FacadeAudio	35
4.2.2	Componente FacadeTracker.....	39
4.2.3	ARTAB	42
4.3	Interfaces e interacción.....	46
4.4	Evaluación funcional.....	50
Capítulo 5: Evaluación de la usabilidad		52
5.1	Introducción	52
5.2	Participantes	52
5.3	Instrumentos	53
5.4	Procedimiento	54
5.5	Resultados y análisis de usabilidad para primer prototipo	56
5.5.1	Análisis cualitativo	56
5.5.2	Análisis cuantitativo	58
5.6	Resultados de usabilidad para el segundo prototipo.....	59
5.6.1	Análisis cuantitativo.....	59
Capítulo 6: Conclusiones y trabajo futuro		61
6.1	Conclusiones	61
6.2	Trabajo futuro.....	63
Referencias.....		64
Anexos		71

Anexo 1	71
Anexo 2	73
Anexo 3	74

Índice de figuras

Figura 1: Milgram's Reality-Virtuality Continuum [31].	5
Figura 2: Ejemplo de Virtualidad Aumentada: (A) Conductora del programa posa delante de un fondo azul; (B) Resultado que es transmitido por televisión.	5
Figura 3: Ejemplo de Realidad Aumentada.	6
Figura 4: Ejemplos de HMDs. (A) Dispositivo basado en video; (B) Dispositivo óptico.	7
Figura 5: Ejemplo de un Handheld display.	7
Figura 6: Ejemplos de Spatial Augmented Reality displays: (A) Uso de un monitor de escritorio; (B) Sistema que incluye dispositivo háptico para manipular objetos; (C) El mismo sistema pero ahora los objetos virtuales son desplegados sobre este gracias a un proyector.	8
Figura 7: Modelos de negocio que utilizan AR [1].	9
Figura 8: Captura de pantalla de Augmented Chemistry [2].	10
Figura 9: Estudiantes trabajando con Construct3D [25].	10
Figura 10: Volkswagen, verificación de partes en un modelo [38].	11
Figura 11: Captura de pantalla de EyePet.	12
Figura 12: Portada revista Esquire El hombre que aparece ahí se encuentra sentado en uno de los marcadores que pueden ser utilizados en el sitio web de la revista.	13
Figura 13: Utilización de AR para disminuir accidentes de tránsito.	14
Figura 14: Captura de pantalla de Layar.	15
Figura 15: Validación de hipótesis de AudioDoom [51].	18

Figura 16: Conversión de un entorno real a una representación virtual en AbES. (A) Plano de planta de un edificio (B) Representación virtual del plano de planta desplegada en la interfaz del facilitador.	19
Figura 17: Usuario utilizando See ColOr para detectar colores a través del sonido [12]......	21
Figura 18: Modelo utilizado para el diseño de ARTAB [50][52]	25
Figura 19: Diagrama de flujo simplificado de ARTAB.	27
Figura 20: Ciclo de juego.	29
Figura 21: Estructuración de los componentes de ARTAB.	30
Figura 22: Hardware utilizado por el usuario de ARTAB.	32
Figura 23: Ejemplo de marcador utilizado por ARToolKit.	34
Figura 24: Ejemplo de Umbralización (A) Imagen original (B) Imagen umbralizada.	35
Figura 25: Diagrama de clases de FacadeAudio.	36
Figura 26: Diagrama de clases de FacadeTracker.	39
Figura 27: Diagrama de clases de ARTAB.	42
Figura 28: Funcionamiento general de ARTAB.	46
Figura 29: Interfaz del facilitador del primer prototipo.	47
Figura 30: Interfaz basada en audio del primer prototipo.	48
Figura 31: Interfaz del facilitador del segundo prototipo.	48
Figura 32: Interfaz basada en audio del segundo prototipo.	49
Figura 33: Botonera inalámbrica utilizada en ARTAB.	50
Figura 34: Material de apresto.	53

Figura 35: Usuario realizando apresto.	54
Figura 36: Usuario realizando tarea con ARTAB. (A) El usuario observa el marcador que indica que hay una escalera a la derecha de este; (B) El facilitador observa gracias a una interfaz gráfica, lo que enfoca el usuario.	55
Figura 37: Gráfico de resultados totales de la encuesta de satisfacción (usabilidad) del primer prototipo.	58
Figura 38: Gráfico de resultados totales de la encuesta de satisfacción (usabilidad) del segundo prototipo versus el primero.	59

Índice de tablas

Tabla 1: Pruebas de caja negra.....	51
-------------------------------------	----

Capítulo 1: Introducción

1.1 Descripción general del problema

Una persona que presenta discapacidad visual se encuentra en desventaja comparada con una persona vidente, debido a que le corresponde desenvolverse en un mundo que ha sido diseñado y construido por y para gente que puede ver [43]. En el día a día, una persona podrá notar que existen variadas actividades que requieren como sentido principal la visión para poder ser realizadas de manera satisfactoria [28].

Un ejemplo de una actividad cotidiana, que podría parecer simple, es el desplazarse. Aquí se hace necesario estar constantemente recopilando información a modo de saber el lugar en que la persona se encuentra y la ubicación del punto a donde se quiere llegar, para luego ir tomando decisiones que hacen que el trayecto sea óptimo. A modo de ayudar a una persona ciega surge la pregunta de cómo es posible brindar la información, que una persona vidente adquiere por medio de la visión, de manera eficiente para efectos de su desplazamiento.

Hoy en día la ciencia de la computación se encuentra presente en diversas facetas de nuestras vidas de forma cotidiana, siendo muchas veces transparente para nosotros. Observar como esta ciencia se ha integrado en otras áreas como la medicina, ingeniería o incluso en las artes plásticas, para

potenciarlas en su quehacer, provoca que la computación deba ser vista desde una perspectiva global, considerándola dentro de un conjunto de muchas otras áreas del conocimiento [3]. Una de estas áreas tiene directa relación con el soporte y apoyo al desarrollo de habilidades de Movilidad y Orientación (M&O) en personas ciegas.

Al intentar dar acceso a la información es importante representarla de manera coherente y entendible por quien la solicita. En el caso de las personas que son ciegas se puede generar una representación virtual del espacio real y hacer que interactúen. Luego se puede realizar una transferencia de lo aprendido virtualmente y recorrer el entorno real [45]. Otro modo es tomar ciertos datos del contexto y ayudarlos por medio de un asistente tecnológico que proporcione la información que les permita tomar decisiones [4][61].

Sea la primera o segunda opción, es factible el desarrollo de una interfaz basada en audio, para representar e interactuar con el entorno real o virtual. Con el uso de pistas basadas en audio una persona ciega será capaz de localizar objetos de interés, como lo haría una persona vidente [14]. La creación de estas interfaces eso si requiere de un diseño muy cuidadoso, ya que se hace necesario cautelar que el usuario que la utilice no se sienta saturado por una excesiva cantidad de información [29].

Hace ya un tiempo que se ha acuñado el término de Realidad Aumentada (AR). Este término consiste en el hecho de combinar objetos reales y virtuales en un entorno real [8]. Para ello, se puede capturar una imagen de video del entorno real y, mediante software, disponer objetos virtuales correctamente escalados y orientados en relación a los objetos del entorno real en tiempo real [8][10]. La finalidad de lo anterior es agregar información adicional al contexto real, no quedándose sólo con objetos virtuales representados mediante modelos 3D, sino que también agregando etiquetas de texto, iconos gráficos o sonidos espacializados [8][54].

En esta investigación se presenta ARTAB (*Augmented Reality Tags for Assist the Blind*), un sistema pensado como asistente para que usuarios ciegos puedan identificar, a través de audio espacializado, ciertos objetos de interés dentro de un contexto *indoor*.

El diseño de este tipo de aplicaciones no es una tarea trivial, debido a que el proceso completo debe considerar a los usuarios finales con sus formas de interacción e intereses. Para ello se utilizan metodologías de usabilidad [34][59] en las distintas etapas de diseño y desarrollo del sistema. Con

estas metodologías es posible evaluar las interfaces desarrolladas y la interacción del usuario para luego realizar las mejoras que se requieran.

1.2 Motivación

La motivación de este trabajo de título es integrar el uso de AR en un sistema, en donde se puedan identificar ciertos objetos de interés de un entorno real *indoor* y generar una representación basada en audio espacializado, que permita a un usuario ciego detectar su posición y obtener información relevante asociada a M&O.

1.3 Objetivos

1.3.1 Objetivo general

Diseñar, desarrollar y evaluar un sistema para usuarios ciegos que utilice AR y sea capaz de agregar información al contexto acerca de la posición de objetos y obstáculos importantes, definidos por un facilitador, utilizando interfaces basadas en audio para asistirlos en tópicos de M&O.

1.3.2 Objetivos específicos

Los objetivos específicos son:

- Estudiar el estado del arte de sistemas que integren el uso de AR.
- Diseñar y desarrollar un sistema que integre AR y permita a través de audio espacializado entregar información relevante y ordenada al usuario referente a su M&O.
- Evaluar la funcionalidad del sistema propuesto.
- Evaluar la usabilidad del sistema propuesto.

Capítulo 2: Trabajo relacionado

2.1 Realidad Aumentada

En esta sección se busca introducir al lector el concepto AR y su aplicación concreta en sistemas con variados fines.

En el año 1994 Paul Milgram y Fumio Kishino [31] definen el concepto de *Milgram's Reality-Virtuality Continuum*. Este consiste en la existencia de un continuo que va desde un “Entorno Real” a un “Entorno Virtual”. Entre medio surge lo denominado como *Mixed Reality* que abarca la Virtualidad Aumentada (VA) y AR, como aparece en la Figura 1.

La VA busca integrar objetos del entorno real dentro de un entorno virtual. Un ejemplo de esto puede encontrarse en programas de televisión en los cuales los conductores posan delante de una pantalla de color verde o azul donde, en tiempo real y como aparece en la Figura 2, es reemplazada por un escenario virtual generado por computador.

La AR busca introducir al usuario en una simulación que busca combinar el mundo real con elementos virtuales generados por computador [10][8]. Buscando formalizar este concepto y así

definir cuando un sistema realmente utiliza AR. Ronald Azuma en 1997 define que estos sistemas deben cumplir con las siguientes propiedades [9]:

- Combina objetos reales y virtuales en un entorno real.
- Se ejecuta de forma interactiva y en tiempo real.
- Registra y alinea objetos reales y virtuales.

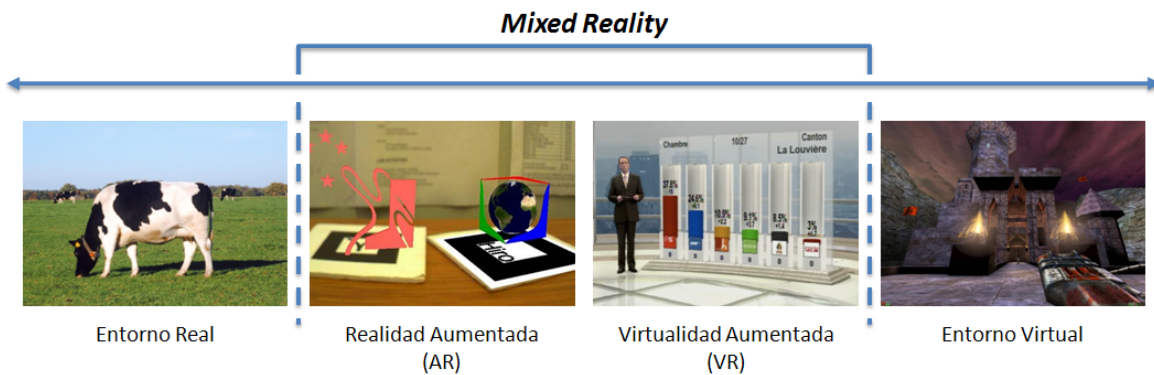


Figura 1: Milgram's Reality-Virtuality Continuum [31].

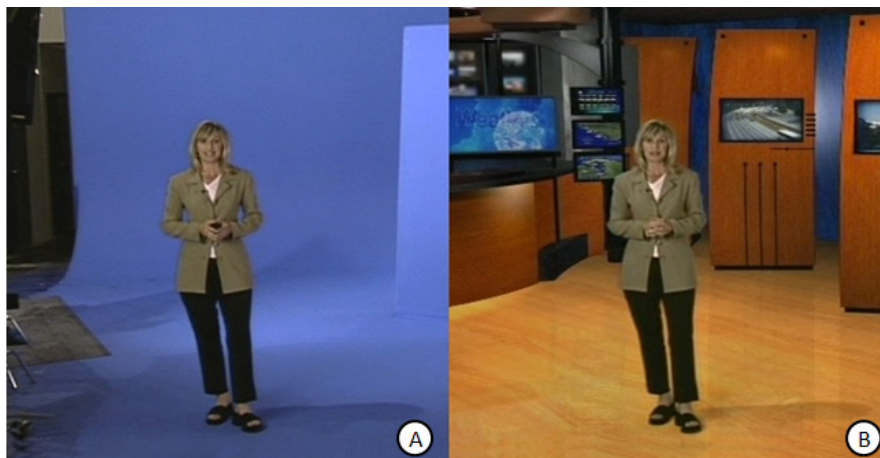


Figura 2: Ejemplo de Virtualidad Aumentada: (A) Conductora del programa posa delante de un fondo azul; (B) Resultado que es transmitido por televisión.

En lo que se refiere a objetos virtuales, estos pueden ser modelos 3D, imágenes, textos o sonido espacializado [54] los cuales se superponen en tiempo real sobre la percepción que el usuario tiene

del mundo real, y que pueden conseguir de manera directa o indirecta un aumento en el conocimiento que el usuario tiene sobre entorno [8].

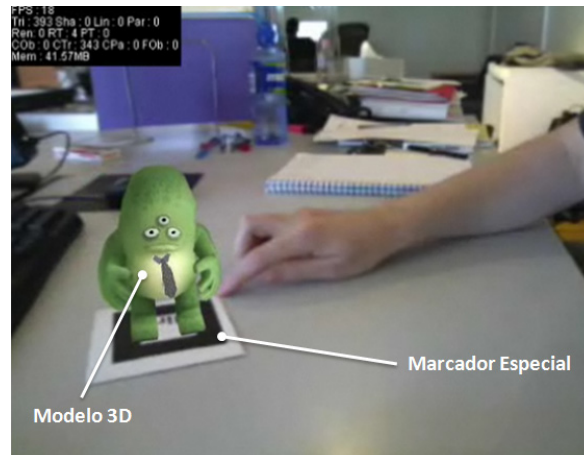


Figura 3: Ejemplo de Realidad Aumentada.

La Figura 3 muestra una aplicación que utiliza AR. A través del uso de un dispositivo de captura de video se registra el entorno real, y sobre este, de manera correctamente alineada y escalada se introduce un modelo virtual 3D. En este caso en particular, para lograr el efecto se utiliza un marcador especial como referencia y sobre este se calculan las dimensiones y orientación.

2.1.1 Técnicas de visualización

Existen distintos tipos de dispositivos que permiten experimentar el uso de AR de manera visual. A continuación se realiza una breve descripción de estos: Los *Head-Mounted displays*, *Handheld displays* y *Spatial Augmented Reality displays* [10].

2.1.1.1 Head-mounted display

Un *Head-mounted display* (HMD) es un dispositivo el cual se monta en la cabeza del usuario y que provee imágenes justo delante de sus ojos. Existen 2 tipos de HMDs: los basados en video y los ópticos. El primero utiliza un dispositivo de captura de video que capta el entorno real, el cual será utilizado como fondo para los objetos virtuales que se agregarán. Todo esto es desplegado en un par de monitores. El segundo usa un par de lentes traslucidos que dejan pasar la luz que refleja el mundo real y sobre esta lente un monitor refleja los objetos virtuales (ver Figura 4)

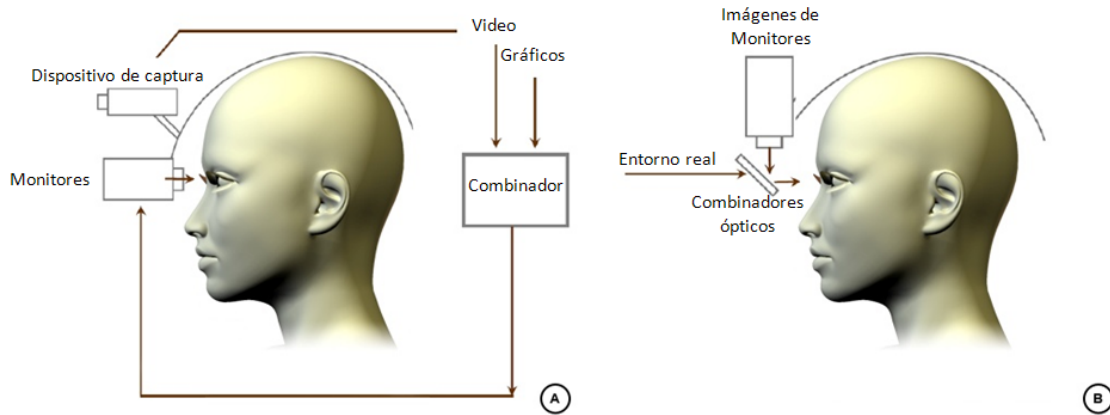


Figura 4: Ejemplos de HMDs. (A) Dispositivo basado en video; (B) Dispositivo óptico.

2.1.2 Handheld displays

Algunos sistemas AR usan un dispositivo *Handheld*, el cual cuenta con una pantalla y un dispositivo de captura de video para ofrecer un funcionamiento similar al que ofrecería un dispositivo HMD basado en video. La pantalla del *Handheld* actúa como una ventana que muestra el mundo real con objetos virtuales superpuestos, tal como parece en la Figura 5, donde un *Pocket PC* con una cámara captura el entorno real y superpone el modelo 3D de un sofá.



Figura 5: Ejemplo de un Handheld display.

2.1.3 Spatial Augmented Reality displays

En contraste a los dispositivos anteriores, los *Spatial Augmented Reality displays* buscan separar la tecnología del usuario e integrarla al entorno.

Estos dispositivos pueden ir desde un simple monitor de escritorio hasta proyectores digitales para mostrar información gráfica sobre los objetos físicos. Debido a que los *displays* no están asociados a cada usuario, permite a los grupos de usuarios utilizarlos de manera conjunta y coordinar así trabajo entre ellos (ver Figura 6).

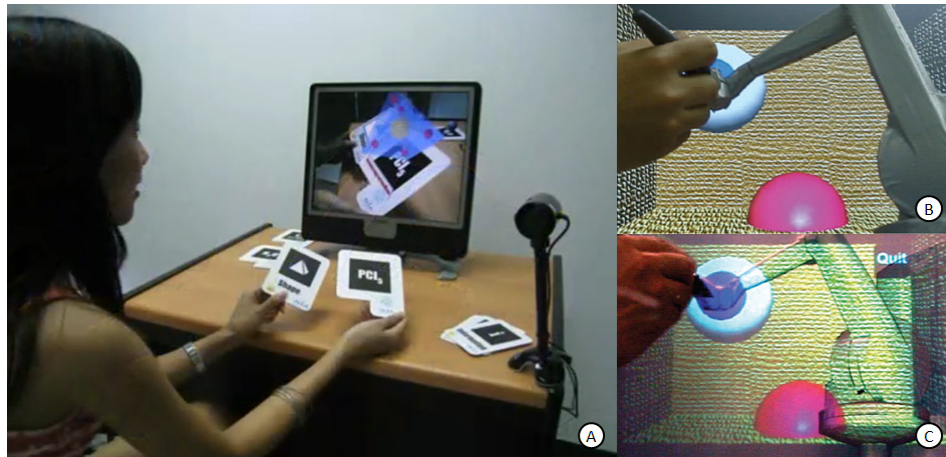


Figura 6: Ejemplos de Spatial Augmented Reality displays: (A) Uso de un monitor de escritorio; (B) Sistema que incluye dispositivo háptico para manipular objetos; (C) El mismo sistema pero ahora los objetos virtuales son desplegados sobre este gracias a un proyector.

2.1.4 Aplicaciones

El uso de AR ha penetrado en distintas áreas. Un ejemplo de esta penetración la propone Gary Hayes en su sitio Web [1], quién diseñó un gráfico, el cual corresponde a la Figura 7, donde busca categorizar los tipos de negocios orientados a sistemas que involucren el uso de AR para poder identificar su oportunidad.

En el gráfico existen 16 modelos que han sido dispuestos según el eje de valor comercial (ingresos probables o comercialización potencial) vs adopción (escala de popularidad frente a un nicho, el cliente de base de usuarios). Según éste, las aplicaciones con mayor futuro comercial son las asociadas a entrenamiento, en donde por ejemplo se puede ir entregando información visual en tiempo real al operario de alguna maquinaria compleja.

Sumado a lo anterior Kroeker [26] publica en un artículo de *Communications of the ACM*, que el enfoque que debe perseguir el uso de AR va de la mano con hacer que la tecnología involucrada para poder llevarla a cabo sea ubicua, además de crear estándares que permitan que cualquier

persona, compañía o organización pueda hacer uso de ésta, creando sus propios sistemas y contenidos.

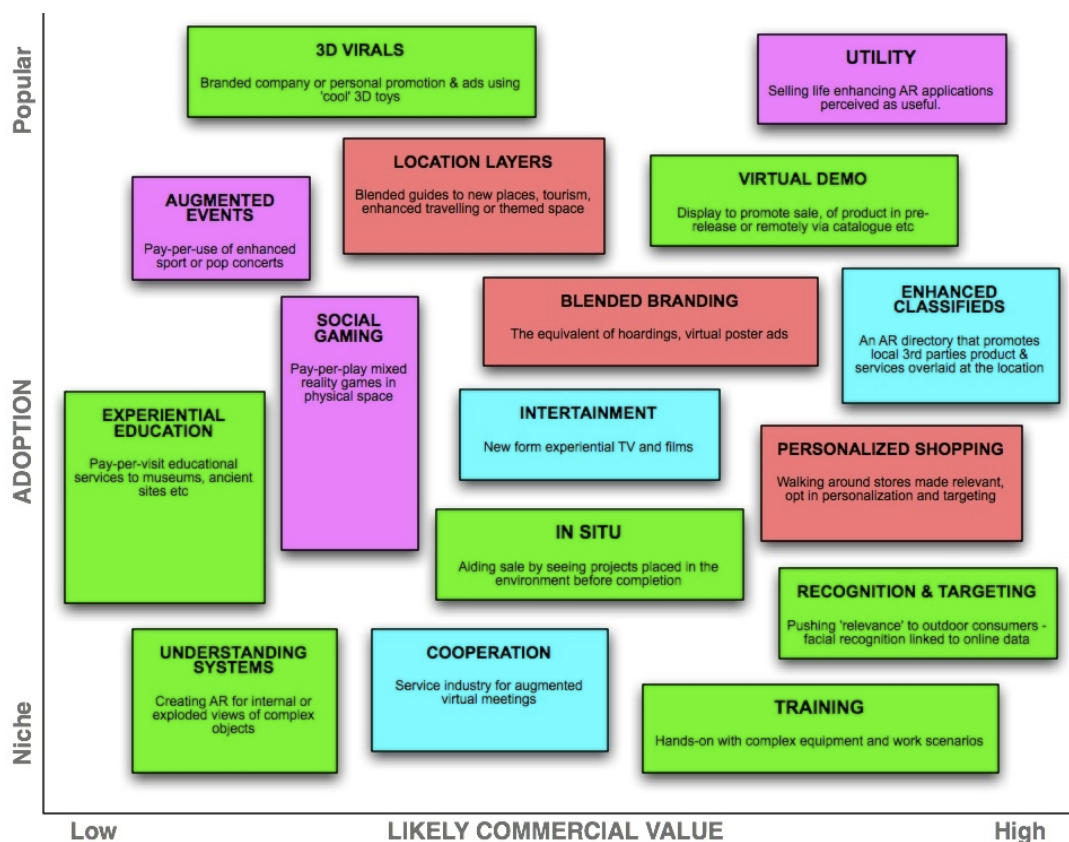


Figura 7: Modelos de negocio que utilizan AR [1].

A continuación se profundizará en áreas como la educación, industria, entretenimiento, marketing y M&O, mostrando ejemplos de sistemas que utilizan AR.

2.1.4.1 Educación

En el caso de la educación ha permitido un nuevo enfoque a problemas donde el alumno requiere cierto nivel de abstracción para poder entender una materia en particular. Esto es gracias a los modelos tridimensionales y la interacción que permite el uso de AR. Un ejemplo en esta área es su uso para la enseñanza de Química Orgánica en estudiantes de secundaria [2]. Este sistema, llamado Augmented Chemistry, permite a los estudiantes aprender sobre relaciones espaciales, adquiriendo rápidamente habilidades de abstracción, ya que pueden manipular modelos 3D de estructuras moleculares de forma sencilla. Las estructuras moleculares son visualizadas mediante AR como si

se encontraran sobre el escritorio, donde el usuario utilizando marcadores especiales puede manipularlas (ver Figura 8). Los usuarios de este sistema encontraron que la forma de interacción que ofrece Augmented Chemistry les permitió utilizarlo con facilidad.

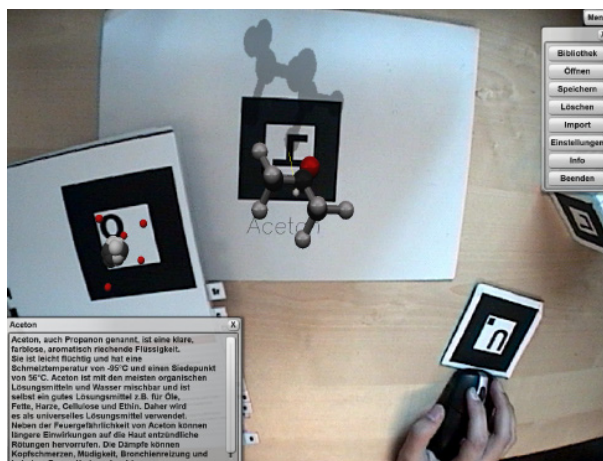


Figura 8: Captura de pantalla de Augmented Chemistry [2].

Enfocado a problemas asociados a las Matemáticas, un grupo de investigadores han desarrollado un sistema llamado Construct3D [25], un sistema AR móvil y colaborativo para la construcción de volúmenes geométricos especialmente diseñado para matemáticos y enfocado a la enseñanza de la geometría. La idea de este software es incrementar las habilidades espaciales de los usuarios. En la Figura 9 se aprecia el sistema desde la perspectiva que tendría una tercera persona, utilizando un HMD, mientras ve como dos de sus compañeros interactúan con volúmenes virtuales.

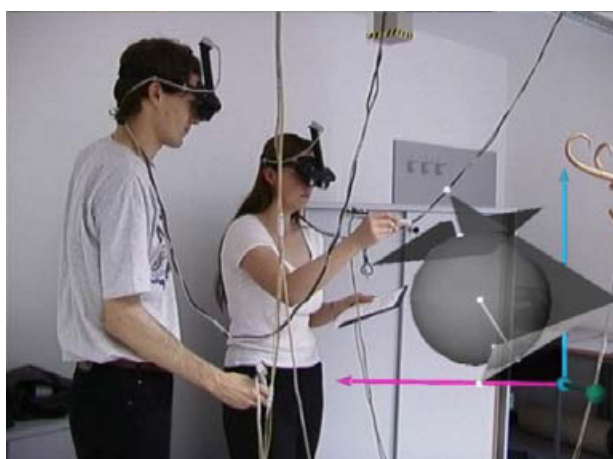


Figura 9: Estudiantes trabajando con Construct3D [25].

Si bien Construct 3D cumple el objetivo planteado, esto es, incrementar las habilidades espaciales de los usuarios, sólo fue testeado en laboratorio, faltando así una prueba real que permita obtener resultados cuantificables.

De los ejemplos anteriores se puede apreciar que el potencial que posee el uso de AR va enfocado a la representación e interacción por parte de los usuarios.

2.1.4.2 Industria

En el área de la industria, el uso de AR busca asistir a operarios de maquinaria y técnicos en sus tareas, agregando información visual a lo que ellos ven mediante el uso de HMD.

La empresa BMW se encuentra realizando investigación sobre AR y su uso para apoyar procesos dentro de la industria automotriz. Su equipo de investigación ha desarrollado la manera de ayudar a los técnicos mostrándoles la información necesaria para montar o desmontar partes de un automóvil [11] evitando así tener que buscarla de forma anexa en manuales técnicos.

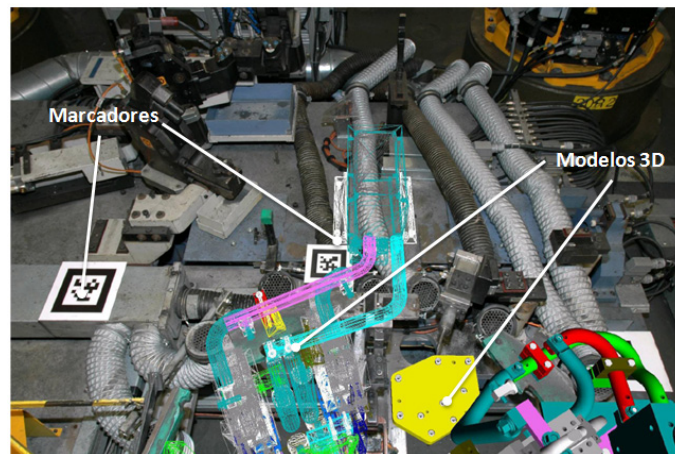


Figura 10: Volkswagen, verificación de partes en un modelo [38].

Volkswagen ha desarrollado prototipos de herramientas que utilizan AR, las cuales cubren todas las partes del proceso de creación de un producto, involucrando diseño, planeamiento, producción, servicio y mantención [38]. En particular, en la Figura 10 se puede ver como esta empresa ocupa AR para comprobar las componentes que restan por instalar en un armazón, mostrándolas traslucidas sobre las piezas ya instaladas, así los operarios se enfocan en las tareas que restan por realizar. Los marcadores son ocupados como puntos de referencia en el armazón, permitiendo que los modelos 3D no necesariamente se tengan que mostrar sobre ellos, sino que de forma desplazada.

2.1.4.3 Entretenimiento

Lo atractivo del uso de AR ha hecho que se incluya también en el desarrollo de videojuegos, cambiando totalmente el paradigma de interacción por parte del usuario.

En el E3 que se realizó en junio de 2009, la empresa London Studio presentó oficialmente un videojuego para PlayStation 3, llamado EyePet, el cual integra el uso de AR. El juego utiliza la cámara PlayStation Eye (la cual no tiene ninguna característica en especial, salvo que es compatible con el PlayStation) para permitir que una mascota virtual pueda interactuar con personas y objetos capturados del entorno real [55]. En la Figura 11 se puede ver al personaje principal del juego, el cual parece ser consciente del entorno, pudiendo reaccionar a éste. El jugador puede colocar objetos delante del personaje y este es capaz de interactuar con éstos. Por ejemplo, si el jugador lanza una pelota hacia él, saltará fuera del camino para evitar ser lastimado, también reaccionará a las acciones del jugador, como al hacerle cosquillas o al aplaudir sus manos para asustarlo.



Figura 11: Captura de pantalla de EyePet.

Otro juego, pero que no salió al comercio siendo solo un prototipo, es ARQuake [56]. Este juego, basado en el juego Quake de idSoftware, permite al jugador visualizar los enemigos de Quake en un entorno real *outdoor* y utilizando AR. El jugador debe usar un HMD que le permite visualizar tanto los objetos reales como los virtuales, para ello el HMD es capaz de saber la orientación y posición de la cabeza para luego calcular la posición de los objetos virtuales.

2.1.4.4 Marketing

Muchas empresas de publicidad ven la AR como una forma de llamar la atención de posibles clientes. La empresa Universal McCann ha experimentando con AR buscando revolucionar el sector del marketing y la comunicación en España [27]. Los nuevos anuncios publicitarios basados en AR presentan múltiples posibilidades de aplicación en los medios más diversos. Asimismo, presentaciones, eventos, *packaging*¹ e información al consumidor en punto de venta son otras posibilidades en las que la utilización de AR puede enriquecer exponencialmente el mensaje publicitario [57].



Figura 12: Portada revista Esquire El hombre que aparece ahí se encuentra sentado en uno de los marcadores que pueden ser utilizados en el sitio web de la revista.

La revista Esquire, cuya portada se muestra en la Figura 12, presentó una edición especial en donde se integraron marcadores especiales en toda la revista. Cuando se acercaban estos marcadores a una webcam, y se estaba dentro del sitio web de la revista, una aplicación desarrollada por ellos identificaba los marcadores y sobreponía contenido multimedia en la revista. [42]. La misma idea la

¹ Es la ciencia, el arte y la tecnología de encerrar o la protección de productos para distribución, almacenamiento, venta y uso. También se refiere al proceso del diseño, evaluación y producción de paquetes.

implementó la revista chilena Paula [16], utilizando el mismo principio de adjuntar un marcador en su portada.

2.1.4.5 Movilidad y Orientación

La AR es utilizada también para ayudar en tareas de M&O, entregando información extra al usuario sobre su entorno. Un ejemplo de esto lo protagonizan los investigadores de Carnegie Mellon [7], quienes desarrollaron una tecnología capaz de sincronizar dos cámaras, una fija que transmite un *streaming*² y otra dentro de un automóvil que comunica a un computador a bordo del ángulo de visión relacionado con la cámara fija. Un software especial relaciona la visión de ambas cámaras y añade a la primera lo que la cámara fija puede ver y la de a bordo no. El ejemplo de lo que observaría el usuario a bordo del automóvil se puede ver en la Figura 13.

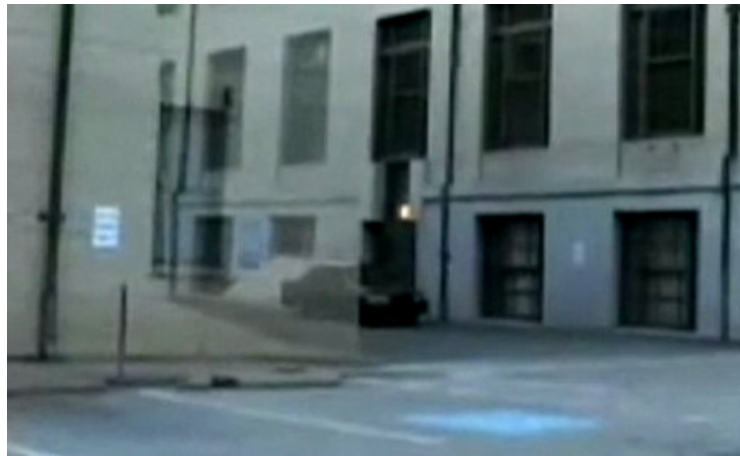


Figura 13: Utilización de AR para disminuir accidentes de tránsito.

Como se puede ver, las aplicaciones de AR son diversas, y el rol de esta tecnología es la de agregar información en tiempo real al contexto real, además de cambiar la forma de cómo el usuario interactúa con una aplicación. Este trabajo de memoria enfoca su uso para ayudar a personas ciegas en tareas de M&O.

Otro ejemplo es Layar [6], su navegador AR que utiliza como *Handheld displays* teléfonos con sistema Android o iPhone OS. Layar, utiliza el receptor GPS y la brújula de los teléfonos para

² Consiste en la distribución de audio o video por Internet. Este tipo de tecnología permite que se almacenen en un búfer lo que se va escuchando o viendo.

determinar su orientación o hacia dónde están mirando y GPS para determinar su posición. El sensor de movimiento se utiliza para conocer el ángulo de inclinación del teléfono. La cámara del teléfono recoge el entorno y reproduce la imagen en la pantalla del teléfono, mientras que el software superpone sobre la imagen información relativa a lo que aparece en pantalla como aparece en la Figura 14.



Figura 14: Captura de pantalla de Layar.

Si, por ejemplo, la cámara del teléfono se dirige hacia un edificio histórico, Layar puede superponer sobre la imagen de éste, en pantalla, información histórica o arquitectónica o relacionada con la construcción visualizada. Layar dispone de diversas capas de datos con diferentes contenidos a elección del usuario. Pudiendo señalar la posición de lugares tales como cafeterías, restaurantes, gasolineras, centros comerciales o, cines.

Como se puede ver, las aplicaciones de AR son diversas, y el rol de esta tecnología es la de agregar información en tiempo real al contexto real, además de cambiar la forma de cómo el usuario interactúa con una aplicación. Este trabajo de memoria enfoca su uso para ayudar a personas ciegas en tareas de M&O.

2.2 Tecnologías de apoyo para reforzar habilidades de M&O

La pérdida de la visión tiene muchas causas, como por ejemplo una enfermedad a los ojos, daño causado por accidentes cerebro - vasculares, diabetes, parto prematuro o traumatismo en el ojo. Según una estimación hecha el 2003 por la Organización Mundial de la Salud, se cree que existen

mundialmente entre 40 y 45 millones de personas ciegas y que 135 millones poseen baja visión [60].

En el contexto nacional, quienes declaraban tener deficiencia visual en el 2004 alcanzaban las 634.906 personas [17]. Esto equivale al 3,9% de la población total del país (1 de cada 25 personas presenta deficiencias visuales). El año 2006, la encuesta Casen mostró que quienes tienen discapacidad visual alcanzaban las 510.370 personas, donde 59,4% eran mujeres y 40,6% eran hombres (la encuesta Casen no abarca el total de la población).

En nuestro país, según el decreto N°2505, de 1995, del Ministerio de Salud, se considera ciego legal a las personas que sufren alguna discapacidad visual que disminuye: “...en a lo menos un tercio la capacidad del sujeto para desarrollar actividades propias de una persona no discapacitada, en situación análoga de edad, sexo, formación, capacitación, condición social, familiar y localidad geográfica. Las deficiencias visuales y auditivas se ponderarán, considerando los remanentes del mejor ojo u oído corregido el defecto”.

De la anterior definición se obtiene que la movilidad de una persona ciega se verá afectada, disminuyendo así su autonomía y viéndose en desventaja en relación con las personas con visión normal.

Es interesante observar que para interactuar con el mundo real, además de conocer y representar mentalmente el espacio a recorrer o ya recorrido, es importante la interacción que se da con los objetos que se encuentran distribuidos en el ambiente. Es por eso, que los usuarios que presentan una discapacidad visual están habilitados a describir sus rutas con mayor detalle comparado con la descripción que daría una persona con visión, pues se encuentran mucho más pendientes de los objetos de referencia que les facilitarán el desplazamiento, como por ejemplo, si hay agujeros en la ruta, sonidos de fábrica, colegio u olores característicos.

Sin embargo, uno de los mayores problemas que tienen estas personas ciegas al desplazarse es que se encuentran restringidos a un mismo espacio siempre, ya que cambiar de ruta o desviarse del camino puede ser peligroso para su integridad. Es posible además, que el cambio les cause confusión, impidiendo que éstos puedan retomar la ruta que llevaban. En síntesis, los usuarios ciegos están restringidos a seguir rutinas de movimiento y negados a explorar nuevos lugares [23].

El problema de desplazamiento puede darse en dos contextos [40], uno donde el usuario no conoce el ambiente y otro donde sí lo conoce. Cuando se recorre el ambiente por primera vez, la persona intentará encontrar marcas o zonas reconocibles que le ayudan a orientarse y a armar un esquema mental del ambiente que está recorriendo. Una vez realizado lo anterior, el ambiente pasará a ser conocido, pudiendo ya recorrerlo nuevamente pero realizando decisiones inteligentes en base al esquema mental armado.

Tecnológicamente hablando es posible ayudar a una persona ciega para reforzar sus habilidades de M&O en base a dos enfoques. El primero se basa en el uso de aplicaciones computacionales que representan virtualmente el entorno real que se busca recorrer, con el cual el usuario ciego puede interactuar y simular la navegación. Este enfoque se categorizará como aplicaciones de Entrenamiento. El segundo enfoque se basa en entregar información *in situ* al usuario ciego a través de una aplicación computacional que corra en algún dispositivo portable cuando este se encuentra recorriendo el entorno real. Este enfoque se categorizará como aplicaciones de Asistencia.

Es claro que sea cual sea el enfoque, la forma en que se entregará la información o *feedback*³ al usuario ciego debe ser prescindiendo del sentido de la visión. Por ejemplo, es factible construir una interfaz basada en audio que permita representar el entorno virtual, o utilizando, ya sea de manera conjunta o por separado, interfaces hápticas⁴, gracias a dispositivos como el SensAble PHANTOM [39] o el Novint Falcon [36]. Así, una persona podrá ser capaz de localizar objetos u otras personas en un entorno virtual tal como lo haría en un entorno real [62].

En relación al sonido, su uso espacializado ha permitido que personas ciegas sean capaces de localizar puntos específicos en un entorno 3D de manera precisa, favoreciendo la creación de representaciones mentales [47].

³ En HCI (*Human Computer Interaction*) es un principio de diseño que está relacionado con el estímulo que recibe un usuario luego de realizar una acción.

⁴ Significa todo aquello referido al contacto, especialmente cuando éste se usa de manera activa. La palabra no está incluida en el diccionario de la Real Academia Española y proviene del griego háptō, que quiere decir tocar o que es relativo al tacto.

El desafío de generar estas interfaces es velar porque el usuario no se sienta saturado por una excesiva cantidad de información al nivel que en lugar de ayudarlo a tomar decisiones, lo complique.

2.2.1 Aplicaciones de entrenamiento

Un trabajo bastante reconocido e innovador en su momento fue AudioDoom [51]. Este videojuego, utiliza sonido estéreo para representar ambientes virtuales para propósitos de aprendizaje, cognición y entretenimiento. En AudioDoom, el jugador ciego debe recorrer los pasillos de un laberinto ficticio en busca de la salida. Para hacer más desafiante esta tarea, en el laberinto se esconden alienígenas que deben ser eliminados con municiones que se encuentran dispersas en los pasillos. Lo que se quiso validar en la investigación es la hipótesis que el uso reiterativo del videojuego, y gracias a la interfaz basada en audio, los jugadores desarrollan una representación mental del laberinto. Para ello se diseñó una actividad en que los usuarios, utilizando bloques Lego®, pudieron recrear el laberinto con exactitud aceptable. En la Figura 15 se puede ver el proceso que se llevó a cabo para validar la hipótesis de la investigación: en (A) se muestra el laberinto que debían recorrer los usuarios; en (B) aparece uno de los usuarios con un set de bloques Lego® iniciando el proceso de construcción del laberinto; y en (C) se ve el resultado final del experimento.

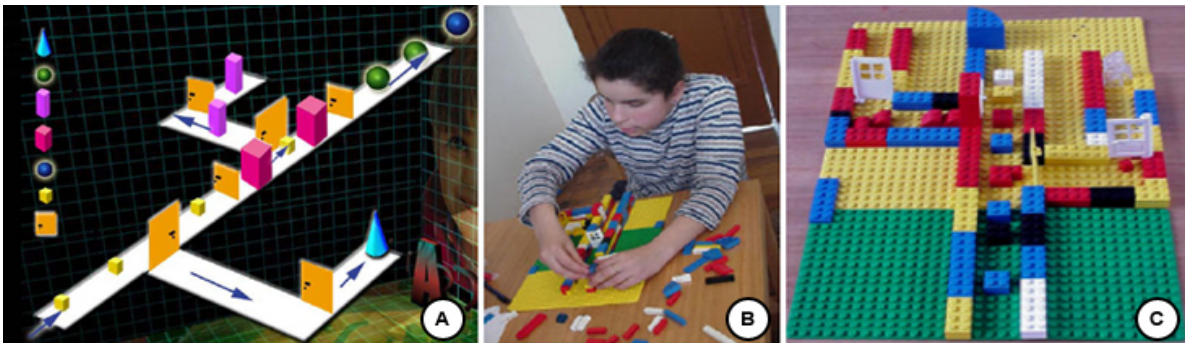


Figura 15: Validación de hipótesis de AudioDoom [51].

Otro videojuego del mismo estilo es AbES [53]. Este videojuego, cuya sigla significa en inglés “*Audio-based Environment Simulator*”, extiende el concepto de los pasillos ficticios de AudioDoom para generar una representación virtual basada en audio de entornos reales, sirviendo así como una aplicación de entrenamiento útil para los usuarios ciegos.

Para lograr crear el ambiente virtual se toma un plano de planta de la construcción que se busca simular, trazándose una cuadrícula formadas por celdas que poseen el tamaño de un paso promedio de una persona adulta. Dependiendo de la posición de la celda y su correspondencia con el plano, ésta toma las propiedades que representan dicho elemento en el mapa. Por ejemplo, una celda puede representar parte de un muro, una puerta, un espacio vacío, un escalón o muebles. Dado lo anterior la celda generará cierto tipo de *feedback* icónico sonoro estéreo en la simulación, el cual varía el canal de salida según la relación entre la orientación del usuario y la celda. Conformado el ambiente virtual, el usuario al recorrerlo puede preguntar en cualquier momento en qué habitación se encuentra y hacia dónde observa, obteniendo *feedback* gracias al uso de *Text to Speech* (TTS). Sus desplazamientos son, a modo de simplificar la tarea cognitiva del mapping mental, verticales, horizontales y siempre avanzando.

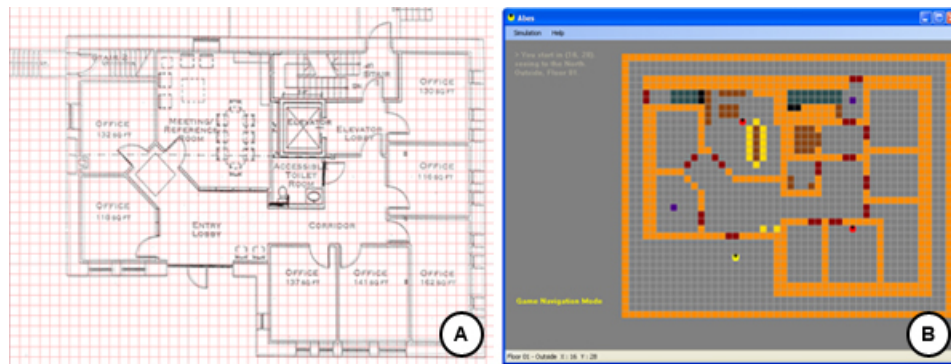


Figura 16: Conversión de un entorno real a una representación virtual en AbES. (A) Plano de planta de un edificio (B) Representación virtual del plano de planta desplegada en la interfaz del facilitador.

En lo referente a la forma de cómo se le entrega la información al usuario, el uso de háptica también es una opción factible. Mioduser y Lahav [32] han generado una aplicación que cuenta con 2 modos. El primer modo permite que el facilitador pueda diseñar un plano de planta en base al uso de figuras geométricas. Una vez formada esta representación del entorno real, la aplicación es capaz de generar una simulación virtual basada en háptica del mapa. En el segundo modo, el usuario ciego puede recorrer el mapa e ir utilizando un dispositivo háptico para detectar volúmenes que representan obstáculos en el entorno real. Lo que percibe el usuario sería equivalente a tener una maqueta y trazar posibles rutas utilizando un lápiz. La sensación que se percibe al colisionar con un objeto corresponde a la sensación háptica que entrega el dispositivo.

2.2.2 Aplicaciones de asistencia

Existen variadas soluciones que son diseñadas como asistentes tecnológicos para que personas ciegas puedan obtener información de su entorno en tiempo real y así tomar mejores decisiones para su desplazamiento.

La característica principal de estos asistentes es que de alguna u otra forma deben recolectar datos del entorno en tiempo real para procesarlos y entregar información al usuario.

Una manera de obtener datos del entorno es utilizando dispositivos de captura de video y aplicando técnicas relacionadas al área de *Computer Vision*. Estas permiten transformar los datos obtenidos, correspondientes a cuadros de video, en decisiones o nuevas formas de representación [24].

Bourbakis y Kavraki [13] crearon una aplicación que utiliza una pieza de hardware especial consistente en una matriz de puntos y un par de cámaras. Las cámaras, ubicadas al frente del individuo permiten capturar una imagen 3D del entorno frente a él. Luego la aplicación transforma la imagen 3D en una 2D que es representada en el hardware especial. Este hardware consiste en una matriz de puntos que pueden moverse de abajo hacia arriba, generando relieve. En este hardware, que se coloca con los puntos hacia el pecho del usuario, se van dibujando los contornos y relieves que representan la escena generando *feedback* háptico de los obstáculos que logran capturar las cámaras.

El hecho de encontrar obstáculos en el entorno no es necesariamente la información más relevante que el usuario puede desear. See CoLoR [12] es un sistema que, usando dos cámaras para obtener una imagen 3D, es capaz de distinguir colores de los objetos que lo rodean. Para ello, los colores de la imagen capturada son convertidos a un modelo HSL⁵ (*Hue, Saturation, Lightness*). Estas 3 componentes del color son convertidas por la aplicación a sonidos especializados de instrumentos musicales clásicos. La tonalidad (*Hue*) es convertida a un timbre en particular de un instrumento musical, la saturación (*Saturation*) es convertida a 4 posibles notas musicales y la luminosidad (*Lightness*) es representada con bajos cuando es oscura y una voz cantando cuando es clara. La distancia a los objetos fue codificada con la duración del sonido, teniendo 6 posibles niveles.

⁵ El modelo HSL define el color en términos de sus componentes constituyentes, los cuales son la Tonalidad, Saturación y Luminancia.

En el estudio asociado al proyecto See CoLoR, los usuarios fueron entrenados para asociar los sonidos espacializados con colores y sometidos a pruebas, en donde los usuarios debían observar utilizando el sistema un grupo de globos y distinguir el color de cada uno, como aparece en la Figura 17. Después de varias sesiones de entrenamiento, los usuarios de See CoLoR lograron identificar colores en objetos sin problemas. Además, el reconocimiento de colores ha permitido el reconocimiento de escenas perdiendo ambigüedad, sin embargo el tiempo que requiere un usuario para lograr eso es bastante largo.



Figura 17: Usuario utilizando See CoLoR para detectar colores a través del sonido [12].

Algunas aplicaciones del tipo asistente requieren saber continuamente la posición del usuario, como aquellas que utilizan GPS o marcadores RFID. En la investigación, el equipo de Zhang [63] se propone un sistema para ayudar a personas ciegas con motivación en que la información sea provista sólo en algunos puntos de un entorno real para realizar decisiones. Para ello, se han pegado carteles monocromáticos o marcadores en ciertos puntos, los cuales pueden ser capturados mediante una cámara web e interpretados por el sistema para luego representar la información mediante una interfaz visual, háptica o basada en audio (cuál es el objeto, en qué lugar están, etc.). El equipo de Zhang plantea obtener los datos a través del uso de la librería ARToolKit [5], la cual permite el desarrollo de aplicaciones de Realidad Aumentada, pero no propone una interfaz que le brinde información al usuario ciego.

2.3 Metodologías de usabilidad

Según Jacob Nielsen, quien ha sido reconocido como uno de los principales expertos del mundo en diseño amigable con el usuario[58], dice que la usabilidad es un atributo de calidad que evalúa cuán

fácil de utilizar son las interfaces de usuario, además se refiere a este concepto como a los métodos que permiten mejorar la facilidad de uso durante el proceso de diseño [35]. Sin embargo, según la norma ISO 9241 la definición de Usabilidad corresponde a la medida en que un producto puede ser usado por usuarios específicos para lograr los objetivos especificados con efectividad, eficiencia y satisfacción en un contexto de uso específico [22].

Sea cual sea la definición que se le dé al concepto de Usabilidad, se tiene claro que hay un consenso con relación al uso y satisfacción de un producto o sistema.

La usabilidad posee 5 atributos [34]:

- Aprendizaje: El sistema debería ser fácil de aprender, tanto que el usuario puede rápidamente comenzar a trabajar usando el sistema.
- Eficiencia: El sistema debería ser eficiente en el uso, tanto que si el usuario ha aprendido a usar el sistema, existe un alto grado de productividad.
- Memoria: El sistema debería ser fácil de recordar, de tal forma que si el usuario deja de interactuar con el sistema por un período de tiempo, puede volver a utilizarlo sin aprender nada.
- Errores: El sistema debería tener una tasa baja de errores de tal forma que al interactuar con el sistema, el usuario tenga pocos errores, y cuando los tenga, pueda recuperarse rápidamente.
- Satisfacción: El sistema debería ser placentero en su uso. El usuario debería quedar satisfecho al utilizarlo.

La usabilidad busca medir qué tan usables son las interfaces o los productos. Para llevar a cabo esto, se utilizan una serie de métodos que permiten rescatar información del uso y de los intereses de los usuarios finales, con tal de mejorar la eficiencia y la satisfacción en la interacción. Estos métodos pueden ser tanto cuantitativos como cualitativos, y deben ser considerados desde el diseño del producto ya que si no es así, probablemente aumentarán los costos y será difícil la implementación [19].

Capítulo 3: Diseño de ARTAB

3.1 Consideraciones preliminares

Es sabido que las personas con discapacidad visual tienen una percepción distinta del mundo que las rodea. Por lo anterior, es importante que todo desarrollo que apunte a considerarlos como usuarios finales, los involucre durante el proceso de diseño e implementación. En definitiva se debe adoptar un diseño centrado en el usuario [21], en que las limitaciones de estos sean consideradas en todo el proceso de diseño, logrando un producto que por un lado, será utilizado y por otro, será funcional a sus necesidades.

El audio verbalizado es preferido a otro tipo de audio por muchos de los usuarios ciegos [46]. Sin embargo, depende del contexto el tipo de información auditiva que es más apropiado proporcionar. Hay casos en que demasiada información distrae, y puede llevar a la confusión.

Para la representación de acciones y el *feedback* auditivo por medio de audio, existen diversas formas que han sido evaluadas con éxito, tales como sonidos semejantes para acciones similares, sonidos inversos para acciones inversas, jugar con la frecuencia de los sonidos, y utilizar notas musicales [48][49].

3.2 Descripción general

ARTAB (*Augmented Reality Tags for Assist the Blind*) es un sistema que utiliza Realidad Aumentada para agregar información al entorno real mediante un dispositivo para captura de video, identificando ciertos objetos de interés y generando una representación virtual basada en audio que permita saber su posición dentro del rango de visión que permita dicho dispositivo.

El sistema, el cual requiere de un computador portátil, funciona de la siguiente manera: previamente en el entorno real donde se utilizará ARTAB se han identificado los objetos del entorno que son de importancia y se les ha asignado un marcador, el cual es un cuadrado con un diseño monocromático único. Una cámara web es situada justo a la altura de los ojos del usuario, a modo de utilizar movimientos del cuello para direccionarla hacia un punto de interés entre los distintos marcadores. El usuario interactúa con la aplicación gracias a una pequeña botonera. Los objetos de importancia son todos aquellos objetos con los cuales se desea que el usuario ciego interactúe. Éstos, por ejemplo, podrían ser puertas, inmobiliario y obstáculos varios.

La identificación de los marcadores se producirá cuando estos estén dentro del campo visual que permite el dispositivo para la captura de video. Si en la imagen obtenida se identifica algún objeto o conjunto de objetos, se emite un sonido característico constante que alerta al usuario de su presencia dentro del campo visual. Si éste se encuentra en un sector céntrico de la imagen, el usuario podrá preguntar información sobre el objeto presionando un botón en la botonera.

Se busca que el uso de este sistema, y sobre todo la interacción con esta interfaz basada en audio, permitan a los usuarios identificar objetos en un contexto dado con mayor facilidad, tal como si se tratase de un individuo con visión normal.

3.3 Modelo

Para el diseño de ARTAB se utilizó un modelo de desarrollo de aplicaciones móviles, el cual cuenta con los siguientes componentes [50][52] :

- Modelamiento: Esta componente corresponde principalmente a cómo se realizará la representación del ambiente que el usuario recorrerá. Para ello, se han marcado partes del ambiente utilizando marcadores especiales en los puntos y objetos de importancia. Estos marcadores serán capturados a través de un dispositivo de captura de video e interpretados por ARTAB.

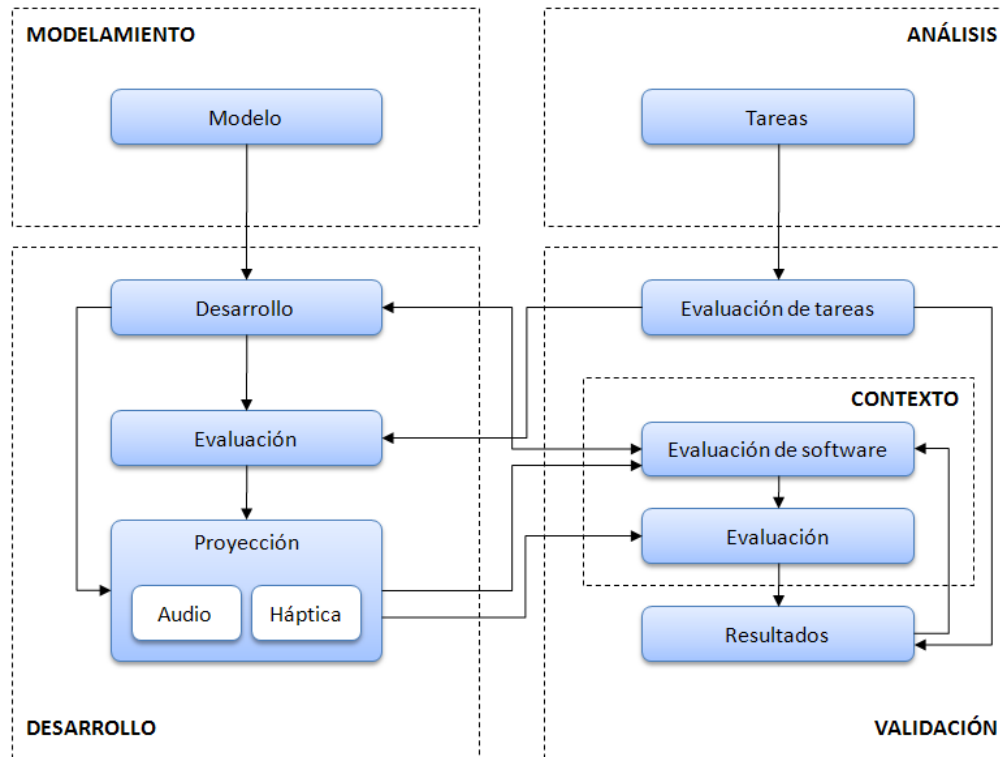


Figura 18: Modelo utilizado para el diseño de ARTAB [50][52]

- Desarrollo: El proceso de desarrollo del sistema consta de etapas específicas que permiten crear una aplicación que cumpla con los objetivos planteados y que sobre todo sea accesible para personas con discapacidad visual. Para el caso de ARTAB, la accesibilidad viene dada por el hecho que éste posee una interfaz basada en audio y háptica (por la botonera utilizada para realizar preguntas al sistema). ARTAB entrega la información (*output*) por medio de TTS pre grabados y audio espacializado. La interacción del usuario con el dispositivo (*input*) se realiza por medio de la posición relativa de la cámara en relación a los marcadores y una botonera. La proyección determina como la información debe ser representada por medio del audio y el modo de interacción. Esta información es necesaria para que el sistema sea funcional para el usuario. En ARTAB el sonido espacial escuchado por el usuario corresponde a la posición relativa del marcador con relación al dispositivo de captura. La evaluación corresponde a determinar si el sistema es correcto en los datos que evalúa y la información que entrega a los usuarios.

- **Análisis:** Con el objetivo de analizar el uso del sistema desarrollado se deben diseñar tareas específicas para estudiar el correcto impacto del sistema. En ARTAB, para evaluar la usabilidad se diseñaron tareas de desplazamiento por lugares desconocidos para usuarios ciegos. Estos debían encontrar ciertos objetos, representados por los marcadores.
- **Validación:** Esta etapa permite realizar correcciones al sistema, redefiniendo las interfaces para mejorar la interacción del usuario. Con el objetivo de revisar el desempeño de los usuarios en las tareas definidas, se realizaron pruebas que permitieron corregir errores de diseño de la plataforma. Este proceso fue fundamental para mejorar la pertinencia del sistema con relación a cómo los usuarios interactúan con el dispositivo móvil. Durante esta etapa se realizaron evaluaciones de usabilidad en un contexto real consistentes en utilizar el sistema para evaluar el modo de interacción y las interfaces de entrada y salida.

3.4 Decisiones de diseño

Para el diseño de ARTAB se tomó la idea del equipo de Zhang, integrándola a una interfaz basada en audio considerando conceptos del proyecto See ColOr. Este sistema podría tomar datos como la posición espacial de los marcadores y asociando cada eje a un efecto sonoro para dar una sensación de espacialidad al usuario y brindarle información. Esta idea es similar a la que comparten otros investigadores, quienes proponen que el desarrollo de un asistente enfocado al uso de audio espacializado, que permita la localización de objetos mediante la captura de éstos con el uso de cámaras, sería útil para los usuarios con discapacidad visual [15].

Inicialmente se generó el diagrama de flujo mostrado en la Figura 19, el cual muestra a grandes rasgos como será el comportamiento del software que utilizará el sistema. Un aspecto observable de este flujo es que se asemeja mucho a lo que se conoce como ciclo de juego, en el cual se espera por el input del usuario, se actualizan los estados del programa y luego se despliega la información.

Lo anterior cambia la manera de cómo el software debe ser implementado. Volviendo al concepto de ciclo de juego, una diferencia clave entre el desarrollo de un videojuego (el cual incluye el ciclo) y las aplicaciones típicas de escritorio es el concepto de “*polling*” versus el registro de eventos [41].

Muchas aplicaciones, las cuales no son videojuegos, están escritas exclusivamente para escuchar eventos generados por los usuarios de dichas aplicaciones. Como ejemplo, un programa que posee una pantalla que solicita al usuario el nombre y que tiene los botones “Aceptar” y “Cancelar”,

independiente del lenguaje de programación que se haya escrito, por lo general no hará nada hasta que el usuario presiona algún botón. Cuando el usuario pulsa cualquiera de los botones, el sistema lanza un evento que el programa captura. Por el contrario, en un juego se utiliza “polling”, donde los eventos se van encolando hasta que el ciclo llegue al momento en donde éstos son revisados.

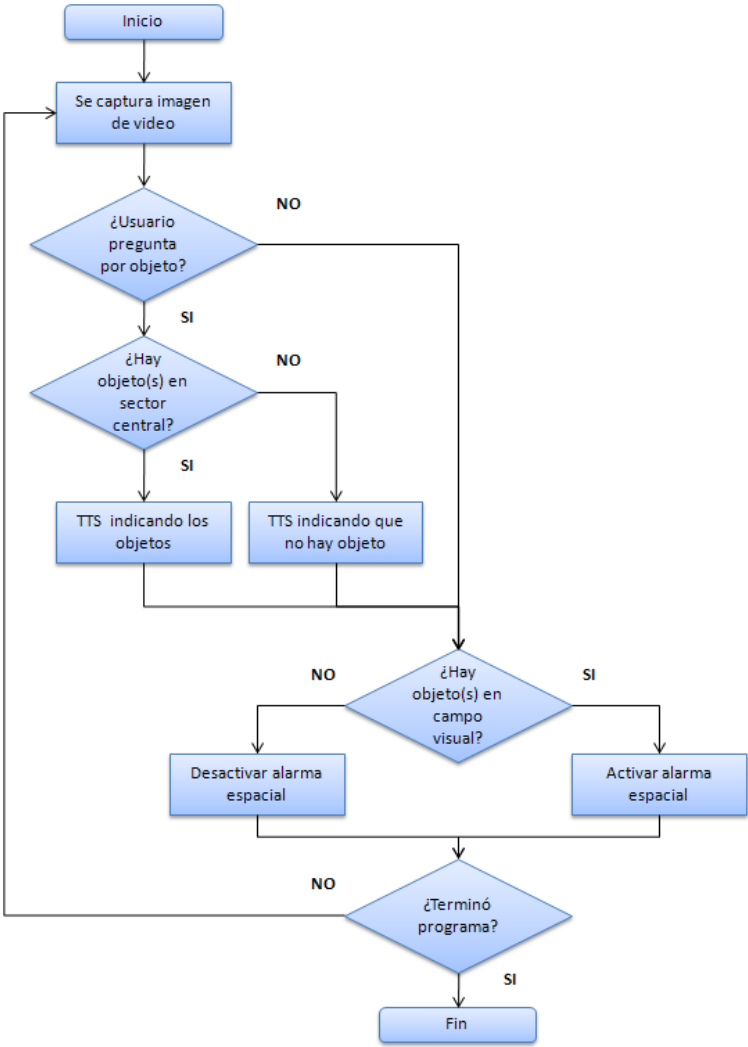


Figura 19: Diagrama de flujo simplificado de ARTAB.

Para clarificar lo anterior se puede tomar como ejemplo el movimiento del ratón por parte del usuario. El juego será responsable de preguntar si el ratón se ha movido, mientras tanto la aplicación se encuentra siempre realizando acciones [41].

Tomaremos como base el flujo que tiene un ciclo de juego, el cual se muestra en la Figura 20, acomodando las distintas tareas del flujo anterior de la siguiente manera:

- **Inicialización:** En esta sección se realiza la inicialización y asignación de variables, instancias o estructuras necesarias para que el programa funcione. En este flujo en particular se ha fusionado lo que tiene relación con la carga de contenidos y la inicialización, debido a que en ARTAB esto se realiza una sola vez. Aquí se hará la inicialización de la cámara para comenzar la captura y la carga de contenidos como la información que se requiere para el reconocimiento de los marcadores y sonidos necesarios para la interfaz basada en audio.
- **Procesamiento Input:** El usuario interactúa con el sistema a través de una botonera. Cuando este presiona alguno de sus botones se va ingresando información a una estructura que indica qué botón se presionó. Esta estructura es analizada en este punto del flujo, por lo cual toda acción de input quedará encolada y todas impactarán en el proceso de la aplicación una vez pasado este punto.
- **Update:** En este punto se encapsulan la mayoría de las tareas presentadas en el diagrama de flujo de la Figura 19. Inicialmente se toma el cuadro actual de la captura de video que realiza el programa y se extrae información de ésta, correspondiente a si existen marcadores o no y cuál es su posición en relación al campo visual del dispositivo. Si existió input del proceso anterior, se realiza la acción de corroborar si hay o no un objeto en el sector central y en base a eso cargar el TTS adecuado. Además, si se reconocieron marcadores o no se procede a la modificación del estado de las alarmas, pasando de activas a inactivas o al revés según corresponda.
- **Render:** La etapa de *Render* es aquella asociada a presentar los elementos del juego en pantalla. En particular, aquí se encuentran todas las acciones que tienen relación con la interfaz de salida, la cual será utilizada por el usuario del sistema, correspondiente a una interfaz basada en audio, y por el facilitador, la cual será una interfaz gráfica. La salida gráfica será utilizada para tareas de detección de errores del programa y para análisis del comportamiento del usuario del sistema, ya que gracias a ésta podremos saber qué es lo que se encuentra enfocando y los elementos que están visibles en dicho momento.

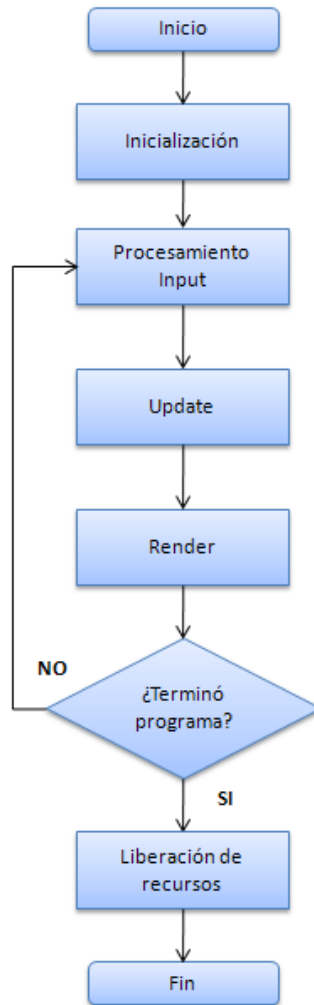


Figura 20: Ciclo de juego.

- Liberación de recursos: Si la ejecución del programa termina, esto quiere decir que el facilitador ha enviado la orden para que el programa detenga su ejecución, todo recurso cargado en memoria será liberado en esta etapa. Esta etapa debería considerarse en cualquier flujo de programa ya que es considerado una buena práctica.

El paso final del diseño fue la estructuración del sistema, de esta forma se definió que ARTAB requiere de dos componentes principales para funcionar:

- Sistema de Tracking: Esto corresponde al sistema que es capaz mediante el procesamiento de un cuadro de video, obtener información que permite identificar ciertos objetos de

interés y hacerles seguimiento, detectando su posición, orientación y escala relativa a la cámara en todo momento. Esto es básico para realizar aplicaciones de Realidad Aumentada.

- Sistema de Sonido: Ya que los usuarios obtendrán la información utilizando una interfaz basada en audio, se hace necesario tener un sistema que facilite su uso e integración con la aplicación a desarrollar. En particular que permita el uso de sonido 3D por software y la aplicación de efectos tales como ajustar dinámicamente el pitch, volumen o reverberación, lo cual permitirá generar distintas representaciones de audio espacializado.

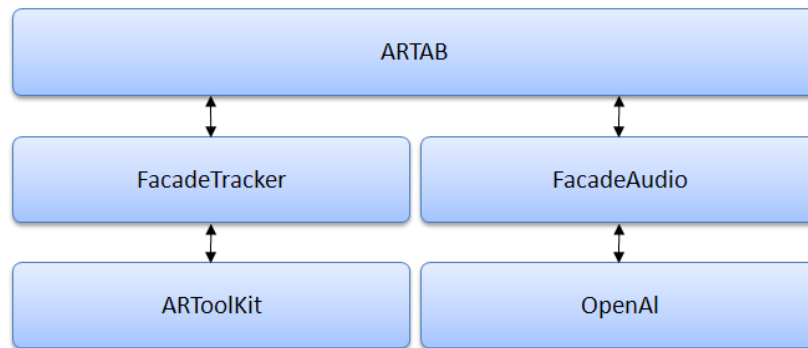


Figura 21: Estructuración de los componentes de ARTAB.

Para llevar a cabo lo anterior existen variadas librerías, como por ejemplo ARToolKit [5], y OpenCV [37] para implementar el tracking y DirectX [30] y OpenAL [20] para implementar sonido. Dado esto, es posible que en el futuro se quisiera cambiar los objetos de interés y por ejemplo, capturar ya no sólo marcadores sino otros objetos distintos, lo mismo puede ocurrir con el sonido, donde puede surgir una librería que consuma menos recursos.

Además, algunas librerías suelen entregar mayor cantidad de funcionalidades en relación a las que son verdaderamente requeridas por la aplicación en desarrollo. Es por esto que la estructuración de los componentes que integran el sistema ARTAB será como aparece en la Figura 21.

La aplicación se comunicará con dos componentes llamados FacadeTracker y FacadeAudio, los cuales encapsulan las librerías ARToolKit y OpenAL respectivamente, haciendo de capa de abstracción. Lo anterior sigue el patrón denominado Facade, el cual busca proveer una interfaz unificada a un conjunto de interfaces en un subsistema, definiendo una interfaz de alto nivel fácil de utilizar [18].

Capítulo 4: Desarrollo de ARTAB

4.1 Herramientas de desarrollo

4.1.1 Hardware

Como dispositivo de captura de video se utilizó una cámara web Creative con una resolución de 640 x 480 pixeles y una tasa de captura de 30fps. Para ejecutar los prototipos se utilizaron 20 Megas de disco duro en un notebook Intel Core 2 Duo de 2.00 GHz y 2 Gigas en RAM. Para la salida de audio se utilizaron audífonos sin características especiales.

La cámara fue adaptada para que ésta pudiese ser utilizada por el usuario a la altura de los ojos, para ello, se utilizó un sombrero como aparece en la Figura 22, donde esta fue fijada en la visera. Si esto se quisiera comparar con los dispositivos utilizados para realidad aumentada, se diría que este es un HMD basado en video, donde en lugar de monitores para *feedback* visual se tienen audífonos estéreo para *feedback* auditivo.

Estos dispositivos son conectados a un notebook donde los datos son procesados para poder entregar luego la información al usuario. Este notebook es además utilizado por un facilitador para

la realización de evaluaciones asociadas a conductas del software y el usuario durante el desarrollo de una tarea en particular.

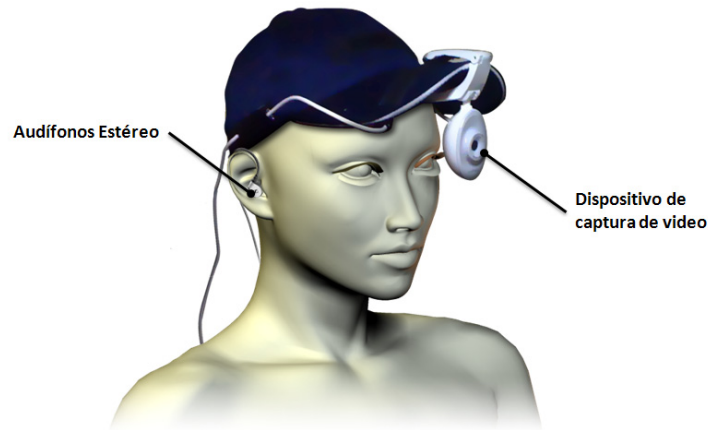


Figura 22: Hardware utilizado por el usuario de ARTAB.

4.1.2 Software

ARTAB fue desarrollado utilizando el C++, el cual es un lenguaje de programación basado en C que permite la manipulación de objetos.

Durante el proceso de desarrollo se utilizaron las siguientes herramientas:

- Microsoft Visual Studio 2008: Es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET,
- Loquendo TTS: Son un conjunto de voces sintetizadas que pueden ser utilizadas para realizar TTS.
- NextUp TextAloud 2.255: Programa que permite utilizar voces TTS y cambiar sus parámetros como tono y velocidad para luego grabar la salida en archivos de audio.
- Syntrillium Software Cool Edit Pro 2.0: Programa que permite la edición de archivos de audio, pudiendo exportar los resultados a diversos formatos.

Además, se utilizaron 3 librerías que permitieron manejar el ciclo de juego, trabajar con audio espacializado y soportar las tareas asociadas a AR:

- GLUT: Es una librería multiplataforma de utilidades para programas basados en OpenGL que proporciona diversas funciones de entrada/salida con el sistema operativo. Entre las funciones que ofrece se incluyen declaración y manejo de ventanas y la interacción por medio de teclado y ratón.
- OpenAL: es una librería de audio multiplataforma desarrollada por Creative Labs para el renderizado eficiente de audio posicional y multicanal en tres dimensiones. Está ideada para su uso en videojuegos. Su estilo y convenciones de código es similar al de OpenGL.
- ARToolKit: Es una librería que permite la creación de aplicaciones AR. Para ello, utiliza las capacidades del tracking visual con el fin de calcular, en tiempo real, la posición de la cámara y la orientación relativa de marcadores especiales físicos, de los cuales obtiene información para disponer los objetos virtuales.

Debido a la importancia de esta última librería, en la siguiente sección se profundizará sobre su funcionamiento.

4.1.3 ARToolKit

ARToolKit [5] fue desarrollada originalmente por Hirokazu Kato en 1999 [5] y fue publicada por el HIT Lab de la Universidad de Washington. Actualmente se mantiene como un proyecto de código abierto alojado en SourceForge con una versión comercial administrada por ARToolWorks.

En una aplicación AR se hace necesario calcular el punto de vista de la cámara para así poder realizar las transformaciones necesarias sobre los objetos virtuales [10], es decir, se debe alterar la posición, tamaño y orientación para que esos objetos sean percibidos por el usuario en el mundo real como si realmente estuvieran allí.

Para ello se utilizan unos marcadores de forma cuadrada, que se componen de un cuadrado negro con un cuadrado blanco cuatro veces más pequeño en su centro, y un dibujo sencillo en el interior de éste. La aplicación, utilizando las funciones y utilidades proporcionadas por ARToolKit, será capaz de detectar cada uno de los marcadores en las imágenes de vídeo capturadas. Un ejemplo de los marcadores utilizados se muestra en la Figura 23.

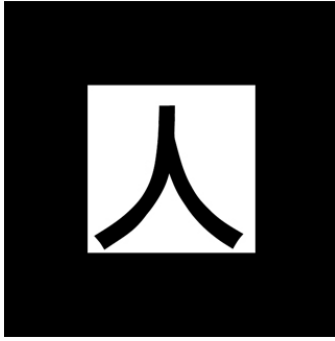


Figura 23: Ejemplo de marcador utilizado por ARToolkit.

Cuando el marcador es detectado, se analiza su orientación, posición y tamaño, con esto es posible calcular la posición de la cámara respecto al marcador y con ello obtener una matriz de transformación que permitirá adecuar un modelo 3D para que parezca que está sobre el marcador.

El funcionamiento básico de una aplicación que utiliza la librería ARToolkit es descrito a continuación:

- Se captura una imagen del mundo real mediante un dispositivo de video, como por ejemplo una webcam.
- La imagen obtenida es umbralizada dado cierto valor umbral, de forma tal que los píxeles cuya intensidad supere el valor del umbral son transformados en píxeles de color negro. El resto es transformado a píxeles blancos (ver Figura 24).
- Se buscan todos los marcos negros, existentes en la imagen (desde el punto de vista del resultado luego de umbralizar la imagen, los marcos buscados aparecen de color blanco).
- El interior de cada marco en los marcadores es comparado con un listado de marcadores almacenados en la aplicación.
- Si el interior del marcador coincide con alguno existente en la aplicación, se calcula la posición y orientación relativas de la cámara a la plantilla, y se guarda en una matriz.
- La matriz resultante es utilizada para establecer la posición y orientación de la cámara virtual (transformación de la vista), lo que es equivalente a una transformación de las coordenadas del objeto a dibujar.

- Al posicionar la cámara virtual en el mismo lugar y con la misma orientación que la cámara real, el objeto virtual se dibuja sobre la plantilla, de manera alineada.
- Se repite el proceso por cada nueva imagen capturada.

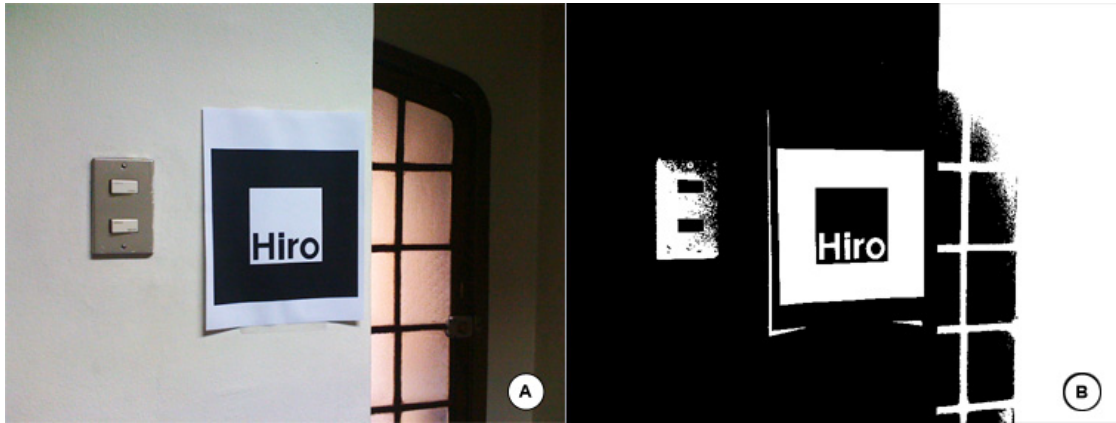


Figura 24: Ejemplo de Umbralización (A) Imagen original (B) Imagen umbralizada.

Existen algunas limitaciones en los sistemas basados en AR asociados a la ciencia de *Computer Vision*, de donde se obtienen los algoritmos de tracking. Naturalmente, los objetos virtuales sólo aparecerán cuando los marcadores se encuentren en el campo visual del dispositivo de captura. Esto puede limitar el tamaño de los marcadores o el movimiento del dispositivo de captura sobre estos. Además, cuanto mayor sea el marcador, a mayor distancia podrá ser detectado.

4.2 Diagramas de clase

4.2.1 Componente FacadeAudio

FacadeAudio es el componente que permite el uso de sonido espacializado en ARTAB. Cuenta con dos clases, *Listener* y *Source*. Para que pueda realizarse una reproducción de audio espacializado durante la ejecución del programa debe existir una única instancia de *Listener*.

Para forzar el hecho anterior, se ha utilizado el patrón *Singleton* (instancia única) el cual está diseñado para restringir la creación de objetos pertenecientes a una clase o restringir el valor de un tipo a un único objeto. Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella [18].

El patrón *Singleton* se implementa creando en la clase un método que crea una instancia del objeto sólo si todavía no existe alguna. Para asegurar que la clase no puede ser instanciada nuevamente se regula el alcance del constructor (con atributos como protegido o privado).

Una solución posible en C++ es conocida como el *Singleton* de Meyers, en la cual el *Singleton* es un objeto local estático. En particular esta solución no es segura en programas *multithread*, pero debido a que ARTAB no funciona con varios *threads* accediendo a esta componente, este problema no es relevante.

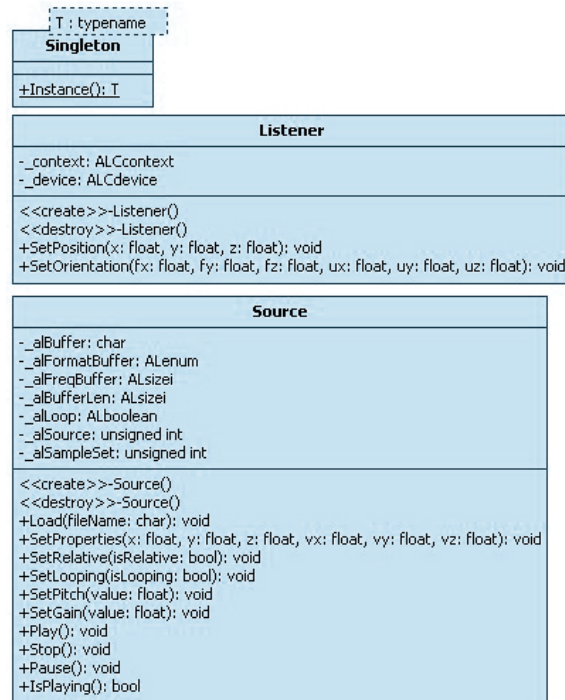


Figura 25: Diagrama de clases de FacadeAudio.

Como se puede ver en el diagrama de clases mostrado en la Figura 25, aparentemente las clases de esta componente no tienen relación entre sí. Como éstas fueron creadas para encapsular funcionalidades de OpenAL, toda la comunicación entre *Listener* y *Source* se realiza a través de OpenAL. Aquí se utilizó el patrón *Facade*, el cual simplifica el acceso a un conjunto de clases proporcionando una única clase que todos utilizan para comunicarse con dicho conjunto de clases [18]. La idea fue tomar este patrón pero separando las funcionalidades en las dos clases ya mencionadas. Los *headers* de ambas clases se muestran a continuación:

Listener.h

```
#ifndef _LISTENER_H
#define _LISTENER_H

#include <windows.h>
#include <stdio.h>
#include <al.h>
#include <AL\alut.h>

namespace FacadeAudio
{
    template<typename T> class Singleton
    {
        public:
            static T& Instance()
            {
                static T singletonIntance;
                return singletonIntance;
            }
    };

    class Listener : public Singleton<Listener>
    {
        friend class Singleton<Listener>;

        public:
            /* Constructor */
            __declspec(dllexport) Listener();
            /* Destructor */
            __declspec(dllexport) ~Listener();
            /* Define la posición del escucha */
            void __declspec(dllexport) SetPosition(float x, float y, float z);
            /* Define la orientación del escucha */
            void __declspec(dllexport) SetOrientation(    float fx,
                                                         float fy,
                                                         float fz,
                                                         float ux,
                                                         float uy,
                                                         float uz);

        private:
            ALCcontext * _context;
            ALCdevice * _device;
    };
}

#endif
```

Source.h

```
#ifndef _SOURCE_H
#define _SOURCE_H

#include <windows.h>
#include <stdio.h>
#include <al.h>
#include <AL\alut.h>
```

```

namespace FacadeAudio
{
    class Source
    {
    public:
        /* Constructor */
        __declspec(dllexport) Source();
        /* Destructor */
        __declspec(dllexport) ~Source();
        /* Carga un sonido WAV */
        void __declspec(dllexport) Load(const char* fileName);
        /* Define la posición y velocidad de la fuente de sonido */
        void __declspec(dllexport) SetProperties(        float x,
                                                         float y,
                                                         float z,
                                                         float vx,
                                                         float vy,
                                                         float vz);

        /* Define si la fuente es o no relativa al escucha */
        void __declspec(dllexport) SetRelative(bool isRelative);
        /* Define si la fuente es o no cíclica */
        void __declspec(dllexport) SetLooping(bool isLooping);
        /* Define el pitch de la fuente */
        void __declspec(dllexport) SetPitch(float value);
        /* Define el gain de la fuente */
        void __declspec(dllexport) SetGain(float value);
        /* Reproduce la fuente */
        void __declspec(dllexport) Play();
        /* Detiene la fuente */
        void __declspec(dllexport) Stop();
        /* Pausa la fuente */
        void __declspec(dllexport) Pause();
        /* Devuelve true si se está reproduciendo */
        bool __declspec(dllexport) IsPlaying();

    private:
        char* _alBuffer;
        ALenum _alFormatBuffer;
        ALsizei _alFreqBuffer;
        ALsizei _alBufferLen;
        ALboolean _alLoop;
        unsigned int _alSource;
        unsigned int _alSampleSet;
    };
}
#endif

```

Para definir un *Listener* basta con utilizar el constructor de clase, definir sus coordenadas con el método *SetPosition* y luego definir la orientación que este tendrá con *SetOrientation*, donde los 3 primeros parámetros corresponden al vector que define dónde está el frente del *Listener* y los últimos 3 definen un vector que apunta donde está la parte superior de éste.

Un *Source* es una fuente de sonido. En el *header* se han definido varios métodos que permiten el manejo de éste, como cargar un archivo, el cual debe ser en formato *wav*, comenzar a reproducir, pausar, detener y además la aplicación de efectos sobre este, alterando por ejemplo su pitch.

4.2.2 Componente FacadeTracker

FacadeTracker busca encapsular alguna librería que permita la realización de tracking visual, en este caso se encapsula la librería ARToolKit. El diagrama de clases puede ser visto en la Figura 26.

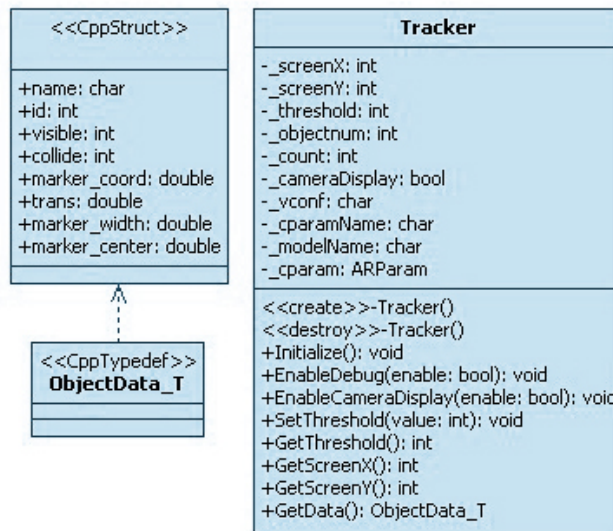


Figura 26: Diagrama de clases de FacadeTracker.

Esta componente cuenta con una clase principal, la cual se encarga, mediante un ciclo (método *GetData*), de capturar un cuadro de video obtenido desde el dispositivo de captura y ver si existen marcadores en éste.

ARToolKit carga unos archivos binarios, los cuales son generados por una aplicación que trae la misma librería, que definen un marcador. Para cargar marcadores propios sólo basta con ejecutar este programa, mover el patrón frente al dispositivo de captura y ARToolKit generará un descriptor del marcador. A su vez, para cargar estos descriptores en la aplicación, haciendo que sean reconocibles, se utiliza un archivo de texto que posee la siguiente estructura:

object data artab

```
#Numero de patrones a reconocer
3
```

```
#Espejo
Espejo
bin/debug/data/patt.hiro
80.0
0.0 0.0

#Globo
Globo
bin/debug/data/patt.sample2
80.0
0.0 0.0

#Escalera
Escalera
bin/debug/data/patt.kanji
80.0
0.0 0.0
```

Este archivo de texto es cargado con la función de ARToolKit *read_ObjData*, que recibe como parámetro la ruta a este archivo. El archivo tiene la siguiente estructura:

- Toda línea que comience con el carácter # es considerada como comentario.
- La primera línea indica cuantos marcadores se desea cargar.
- Luego la secuencia de líneas del archivo es así:
 - Nombre del marcador (para identificarlo en el programa)
 - Ruta del archivo descriptor
 - Dimensión (en milímetros)
 - Centro del marcador (usualmente es 0.0 0.0)

La función *read_ObjData* ha sido encapsulada dentro de *Initialize* en *FacadeTracker*. Estos datos son almacenados en la estructura de datos llamada *ObjectData_T*, teniendo acceso a sus propiedades, como por ejemplo, si está visible y coordenadas. En cada iteración provocada por el método *GetData*, un arreglo de estructuras *ObjectData_T* es retornado con valores actualizados. Esta clase es también responsable de administrar el dispositivo de captura de video. El *header* de la clase *Tracker* se muestra a continuación:

Tracker.h

```
#ifndef _TRACKER_H
#define _TRACKER_H

#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <GL/glut.h>
#include <AR/param.h>
#include <AR/ar.h>
#include <AR/video.h>
#include <AR/gsub.h>
#include "Object.h"

namespace FacadeTracker
{
    class Tracker
    {
    public:
        /* Constructor */
        __declspec(dllexport) Tracker();
        /* Destructor */
        __declspec(dllexport) ~Tracker();
        /* Inicializa captura de camara y tracking */
        void __declspec(dllexport) Initialize();
        /* Habilita o deshabilita el modo Debug */
        void __declspec(dllexport) EnableDebug(bool enable);
        /* Habilita o deshabilita el display de la camara */
        void __declspec(dllexport) EnableCameraDisplay(bool enable);
        /* Setea el valor umbral */
        void __declspec(dllexport) SetThreshold(int value);
        /* Devuelve el valor umbral */
        int __declspec(dllexport) GetThreshold();
        /* Devuelve el ancho de pantalla */
        int __declspec(dllexport) GetScreenX();
        /* Devuelve el alto de pantalla */
        int __declspec(dllexport) GetScreenY();
        /* Realiza el loop de captura y devuelve
        un arreglo con información de los objetos visibles / no visibles */
        ObjectData_T* __declspec(dllexport) GetData();

    private:
        int _screenX;
        int _screenY;
        int _threshold;
        int _objectnum;
        int _count;
        bool _cameraDisplay;
        char* _vconf;
        char* _cparamName;
        char* _modelName;
        ARParam _cparam;
    };
}

#endif
```

4.2.3 ARTAB

En la Figura 27 se muestra el diagrama de clases de ARTAB, incluyendo las clases llamadas desde FacadeAudio y FacadeTracker. Por simpleza en la representación del diagrama se eliminaron las flechas que representan el uso de clases. *SoundWrapper*, *Hud*, *ScreenSector* y *Simulation* utilizan la clase *Utils* y la estructura *Vector3f*, la cual modela un vector de 3 variables.

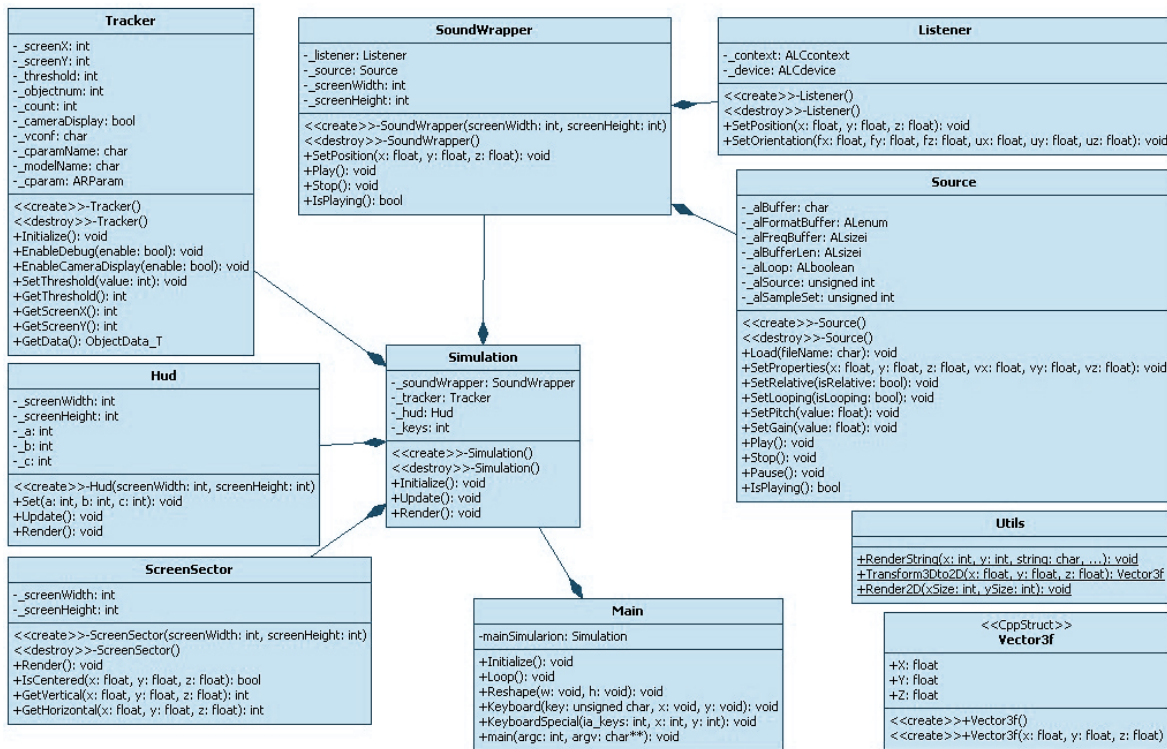


Figura 27: Diagrama de clases de ARTAB.

Para realizar el ciclo de juego se barajaron distintas alternativas, la inicial era utilizar el sistema de mensajes que provee WINAPI, adaptándolo a algo similar a lo propuesto en la Sección 3.4. Se quiso mantener el desarrollo lo más simple posible, es por ello que se optó por GLUT, ya que permite realizar la implementación de este ciclo de manera sencilla. Sin embargo, GLUT es una librería antigua que no ha sido actualizada en años, por lo que a modo de desligar el código de esta librería en caso de que en el futuro se desee realizar cambios por una librería mejor, se crearon las clases *Main* y *Simulation*.

Como se puede ver en el diagrama de clases, la mayoría de las clases son llamadas desde *Simulation* y esta es llamada por *Main* para comenzar el ciclo. El siguiente código resumido pertenece a esta clase:

Main.cpp

```
#include "Globals.h"
#include "Simulation.h"
using namespace Artab;

Simulation * mainSimulation;

void Initialize()
{
    mainSimulation->Initialize();
}

void Loop()
{
    mainSimulation->Update();
    mainSimulation->Render();
    glutSwapBuffers();
}

void Reshape(int w, int h)
{
    /* Código para el redimensionado de pantalla */
}

void Keyboard(unsigned char key, int x, int y)
{
    /* Código para la captura del teclado */
}

void KeyboardSpecial(int a_keys, int x, int y)
{
    /*
     * Código para la captura de teclas especiales como flechas
     * teclas F1 a F12, PageUp, PageDown, etc.
     */
}

void main(int argc, char** argv)
{
    mainSimulation = new Simulation();

    glutInit(&argc, argv);

    Initialize();
    glutSetWindowTitle("ARTAB - Augmented Reality for Assist the Blind");

    glutDisplayFunc(Loop);
    glutIdleFunc(Loop);
    glutReshapeFunc(Reshape);
    glutKeyboardFunc(Keyboard);
    glutSpecialFunc(KeyboardSpecial);
    glutMainLoop();
}
```

Aquí se definen varias funciones que tienen relación con el ciclo de juego, en cada una de ellas según corresponda se llama a un método de la instancia creada de *Simulation*. Esto ofrecerá transparencia en el desarrollo ya que no será necesario preocuparse más del ciclo. En el método *main* se definen todas las funciones que serán utilizadas como "*Callbacks*" o funciones controladoras de eventos para finalmente llamar a *glutMainLoop* y dar comienzo al ciclo del juego.

El siguiente código corresponde al *header* de la clase *Simulation*:

Simulation.h

```
#ifndef _SIMULATION_H
#define _SIMULATION_H

#include "Globals.h"
#include "Utils.h"
#include "Hud.h"
#include "SoundWrapper.h"

namespace Artab
{
    class Simulation
    {
    public:
        /* Constructor */
        Simulation();
        /* Destructor */
        ~Simulation();
        /* Inicialización de variables y estados */
        void Initialize();
        /* Actualización de variables y estados */
        void Update();
        /* Render */
        void Render();

    private:
        SoundWrapper* _soundWrapper;
        Tracker* _tracker;
        Hud* _hud;
    }
}

#endif
```

Como se puede ver en las firmas de los métodos de la clase *Simulation*, éstos tienen correspondencia con las etapas que conforman el ciclo de juego planteado en la Sección 3.4. Éste además se encarga de capturar los eventos que genera la botonera. Para ello existe un arreglo donde cada celda representa una tecla. Éste es asignado en cada ciclo de juego marcando la casilla

correspondiente con 1 o 0 si la tecla fue o no presionada, luego cuando el ciclo llega a la etapa de Update, se revisa el arreglo y se limpia.

Otra clase importante es *SoundWrapper*, la cual llama a las clases de FacadeAudio y permite generar el audio espacializado de ARTAB. Posee métodos que reciben como parámetros las coordenadas de un punto 3D y en base a eso generan una salida audible. El resultado de esta transformación puede verse la siguiente sección. El *header* se muestra a continuación:

SoundWrapper.h

```
#ifndef _SOUND_WRAPPER_H
#define _SOUND_WRAPPER_H

#include "Globals.h"
#include "Utils.h"

#define MAX_TAGS 30

namespace Artab
{
    class SoundWrapper
    {
    public:
        /* Constructor */
        SoundWrapper(int screenWidth, int screenHeight);
        /* Destructor */
        ~SoundWrapper();
        /* Define la posición del escucha */
        void SetPosition(float x, float y, float z);
        /* Reproduce la fuente */
        void Play();
        /* Detiene la fuente */
        void Stop();
        /* Devuelve true si se esta reproduciendo */
        bool IsPlaying();

    private:
        Listener* _listener;
        Source* _source[MAX_TAGS];
        int _screenWidth;
        int _screenHeight;
    };
}

#endif
```

Las Clases restantes son complementarias y se describen a continuación:

- ScreenSector: Define de forma gráfica y lógica los sectores de la pantalla, en base a esto se puede estimar cuando un punto 3D representado por un marcador se encuentra dentro o fuera del sector principal.
- Hud: Esta clase esta encargada de desplegar una pequeña interfaz en pantalla que muestra algunos datos importantes para el facilitador.
- Utils: Esta es una clase con métodos estáticos utilitarios, los cuales permiten imprimir texto en pantalla, dibujar en 2D y convertir puntos 3D a coordenadas de pantalla.

4.3 Interfaces e interacción

Un esquema del funcionamiento de ARTAB puede verse en la Figura 28. El sistema transforma lo capturado con la cámara, la cual se encuentra montada en un sombrero especialmente adaptado, el entorno, y en caso de encontrar marcadores los transforma a sonido espacializado.

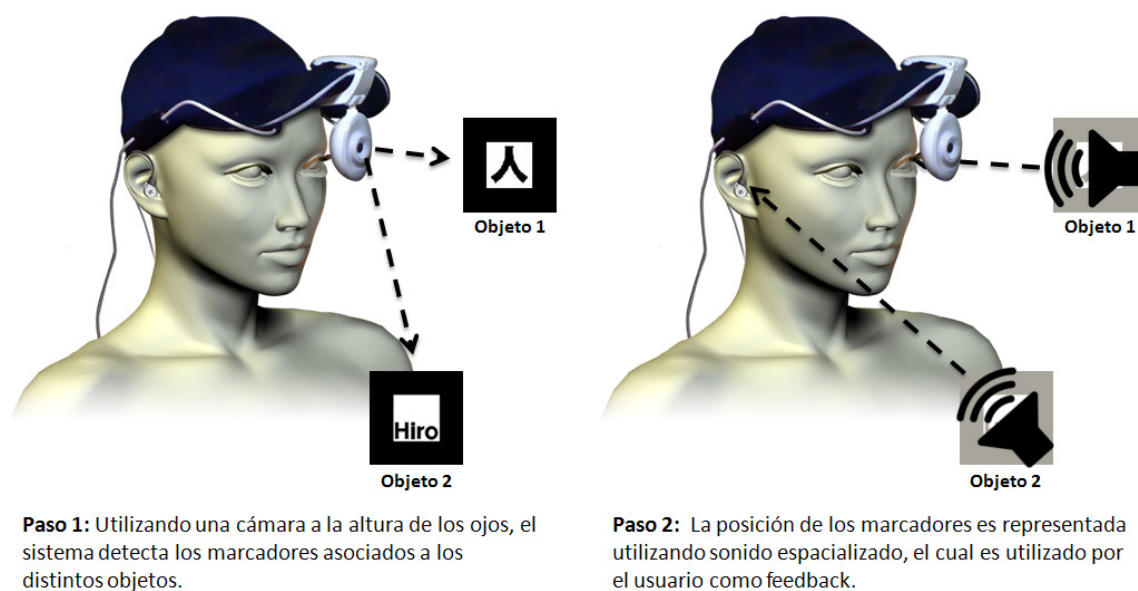


Figura 28: Funcionamiento general de ARTAB.

Se realizaron dos prototipos, los cuales difieren en sus interfaces. El primero fue la base para los estudios de usabilidad y el segundo fue resultado del feedback entregado por los mismos usuarios del sistema.

En el primer prototipo divide el cuadro de video capturado en dos sectores rectangulares concéntricos (ver Figura 29). Cuando un marcador es identificado, dentro del primer sector, se emite un sonido característico constante que alerta al usuario de su presencia dentro del campo visual; cuando éste se posiciona en el centro, correspondiente al segundo sector, el usuario podrá preguntar información sobre el objeto presionando la botonera.

Debido a que la percepción de sonido 3D varía dependiendo del hardware que se utilice, es decir no es lo mismo utilizar cuadrafonía que auriculares estéreo, se ha desarrollado una representación que permite posicionar una fuente de sonido en un entorno 3D variando ciertas propiedades de éste. Para el eje horizontal se varía el canal por donde se emite el sonido constante, el eje vertical varia el pitch del sonido y el eje perpendicular al observador varia el volumen. La variación se hace de manera continua y dentro de los límites que permite el frustrum⁶ (ver Figura 30).

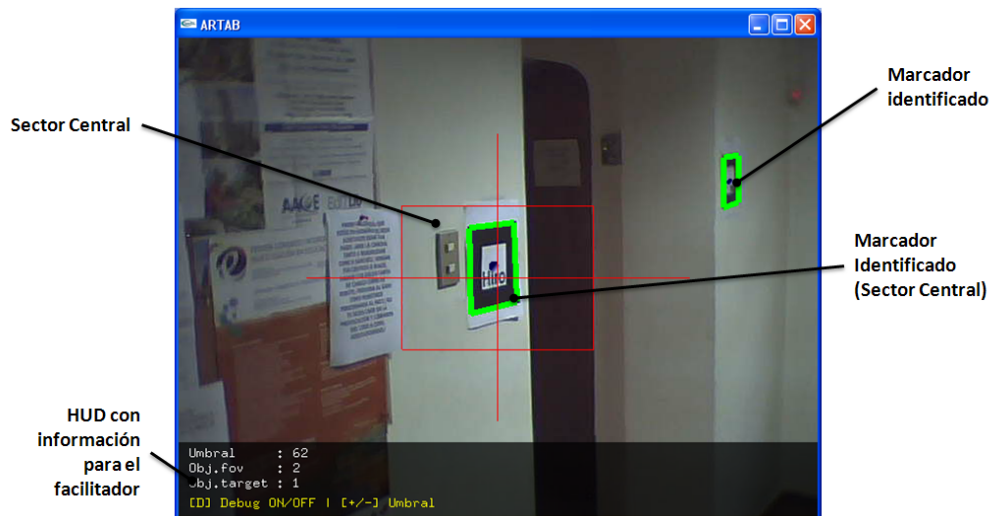


Figura 29: Interfaz del facilitador del primer prototipo.

⁶ Es una porción de una figura geométrica (usualmente un cono o una pirámide) comprendida entre dos planos paralelos. Las intersecciones del sólido con un plano cortante son las bases. En Gráfica 3D delimita lo que la cámara es capaz de hacer rendering en una escena.

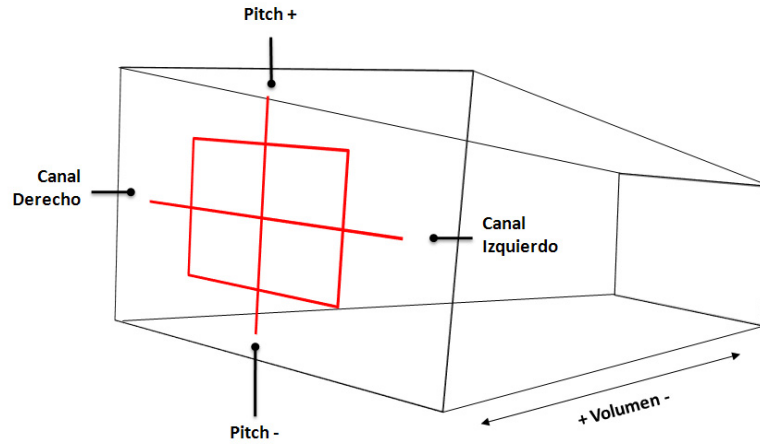


Figura 30: Interfaz basada en audio del primer prototipo.

Según los resultados de la usabilidad, los cuales serán vistos más adelante, se desarrolló el segundo prototipo. Se cambió la forma cómo los objetos (marcadores) son representados con el audio espacial.

Se tomó la imagen capturada y se dividió en 8 filas 11 columnas, generando así una cuadrícula (ver Figura 31). La finalidad de esto es hacer que los sonidos utilizados para la representación espacial fueran discretos, mejorando así el reconocimiento por parte del usuario.

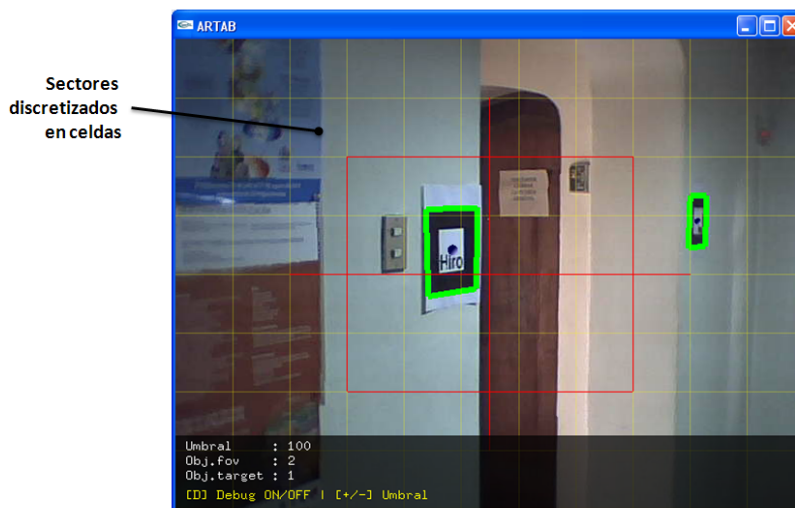


Figura 31: Interfaz del facilitador del segundo prototipo.

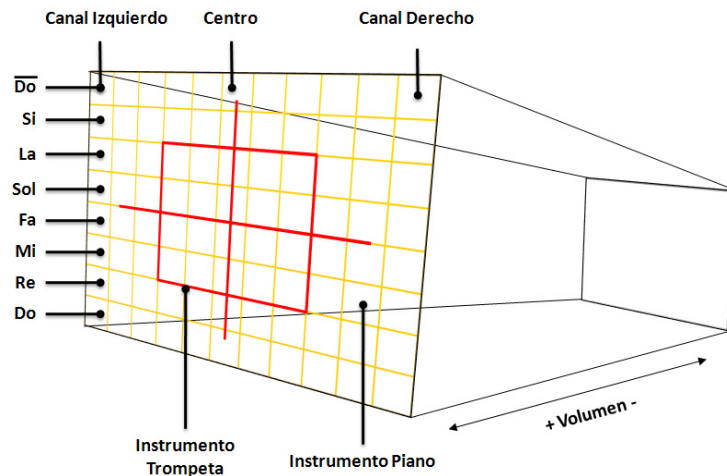


Figura 32: Interfaz basada en audio del segundo prototipo.

En este prototipo cada fila es representada con una nota musical, teniendo así una octava completa que va desde un DO menor a un Do mayor, lo cual representa abajo y arriba respectivamente. En relación al eje vertical, el paso de un sonido desde el canal derecho al izquierdo y viceversa, también es discreto, siendo la columna central la única que hace que el sonido se escuche en ambos canales (ver Figura 32).

Además, se amplió el sector central que permite preguntar por un objeto. A modo de dar mayor *feedback* al usuario sobre la posibilidad que un objeto se encuentre en el sector central, el sonido utilizado como alarma cambia el timbre, es decir, cuando la alarma sonora viene desde fuera del sector central se utiliza el sonido de un instrumento de cuerdas (piano) y si se encuentra dentro, se utiliza el sonido de un instrumento de viento (trompeta).

Con respecto al input del usuario, ambos prototipos utilizan la misma botonera con la misma función, como indica la Figura 33. Si se presiona el botón para pregunta, el sistema emitirá la frase “Nada al frente”, utilizando TTS, si no existe ningún objeto, o el nombre de todos los objetos uno a uno, que se encuentren en el sector central.



Figura 33: Botonera inalámbrica utilizada en ARTAB.

4.4 Evaluación funcional

La evaluación funcional de ARTAB se llevó a cabo mediante pruebas basadas en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software. Evaluaciones funcionales se hacen mediante el diseño de modelos de prueba que buscan evaluar cada una de las opciones con las que cuenta el paquete informático.

Estas pueden realizarse en base a 2 enfoques principales:

- Prueba de caja blanca: Se basa en el minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con el estado deseado o esperado. La prueba de caja blanca es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba.
- Pruebas de caja negra: Se centran en los requisitos funcionales del software, es decir, la prueba de caja negra permite obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. La prueba de caja negra intenta encontrar errores de las siguientes categorías:
 - Funciones incorrectas o ausentes
 - Errores de interfaz
 - Errores en estructuras de datos
 - Errores de rendimiento
 - Errores de inicialización y de terminación

En ARTAB, la mayoría de las pruebas de caja blanca estuvieron enfocadas a corroborar que las distintas partes involucradas en el desarrollo funcionaran entre sí, eso quiere decir que el ciclo de juego llamará correctamente a FacadeTracker y FacadeAudio, que FacadeTracker inicializará la cámara de manera correcta y cargará correctamente los descriptores de los marcadores, entre otros.

Estas pruebas se realizaron constantemente a lo largo del proceso de desarrollo, y al finalizar los prototipos se realizaron pruebas de caja negra descritas en la siguiente tabla:

Prueba	Descripción	Resultado esperado
01	El sistema no captura ningún marcador en pantalla.	El sistema no emite ningún sonido en la interfaz basada en audio.
02	El sistema captura un marcador fuera del sector central.	El sistema emite alarma auditiva la cual debe ser espacializada según corresponda.
03	El sistema captura un marcador dentro del sector central.	El sistema emite alarma auditiva la cual debe ser espacializada según corresponda.
04	El Sistema captura varios marcadores fuera del sector central.	El sistema emite una alarma auditiva por cada marcador visualizado, la cual debe ser espacializada según corresponda.
05	Sistema ve varios marcadores en dentro del sector central	El sistema emite una alarma auditiva por cada marcador visualizado, la cual debe ser espacializada según corresponda.
06	El Usuario pregunta al sistema si hay marcador en el sector central y no hay	El sistema debe responder usando TTS que no hay objetos frente a él. Cualquier alarma auditiva deberá silenciarse durante ese período.
07	Usuario pregunta por marcador y hay uno en el sector central	El sistema debe responder usando el TTS asociado a ese marcador. Cualquier alarma auditiva deberá silenciarse durante ese período.
08	Usuario pregunta por marcador y hay varios en el sector central	El sistema debe responder usando el TTS asociado a cada marcador nombrándolos en secuencia arriba – abajo e izquierda – derecha. Cualquier alarma auditiva deberá silenciarse durante ese periodo.

Tabla 1: Pruebas de caja negra.

Capítulo 5: Evaluación de la usabilidad

5.1 Introducción

Con el objetivo de estudiar la usabilidad de ARTAB, se aplicaron dos tipos de evaluaciones: Una evaluación preliminar de interfaces de forma cualitativa y cuantitativa con un primer prototipo y una segunda evaluación de usabilidad de carácter cuantitativo con un segundo prototipo corregido. La primera evaluación tuvo como objetivo principal evaluar la iconografía sonora definida para ARTAB, una observación de cómo los usuarios interactúan con la interfaz del sistema y la aplicación de un cuestionario de usuario final. La segunda evaluación corresponde a evaluar si efectivamente las correcciones realizadas en base a las observaciones sobre el primer prototipo por parte de los usuarios produjeron un real impacto.

5.2 Participantes

Para la evaluación de usabilidad de ARTAB se consideró una muestra de 10 usuarios totalmente ciegos cuyas edades fluctúan entre 15 y 50 años. Todos ellos son de la ciudad de Santiago de Chile. La muestra constó de 4 mujeres y 6 hombres, ninguno de ellos con déficits neurológicos. Todos los sujetos de la muestra son usuarios del software JAWS (el cual permite leer el contenido que muestra

en pantalla un computador) y por tanto están acostumbrados a escuchar instrucciones TTS a velocidades altas (comparada con una velocidad normal de conversación).

5.3 Instrumentos

Se utilizó una pauta de usabilidad icónica, la cual permitió cuantificar en qué grado los sonidos utilizados en ARTAB son reconocibles. Este cuestionario pregunta acerca de la calidad de los sonidos asociados con las distintas acciones en el simulador, tales como objetos, alertas y TTS.

Todos los usuarios realizaron un apresto antes de utilizar ARTAB, para el cual se construyó el material que aparece en la Figura 34. Este material es una representación de lo que la cámara captura. Esta se divide en sectores, central y periférico. Este último se encuentra subdividido en cuadrantes. Las fronteras de estos cuadrantes se encuentran en relieve, para que el usuario ciego pueda distinguirlos de manera táctil.

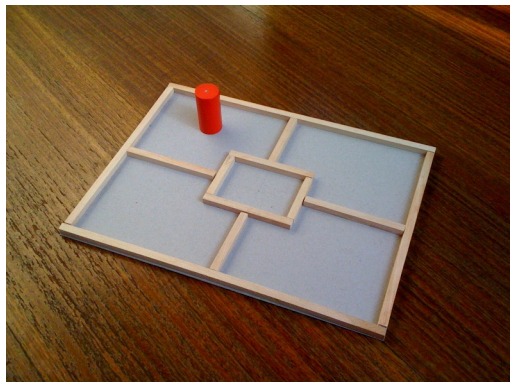


Figura 34: Material de apresto.

Se aplicó además una pauta de evaluación de preguntas abiertas, la cual incluía preguntas como: ¿Te fue posible saber tu posición en relación a los objetos? ¿Fue de tu agrado el *feedback* sonoro? ¿La información que te entrega el software, es suficiente para saber qué hacer y lo que sucede? La idea es recopilar información sobre aspectos relacionados con M&O que constituyen el foco del software, así como el uso de los controles, la información proporcionada por el software, y la navegación del usuario. Los resultados de esta evaluación permiten rediseñar y mejorar las interfaces.

Finalmente, una pauta de usuario final (basado en el Software *Usability for Blind Children Questionnaire* [65]) es aplicada, la cual cuenta de 14 sentencias que los usuarios deben definir en

qué medida se cumplieron, utilizando una escala de 1 ("un poco") a 10 ("mucho"). Los resultados permiten evaluar la usabilidad del software de acuerdo a la satisfacción del usuario, utilizando sentencias como “Me gusta el software”, “El software es motivador” y “Volvería a utilizar el software”. Se puede evaluar el control y uso utilizando sentencias como “Me sentí controlando las situaciones del software”, “El software es interactivo” y “El software es fácil de utilizar”. Finalmente, también se puede evaluar el uso de sonidos con sentencias como: “Me gustan los sonidos del software”, “Los sonidos del software son claramente identificables” y “Los sonidos del software me transmiten información”.

5.4 Procedimiento

Se aplicaron tres pautas de usabilidad. Primero se aplicó la pauta de usabilidad icónica y un apresto, los cuales fueron completados en una sesión de 20 minutos. Para cada usuario, el evaluador probó el timbre y velocidad de los TTS utilizados en la aplicación. Luego, a los usuarios se les presentó el material de apresto y se les explicó en qué consistía (ver Figura 35). Luego se reprodujeron sonidos correspondientes a la ubicación espacializada del objeto (se descartó profundidad). Por cada sonido reproducido, los usuarios debían posicionar un cilindro en el sector que representara la ubicación del sonido en el determinado cuadrante. Los sonidos reproducidos fueron 5, todos con diferentes posiciones.



Figura 35: Usuario realizando apresto.

A medida que el usuario trabajaba con el material y los sonidos eran reproducidos, el facilitador observaba el desempeño y si el usuario presentaba algún tipo de problema.

Luego, los usuarios trabajaron con ARTAB durante una sesión de 40 minutos. Para ello se dispuso una serie de marcadores en un conjunto de habitaciones, donde cada uno podía corresponder a un objeto ficticio o no (esto no es relevante, ya que lo que se desea evaluar es si los usuarios logran o no encontrar los marcadores). Luego el facilitador iba asignando tareas a los usuarios, como por ejemplo, ir encontrando objetos libremente y luego encontrar un objeto en particular. Durante todo el proceso el facilitador observaba a través de una interfaz gráfica qué era lo que observaban los usuarios y si estos lograban centrar los objetos gracias al *feedback* sonoro, como aparece en la Figura 36.



Figura 36: Usuario realizando tarea con ARTAB. (A) El usuario observa el marcador que indica que hay una escalera a la derecha de este; (B) El facilitador observa gracias a una interfaz gráfica, lo que enfoca el usuario.

Cuando el usuario termina con las tareas asignadas, éste procede a responder la pauta de usuario final con ayuda del facilitador, quién lee las preguntas en voz alta y completa las respuestas. Con el resultado de estas pautas se procede luego a hacer un rediseño del primer prototipo.

Para el segundo prototipo los usuarios realizan las tareas relacionadas con la búsqueda de marcadores que representan objetos y luego contestan la pauta de usuario final, pudiendo así contrastar resultados.

5.5 Resultados y análisis de usabilidad para primer prototipo

5.5.1 Análisis cualitativo

5.5.1.1 Resultados de pauta usabilidad icónica

De la primera evaluación relacionada con el uso de TTS para entregar *feedback* a los usuarios, se obtiene que algunos se encuentran muy influenciados por el uso de programas lectores de pantalla como JAWS. La apuesta inicial en el diseño de ARTAB era utilizar TTS pregrabados que fueran lo más agradables y claros posible, para ello se utilizó una voz con acento argentino de Loquendo a una velocidad normal.

El resultado de la evaluación arrojó que este acento era preferido por sobre el acento chileno y español, por ser más claro a la hora de expresar los textos de forma auditiva, sin embargo la velocidad normal no fue del todo aceptada, ya que como la mayoría de los participantes en el testeo son usuarios de JAWS, están acostumbrados al uso de TTS con velocidades más altas, obteniendo así mayor información en menos tiempo.

5.5.1.2 Resultados de apresto

Se pudo observar que la asociación del pitch con la altura sólo se logró cuando los usuarios comparaban dicho sonido con el que se emite si el objeto está correctamente centrado. Esta comparación hace que los usuarios dediquen más tiempo a la alineación vertical versus la horizontal. Esto requirió que se tuviese que entregar asistencia para indicar cuándo el objeto se encontraba centrado verticalmente, para que ellos luego identificaran si la fuente originalmente se encontraba arriba o debajo de ellos.

El modelo mental de los usuarios les permitió asociar horizontalidad con el uso del canal izquierdo y derecho, profundidad con el volumen, pero en relación a la altura no hubo unanimidad en como representarla. Esta dispersión se pudo apreciar en la evaluación realizada y que es clave para poder otorgar autonomía a los usuarios ciegos.

5.5.1.3 Resultados de observación de Interacción

De los resultados obtenidos en la etapa de observación, se pudo identificar ciertos puntos relacionados con la interacción de los usuarios con ARTAB. Estos puntos fueron el resultado de la

observación de los usuarios utilizando el sistema y de la revisión acuciosa de videos grabados (tanto de la interacción como de lo que el sistema registra durante la interacción):

- La localización de objetos de manera horizontal se hace sencilla para los usuarios, pero no así la vertical, en la cual se invierte mayor tiempo para centrar el objeto.
- Se tiende a utilizar poco la función de consulta de objeto a modo de asistencia para saber si el objeto es identificable y por ende si se encuentra centrado.
- Los usuarios logran manejar la situación cuando se encuentran frente a varios objetos, pudiendo enfocar cada uno. Aún así, la tarea se torna compleja cuando los objetos se encuentran alineados verticalmente.
- Al utilizar la botonera, los usuarios tienden a presionar el botón y no soltarlo, produciendo que la información entregada mediante TTS se repita.
- Los desplazamientos realizados con el cuello de los usuarios son en su mayoría correctos, haciendo barridos verticales y horizontales. Existen casos menores donde los usuarios inclinan la cabeza hacia los lados para enfocar, lo cual no es un problema, pero implica tener que insistir sobre el correcto uso del sistema.
- Algunos desplazamientos por parte de los usuarios son muy rápidos, no dejando tiempo al sistema para detectar los marcadores e interpretarlos.

5.5.1.4 Resultados de pauta de usabilidad con preguntas abiertas

Sobre las preguntas de la pauta de usabilidad con preguntas abiertas, los usuarios reconocen que ARTAB les permite ubicar de manera inmediata objetos de interés dentro del contexto en el cual se utiliza el sistema. Creen que el uso de los TTS es adecuado y que efectivamente esta herramienta puede ser útil. Sin embargo, se pudo notar que en varios casos el *feedback* sonoro para representar la posición de un objeto no era de todo el agrado de los usuarios, sugiriendo que en lugar de poner un sonido similar al de un radar se utilizaran notas musicales, las cuales según ellos son más fáciles de reconocer, que indicasen la posición vertical del objeto. En relación a la profundidad, el aumento o disminución del volumen debería ser más notorio y además la botonera se debería aprovechar de mejor manera, ya que al tener 4 botones, los usuarios piensan que en los 3 restantes podrían agregarse nuevas funcionalidades. Junto con esto, los usuarios que probaron ARTAB sintieron que

podía ser útil e inclusive encontraron otras situaciones, como por ejemplo, etiquetar frascos con medicamentos, en los cuales un sistema con estas características podría facilitarles la vida haciéndolos más independientes.

5.5.2 Análisis cuantitativo

En la Figura 37 se muestran los resultados de la pauta de usabilidad final aplicada para el primer prototipo de ARTAB.

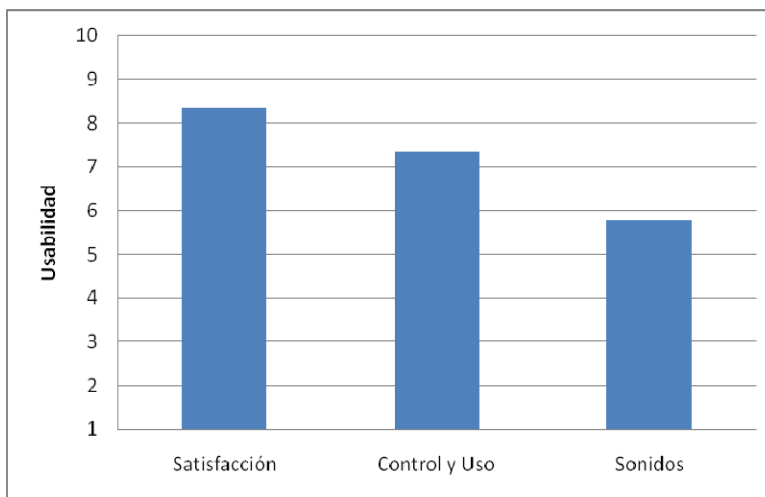


Figura 37: Gráfico de resultados totales de la encuesta de satisfacción (usabilidad) del primer prototipo.

Los resultados arrojados por esta encuesta muestran que ARTAB es altamente usable y entendible por los usuarios ciegos, arrojando resultados de “Satisfacción” igual a 8.4 puntos y de “Control y Uso” de 7.4 puntos de una escala máxima de 10 puntos. Para la categoría de “Satisfacción”, las sentencias que obtuvieron mayor puntaje por parte de los usuarios encuestados son “Volvería a utilizar este software” y “El software es motivador” con un promedio de 9.3 puntos.

La categoría “Control y Uso” obtuvo un puntaje de 7,4 puntos, debido principalmente al hecho que el sistema no permite el control total por los usuarios ciegos. Es decir, que no podía acceder a menús iniciales de configuración, ni pueden iniciar una nueva navegación, por lo cual están limitados a estar con una persona vidente como facilitador a fin de iniciar una sesión. Esto se reflejó en la puntuación obtenida por la declaración: “Me sentí controlando las situaciones del software”, la cual obtuvo 6 puntos de un total de 10, correspondiente a la más baja evaluación en esta categoría. La más alta evaluación obtenida en “Control y Uso” corresponde a “El software es fácil de usar”

con un promedio de 8,7 puntos. Este resultado denota la facilidad de uso alcanzado con ARTAB. La declaración con la segunda puntuación más alta fue "El software es interactivo", con 7,7 puntos.

En la categoría "Sonidos" se obtuvo para las sentencias "Me gustan los sonidos del software" y "Los sonidos del software son claramente identificables" el puntaje correspondiente a 5,3 puntos. La tercera sentencia correspondiente a declaración "Los sonidos del software me transmiten información" obtuvo 6,7 puntos. El problema fue que los usuarios ciegos no lograban diferenciar la variación del pitch asociada a arriba y abajo en la ubicación de objetos.

5.6 Resultados de usabilidad para el segundo prototipo

5.6.1 Análisis cuantitativo

Analizados los resultados de usabilidad del primer prototipo, se realizaron las modificaciones y necesarias y rediseño para generar un segundo prototipo, el cual fue evaluado por los mismos usuarios del primer prototipo. En la Figura 38 se muestran los resultados de la pauta de usabilidad final aplicada para al segundo prototipo de ARTAB, comparados con los del primer prototipo.

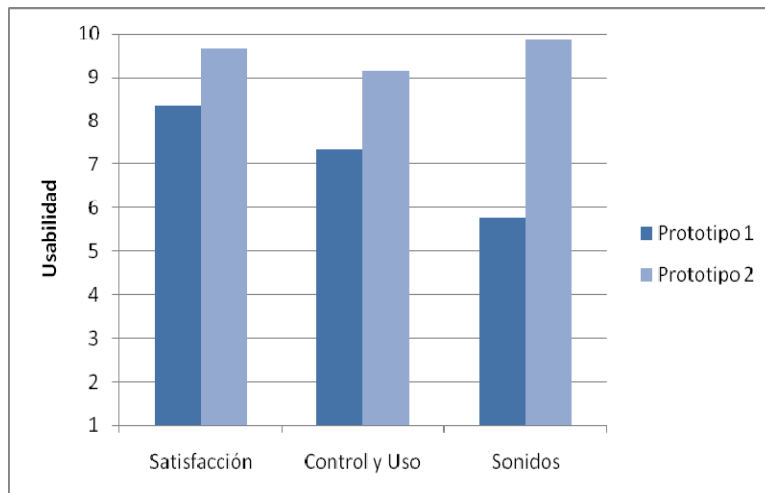


Figura 38: Gráfico de resultados totales de la encuesta de satisfacción (usabilidad) del segundo prototipo versus el primero.

Los resultados arrojados por esta encuesta muestran que ARTAB mejoró bastante en comparación con el primer prototipo. La categoría “Satisfacción” obtuvo 9,7 puntos, “Control y Uso” obtuvo 9,2 y Sonidos obtuvo 9,9.

Cada sentencia tuvo una variación positiva en relación a la encuesta realizada para el primer prototipo. La sentencia con menor variación corresponde a la categoría “Satisfacción”, la cual dice “El software me permitió entender nuevas cosas”, con una puntuación de 8,6 puntos y una variación de 0,3.

Las mayores variaciones se presentan en la categoría “Sonido”, obteniendo la sentencia “Los sonidos del software son claramente identificables” una puntuación de 10 y una variación de 4,7, siendo la mayor de todas. Este resultado es muy importante, ya que en la usabilidad del primer prototipo presentó el peor puntaje, lo cual significa que los cambios en la interfaz que implican el uso de sonido dieron el resultado esperado.

La categoría “Control y Uso” obtuvo el menor puntaje de las 3 categorías. Este resultado se debe a que para los usuarios ciegos aún es complicado tener que realizar los movimientos de manera pausada a modo de que la cámara pueda capturar los marcadores y permitir que ARTAB realice la interpretación de los marcadores.

El remplazo de un sonido continuo que variara el pitch, versus uno discreto que utilizara notas musicales para indicar la altura, hizo que la percepción completa de ARTAB mejorara por parte de los usuarios finales.

Capítulo 6: Conclusiones y trabajo futuro

6.1 Conclusiones

El objetivo de esta investigación ha sido diseñar, desarrollar y evaluar la usabilidad de ARTAB, un sistema que funciona como asistente tecnológico utilizando AR e interfaces basadas en audio, para ayudar a un usuario ciego a identificar elementos de interés dentro de un entorno *indoor*, potenciando habilidades de M&O.

ARTAB es una herramienta que busca asistir a personas ciegas en contextos donde se hace necesario identificar ciertos objetos de interés para realizar alguna actividad. Se buscó utilizar una representación que permitiera a los usuarios identificar la posición de un objeto relativo a ellos, utilizando fuentes de audio espacializado mediante audífonos estéreo. Debido a las características del hardware utilizado, se optó por aplicar distintos efectos sonoros a cada eje del espacio virtual donde se encuentran representados los elementos que visualiza el dispositivo de captura de video que porta el usuario en un sombrero especialmente adaptado.

Según los resultados de usabilidad, arrojados por la pauta de usabilidad con preguntas abiertas, es claro que la representación en el eje horizontal de una fuente sonora, usando los canales izquierdo y

derecho respectivamente, es natural para los usuarios. Durante el testeo, ninguno de los usuarios tuvo problemas para alinear los objetos dicho eje.

Un punto negativo que se detectó fue el uso del efecto pitch en el primer prototipo. Si bien en la tarea de apresto se les indicó a los usuarios el significado de su variación, les fue bastante complicado detectar la posición utilizando el pitch de manera continua. Es decir, los usuarios necesitan variar de manera manual el sonido, moviendo su cabeza de arriba abajo, para saber la posición vertical del objeto representado de forma espacial.

Los usuarios calificaron que la representación de la distancia por medio del volumen es adecuada, otorgando un buen *affordance*⁷. Sin embargo, sugirieron que la variación de esta fuera más drástica, ya que muchos sintieron que a esta propiedad del sonido le faltaba contraste, como para realmente obtener algún tipo de información.

Aún con algunos puntos negativos de la usabilidad del primer prototipo los usuarios sintieron que ARTAB es una aplicación útil que les podría ayudar en sus tareas de M&O.

Al terminar la evaluación de usabilidad del primer prototipo de ARTAB, los usuarios quedaron muy entusiasmados con la aplicación, encontrándola útil y novedosa. Pero aún así esperaban que se hicieran las correcciones en función de las observaciones hechas por ellos mismos.

En el segundo prototipo, la parte superior e inferior se dividió en 4 franjas cada una, haciendo un total de 8 franjas. Cuando un objeto se encuentre dentro de una franja en particular, este será representado verticalmente con una nota musical, obteniendo una octava que va desde un Do menor a un DO mayor. Al discretizar la representación del movimiento vertical y horizontal se logró que los usuarios pudiesen alinear el objeto, representado por marcadores, de mejor manera.

Uno de los problemas, asociado a la librería ARToolKit, es el hecho que los usuarios deben adaptarse a hacer movimientos más pausados para encontrar los marcadores, ya que la librería no es capaz de identificarlos si los movimientos son rápidos.

⁷ En HCI (*Human Computer Interaction*) es un principio de diseño que está relacionado con la impresión de una posible funcionalidad que brinda un objeto al usuario.

6.2 Trabajo futuro

Ya validada la interfaz de ARTAB, se busca enfocar los esfuerzos a mejorar el sistema de tracking visual, utilizando otras librerías de software. Lo anterior puede llevar a trabajar directamente con algoritmos de *Computer Vision*, transformando los datos obtenidos de una imagen a información que pueda ser desplegada por nuestra interfaz, utilizando para ello OpenCV.

Se desea agregar mayor funcionalidad a la botonera, permitiendo al usuario preguntar a qué distancia aproximada, en metros, se encuentra de un objeto. Esto se logra asociando a cada marcador su tamaño y estimando, en base a las dimensiones que posee éste en el cuadro de video donde fue capturado, la distancia al dispositivo de captura.

Finalmente, se está estudiando la opción de acercar el sistema a la realidad de los usuarios ciegos proporcionando la posibilidad de migrar ARTAB a otras plataformas de hardware a las que ellos tengan acceso, tales como *smartphones* y *pocket pc*. Para ello, es necesario considerar la capacidad de procesamiento del dispositivo y si es posible realizar el tracking de los marcadores de manera óptima, mientras de forma simultánea, se soporta la carga que requiere la generación de *feedback* audible de manera especializada.

Referencias

- [1] 16 Top Augmented Reality Business Models | PERSONALIZE MEDIA.
<http://www.personalizemedia.com/16-top-augmented-reality-business-models/> (Último acceso: 09/05/2010)
- [2] Almgren, J., Carlsson, R., Erkkonen, H., Fredriksson, J., Møller, S., Rydgard H., Österberg M., Bötschi K., Voegtli B. and Fjeld M. (2005): Tangible User Interface for Chemistry Education: Visualization, Portability, and Database. Proceedings of SIGRAD 2005, Linköping University Electronic Press, pp. 19-24..
- [3] Andriole S. and Roberts E. (2008) Point/Counterpoint Technology curriculum for the early 21st century. Commun. ACM, vol. 51, pp. 27-32
- [4] Angin P., Bhargava B. and Helal S. (2010) A Mobile-Cloud Collaborative Traffic Lights Detector for Blind Navigation, Eleventh International Conference on Mobile Data Management, pp.396-401
- [5] ARToolKit Homepage <http://www.hitl.washington.edu/artoolkit/> (Último acceso: 09/05/2010)
- [6] Augmented Reality Browser: Layar. <http://www.layar.com/> (Último acceso: 09/05/2010)
- [7] Augmented reality lets drivers see through road obstacles.
<http://www.techchee.com/2009/10/24/augmented-reality-lets-drivers-see-through-road-obstacles/> (Último acceso: 09/05/2010)
- [8] Azuma R., Baillot Y., Behringer R., Feiner S., Julier S., and MacIntyre B. (2001). Recent Advances in Augmented Reality. IEEE Comput. Graph. Appl. 21, 6, 34-47
- [9] Azuma, R. (1997) "A Survey of Augmented Reality," In Presence: Teleoperators and Virtual Environments 6, 4 (August 1997), 355-385.
- [10] Bimber O. and Raskar R. (2005) Spatial augmented reality: Merging real and virtual worlds. Wellesley, Mass: A K Peters

- [11] BMW Augmented Reality.
http://www.bmw.com/com/en/owners/service/augmented_reality_introduction_1.html
(Último acceso: 09/05/2010)
- [12] Bologna, G., Deville, B., Pun, T., and Vinckenbosch, M. (2007). Transforming 3D coloured pixels into musical instrument notes for vision substitution applications. *J. Image Video Process.* 2007, 2 (Aug. 2007), 8-8.
- [13] Bourbakis N. (2008), "Sensing Surrounding 3-D Space for Navigation of the Blind," *Engineering in Medicine and Biology Magazine, IEEE*, vol. 27, pp. 49-55.
- [14] Crossan A. and Brewster S. (2006) Two-Handed Navigation in a Haptic Virtual Environment. *Proceedings of International Conference on Human Factors in Computing Systems*, pp.676-681, Montréal, Québec, (Canada)
- [15] Dramas, F., Oriola, B., Katz, B. G., Thorpe, S. J., and Jouffrais, C. (2008). Designing an assistive device for the blind based on object localization and augmented auditory reality. In *Proceedings of the 10th international ACM SIGACCESS Conference on Computers and Accessibility Halifax, Nova Scotia, Canada, October 13 - 15, 2008. Assets '08.* ACM, New York, NY, 263-264.
- [16] El código secreto de Paula. <http://www.paula.cl/blog/tendencias/2010/02/08/realidad-aumentada/> (Último acceso: 09/05/2010)
- [17] Fonadis (2004). *Primer Estudio Nacional de la Discapacidad, Discapacidad Visual.* ENDISC Chile 2004. Gobierno de Chile.
- [18] Gamma, E. (1995). *Design patterns: elements of reusable object-oriented software.* Addison-Wesley professional computing series. Reading, Mass: Addison-Wesley.
- [19] Holzinger, A. (2005). Usability engineering methods for software developers. *Commun. ACM* 48, 1 (Jan. 2005), 71-74.
- [20] Home – OpenAI <http://connect.creativelabs.com/openai/default.aspx> (Último acceso: 09/05/2010)
- [21] International Standard (1999). ISO 13407. Human-centered design processes for interactive systems.

- [22] ISO/IEC. 9241-14 Ergonomic requirements for office work with visual display terminals (VDT)s - Part 14 Menu dialogues, ISO/IEC 9241-14: 1998 (E), 1998.
- [23] Jacquet C., Bellik, Y. and Bourda, Y. (2006) Electronic Locomotion Aids for the Blind: Towards More Assistive Systems. Intelligent Paradigms for Assistive and Preventive Healthcare. N. Ichalkaranje, A. Ichalkaranje, LC Jain, editores. Springer-Verlag, Abril, pp. 133-163
- [24] Kaehler, A. and Bradski, G. (2008). Learning OpenCV. Farnham: O'Reilly.
- [25] Kaufmann, H. and Schmalstieg, D. (2002). Mathematics and geometry education with collaborative augmented reality. In ACM SIGGRAPH 2002 Conference Abstracts and Applications (San Antonio, Texas, July 21 - 26, 2002). SIGGRAPH '02. ACM, New York, NY, 37-41.
- [26] Kroeker, K. L. (2010). Mainstreaming augmented reality. Commun. ACM 53, 7 (Jul. 2010), 19-21.
- [27] La Realidad Aumentada llega a la Publicidad - video oficial de Universal McCann.
<http://www.youtube.com/watch?v=NnMFqhLYtwg> (Último acceso: 09/05/2010)
- [28] Lahav O. and Mioduser D. (2008) Haptic-feedback support for cognitive mapping of unknown spaces by people who are blind. International Journal on Human-Computer Studies 66, pp. 23–35
- [29] Loomis J., Marston J., Golledge R. and Klatzky R. (2005) Personal guidance system for visually impaired people: Comparison of spatial displays for route guidance. Journal of Visual Impairment Blindness 99, 4, pp. 219-232
- [30] Microsoft DirectX <http://www.microsoft.com/games/en-us/aboutgfw/pages/directx.aspx> (Último acceso: 09/05/2010)
- [31] Milgram, P. and Kishino, A. F. (1994) Taxonomy of Mixed Reality Visual Displays IEICE Transactions on Information and Systems, E77-D (12), pp. 1321-1329.

- [32] Mioduser, D. and Lahav, O. (2004). Anticipatory cognitive mapping of unknown spaces by people who are blind using a virtual learning environment. In Proceedings of the 6th international Conference on Learning Sciences (Santa Monica, California, June 22 - 26, 2004). International Conference on Learning Sciences. International Society of the Learning Sciences, 334-341.
- [33] Mixed Reality Learning Media: effects in classroom education. Available: <http://www.youtube.com/watch?v=iT2ek8N0VIY> (Último acceso: 09/05/2010)
- [34] Nielsen J. (1993) Usability engineering. New York: Academic Press Professional.
- [35] Nielsen, J., (2000) Usability 101: Introduction to Usability <http://www.useit.com/alertbox/20030825.html> (Último acceso: 09/05/2010)
- [36] NOVINT http://home.novint.com/products/novint_falcon.php (Último acceso: 09/05/2010)
- [37] OpenCV Wiki <http://opencv.willowgarage.com/wiki/> (Último acceso: 09/05/2010)
- [38] Pentenrieder, K., Bade, C., Doil, F., and Meier, P. (2007). Augmented Reality-based factory planning - an application tailored to industrial needs. In Proceedings of the 2007 6th IEEE and ACM international Symposium on Mixed and Augmented Reality (November 13 - 16, 2007). Symposium on Mixed and Augmented Reality. IEEE Computer Society, Washington, DC, 1-9.
- [39] PHANTOM OMNI - Sensable. <http://www.sensable.com/haptic-phantom-omni.htm> (Último acceso: 09/05/2010)
- [40] Probert, P., Lee, D. and Kao, G. (1996) Interfaces for multi-sensor systems for navigation for the blind. Proceedings The First European Conference on Disability, Virtual Reality and Associated Technologies 8-10 July 1996, Maidenhead, UK, pp. 209-217
- [41] Reed, A. (2009). Learning XNA 3.0. Farnham: O'Reilly.
- [42] Revista Esquire apuesta por la realidad aumentada. <http://www.fayerwayer.com/2009/10/revista-esquire-apuesta-por-la-realidad-aumentada/> (Último acceso: 09/05/2010)

- [43] Sáenz M. and Sánchez J. (2010) Indoor Orientation and Mobility for Learners Who are Blind. 15th Annual Cybertherapy & CyberPsychology Conference, CyberTherapy 2010, Seoul, Korea, June 13-15, 2010 (accepted, in press)
- [44] Sánchez J. (2003) Software Usability for Blind Children Questionnaire (SUBC), Usability evaluation test, University of Chile
- [45] Sánchez J., Sáenz M. and Ripoll M. (2009) Usability of a Multimodal Videogame to Improve Navigation Skills for Blind Children. Proceedings of the Eleventh International ACM SIGACCESS Conference on Computers and Accessibility, Pittsburgh, PA, USA, October 26-28, 2009, pp. 35-42
- [46] Sánchez, J. and Galaz, I. (2007) AudioStoryTeller: Enforcing Blind Children Reading Skills. In C. Stephanidis (Ed.): Universal Access in HCI, Part III, HCII 2007, LNCS 4556, pp. 786–795, 2007 © Springer-Verlag Berlin Heidelberg 2007
- [47] Sánchez, J. and Sáenz, M. (2005). 3D sound interactive environments for problem solving. In Proceedings of the 7th international ACM SIGACCESS Conference on Computers and Accessibility (Baltimore, MD, USA, October 09 - 12, 2005). Assets '05. ACM, New York, NY, 173-179.
- [48] Sánchez, J. and Sáenz, M. (2006) 3D sound interactive environments for blind children problem solving skills. Behaviour & Information Technology, Vol. 25, No. 4, July – August 2006, pp. 367 – 378.
- [49] Sánchez, J. and Sáenz, M. (2006) Three-Dimensional Virtual Environments for Blind Children. Journal of Cyberpsychology and Behavior, CP&B, Apr 2006, Vol. 9, No. 2, pp. 200-206.
- [50] Sánchez, J. and Sáenz, M. (2007) Usability of Audio-Based Virtual Environments for Users with Visual Disabilities. Virtual Reality and Human Behavior Symposium, LAVAL Virtual 2007, Laval, France , April 18-22, 2007
- [51] Sánchez, J. and Zúñiga, M. (2006). Evaluating the Interaction of Blind Learners with Audio-Based Virtual Environments. Cybersychology & Behavior, Volume 9, Number 6, 2006, pp. 717

- [52] Sánchez, J., Sáenz, M. and Baloian, N. (2007) Mobile Application Model for the Blind. In C. Stephanidis (Ed.): Universal Access in HCI, Part I, HCII 2007, LNCS 4554, pp. 527–536, 2007. © Springer-Verlag Berlin Heidelberg 2007
- [53] Sánchez, J., Tadres, A., Pascual-Leone, A. and Merabet, L. (2009). Blind Children Navigation through Gaming and Associated Brain Plasticity. Proceedings of the IEEE Virtual Rehabilitation 2009 International Conference, June 29-July 2, 2009, Haifa, Israel, pp. 29-36
- [54] Sodnik J., Tomazic S., Grasset R., Duenser A. and Billinghamurst M. (2006). Spatial sound localization in an augmented reality environment. Proceedings of the 18th Australia conference on Computer-Human Interaction: Design: Activities, Artifacts and Environments. Sydney, Australia, ACM: pp. 111-118
- [55] Sony EyePet official E3 Trailer. http://www.youtube.com/watch?v=_ZQpqfzDgOk (Último acceso: 09/05/2010)
- [56] Thomas, B., Close, B., Donoghue, J., Squires, J., Bondi, P., and Piekarski, W. (2002). First Person Indoor/Outdoor Augmented Reality Application: ARQuake. Personal Ubiquitous Comput. 6, 1 (Jan. 2002), 75-86.
- [57] Universal McCann introduce en España la realidad aumentada en un anuncio publicitario <http://www.marketingdirecto.com/actualidad/digital/universal-mccann-introduce-en-espana-la-realidad-aumentada-en-un-anuncio-publicitario/> (Último acceso: 09/05/2010)
- [58] Web-Design: Sieg der Einfachheit - Stuttgarter Zeitung online <http://www.stuttgarter-zeitung.de/stz/page/detail.php/228123> (Último acceso: 09/05/2010)
- [59] Wells A. (2006) Usability: reconciling theory and practice. In Proceedings of the 24th Annual ACM international Conference on Design of Communication (Myrtle Beach, SC, USA, October 18 - 20, SIGDOC '06. ACM, New York, NY, pp. 99-104
- [60] World Health Organization. Up to 45 million blind people globally - and growing. <http://www.who.int/mediacentre/news/releases/2003/pr73/en/> (Último acceso: 09/05/2010)
- [61] Yaagoubi R., Edwards G. and Badard T. (2009) Standards and Spatial Data Infrastructures to help the navigation of blind pedestrian in urban areas. Urban and Regional Data

Management - Krek, Rumor, Zlatanova & Fendel (eds), Taylor & Francis Group, London, pp. 139 – 150

- [62] Yu, W. and Brewster, S. (2002). Multimodal virtual reality versus printed medium in visualization for blind people. In Proceedings of the Fifth international ACM Conference on Assistive Technologies (Edinburgh, Scotland, July 08 - 10, 2002). ASSETS '02. ACM, New York, NY, 57-64.
- [63] Zhang, J., Ong, S. K., and Nee, A. Y. (2009). Design and development of a navigation assistance system for visually impaired individuals. In Proceedings of the 3rd international Convention on Rehabilitation Engineering & Assistive Technology (Singapore, April 22 - 26, 2009). i-CREATe '09. ACM, New York, NY, 1-4.

Anexos

Anexo 1

Pauta Usuario Final

La presente Pauta tiene por objetivo evaluar la usabilidad del software **ARTAB** para personas con discapacidad visual. Esta pauta se divide en 4 secciones

La primera sección se llama “**Antecedentes**”, en ella deben poner su nombre completo, edad y género.

La segunda sección se llama “**Respuestas de 1 a 10**” corresponde a un conjunto de afirmaciones, las cuales deben ser calificadas con un número que vaya desde 1 a 10, donde 1 significa nada de acuerdo y 10 significa muy de acuerdo.

La tercera sección se llama “**Preguntas Abiertas**”, en ella cada pregunta debe ser respondida de la forma más clara y sincera posible.

La cuarta y última sección se llama “**Comentarios**”, en ella puede poner lo que crea sirva para que mejoremos el software ARTAB.

1. Antecedentes

Nombre	
Edad	
Género	

2. Respuestas 1 a 10

Me gusta el software	
El software es entretenido	
El software es desafiante	
El software me hace estar activo	
Volvería a utilizar el software	
Me sentí controlando las situaciones del software	
El software es interactivo	
El software es fácil de utilizar	
El software es motivador	
El software se adapta a mi ritmo	
El software me permitió entender nuevas cosas	
Me gustan los sonidos del software	
Los sonidos del software son claramente identificables	
Los sonidos del software me transmiten información	

3. Preguntas abiertas

¿Qué te gustó del software?	
¿Qué no te gustó el software?	
¿Qué agregarías al software?	
¿Para qué crees que te puede servir el software?	
¿Qué otros usos le darías al software?	
¿Te gustó utilizar la botonera? ¿Por qué?	

4. Comentarios

--

Anexo 2

Usabilidad Icónica: Evaluación TTS

La presente Pauta tiene por objetivo evaluar la usabilidad icónica de los TTS utilizados en el software **ARTAB** para personas con discapacidad visual.

Nombre	
Edad	
Género	

Voz	Velocidad	Frase	Repeticiones	¿Entiende?	¿Agrada?	Observaciones
Francisca (CHILE)	NORMAL (0)	Como toda estrella el Sol posee una forma esférica.				
Diego (ARGENTINA)	NORMAL (0)	El copihue es la flor nacional de Chile.				
Jorge (ESPAÑA)	NORMAL (0)	El fútbol incluye una actividad física muy importante.				
Francisca (CHILE)	15	Como toda estrella el Sol posee una forma esférica.				
Diego (ARGENTINA)	15	El copihue es la flor nacional de Chile.				
Jorge (ESPAÑA)	15	El fútbol incluye una actividad física muy importante.				
Francisca (CHILE)	30	Como toda estrella el Sol posee una forma esférica.				
Diego (ARGENTINA)	30	El copihue es la flor nacional de Chile.				
Jorge (ESPAÑA)	30	El fútbol incluye una actividad física muy importante.				

Anexo 3

Pauta de usabilidad con preguntas abiertas

Nombre	
Edad	
Género	

1) ¿Te fue posible saber tu posición en relación a los objetos?

2) ¿Fue de tu agrado el feedback sonoro? ¿Por qué?

3) ¿Qué otro efecto sonoro asociarías a arriba-abajo?

4) ¿Qué otro efecto sonoro asociarías a izquierda-derecha?

5) ¿Qué otro efecto sonoro asociarías a lejos-cerca?

6) ¿Los textos utilizados fueron claros?

7) ¿La información que te entrega el software, es suficiente para saber qué hacer y lo que sucede?
