



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

VIDEOJUEGO EDUCATIVO PARA EL DESARROLLO DEL PENSAMIENTO GEOMÉTRICO
EN NIÑOS CON DISCAPACIDAD VISUAL

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

EDNAR ALEJANDRO ECHEVERRIA CARRASCO

PROFESOR GUÍA:

JAIME SÁNCHEZ ILABACA

MIEMBROS DE LA COMISIÓN:

JÉRÉMY BARBAY

MARISA ERNST ELIZALDE

SANTIAGO DE CHILE

2020

Resumen

Esta memoria presenta el proceso iterativo de diseño y desarrollo de un videojuego educativo para niños con discapacidad visual, ya sea ceguera parcial o total. El principal requisito de este videojuego, desde el punto de vista diseño de *software*, es que pueda ser utilizado por el público objetivo de una manera sencilla y amigable, sin que tengan que acudir a un experto o especialista que les dé soporte para poder jugarlo.

Este videojuego nace a partir del potencial de las herramientas tecnológicas para dar apoyo a la educación de niños ciegos y frente a la dificultad que existe para enseñar diversos contenidos de la geometría y la matemática a dichas personas. Algunos de estos contenidos son el plano cartesiano, recta numérica, área y perímetro y las transformaciones geométricas básicas.

Enseñar este tipo de contenidos a los niños ciegos suele ser un real desafío ya que muchos de los conceptos geométricos aluden mayormente a aspectos y componentes visuales. Lamentablemente estos aspectos visuales no son claros para las personas ciegas, las que perciben el entorno de un modo completamente diferente al nuestro.

StarAudioGeoxies, el *software* desarrollado para dispositivos móviles durante esta memoria, es un videojuego en donde el jugador debe resolver diferentes problemas geométricos dispuestos sobre un plano cartesiano por medio de la pantalla táctil. El principal enfoque del diseño de este videojuego fue en la usabilidad y en la jugabilidad.

El diseño y desarrollo de este *software* fue acompañado de una constante evaluación con niños ciegos, la que permitió mejorar enormemente la calidad del *software* desarrollado. Además, los resultados de esta misma evaluación y posterior análisis arrojaron que el videojuego cumplió satisfactoriamente los requisitos propuestos. El análisis también arrojó los aspectos de diseño que deben ser considerados y mejorados en futuros trabajos e investigaciones.

Tabla de contenido

| | |
|--|-----------|
| 1. Introducción | 1 |
| 2. Objetivos | 3 |
| 2.1. Objetivo General | 3 |
| 2.2. Objetivos Específicos | 3 |
| 3. Marco teórico | 4 |
| 3.1. Usabilidad | 4 |
| 3.2. Jugabilidad y experiencia de usuario | 6 |
| 3.3. Orientación y movilidad | 8 |
| 3.4. Mapas mentales | 8 |
| 3.5. Educación y matemáticas en niños ciegos | 9 |
| 3.6. Aprendizaje y tecnología | 10 |
| 3.7. Aprendizaje en personas ciegas | 12 |
| 3.8. Aprendizaje con videojuegos | 13 |
| 4. Estado del Arte | 13 |
| 4.1. Educación matemática en niños ciegos | 13 |
| 4.2. Videojuegos para ciegos | 14 |
| 4.3. Software y videojuegos educativos para ciegos | 15 |
| 5. Especificación del problema | 17 |
| 5.1. Problema a nivel de software | 17 |
| 5.2. Problema a nivel de jugabilidad | 18 |
| 5.3. Problema a nivel herramientas educativas | 19 |
| 5.4. Reducción del problema | 19 |
| 6. Metodología | 19 |
| 6.1. Diseño y desarrollo del prototipo | 20 |
| 6.2. Evaluación de usabilidad y jugabilidad | 22 |
| 6.3. Análisis y conclusiones | 23 |
| 6.4. Etapas metodológicas | 23 |
| 7. Diseño y desarrollo del videojuego | 24 |
| 7.1. Aspectos técnicos | 24 |
| 7.1.1. Unity3D | 25 |
| 7.2. Contexto del videojuego | 26 |
| 7.3. Análisis de usabilidad de softwares para ciegos | 26 |
| 7.3.1. Aspectos positivos | 28 |
| 7.3.2. Aspectos negativos | 29 |
| 7.4. Fortalezas del videojuego desarrollado | 32 |
| 7.4.1. Diseño enfocado en la jugabilidad | 32 |
| 7.4.2. Diseño enfocado en la usabilidad | 34 |
| 7.4.3. Plano cartesiano y variedad de problemas | 35 |
| 7.4.4. Repetición y aleatoriedad | 36 |

| | |
|--|-----------|
| 7.4.5. Registro y seguimiento | 36 |
| 7.5. Interfaz y usabilidad del videojuego | 37 |
| 7.5.1. Vistas y menús | 39 |
| 7.6. Sistema de niveles | 45 |
| 7.7. Módulos geométricos | 46 |
| 7.7.1. Módulo 1: Ataque con rayo láser | 48 |
| 7.7.2. Módulo 2: Ataque con misil dirigido | 48 |
| 7.7.3. Módulo 3: Esquivar asteroides | 49 |
| 7.7.4. Módulo 4: Espiar planetas enemigos | 50 |
| 7.7.5. Módulo 5: Extracción energética | 50 |
| 7.7.6. Módulo 6: Esquivar tormentas solares | 51 |
| 7.8. Sistema de registro y seguimiento | 52 |
| 7.9. Historia del videojuego | 53 |
| 7.9.1. Narración de la historia | 54 |
| 7.10. Unity 3D: recursos y arquitectura | 56 |
| 7.11. Arquitectura y diseño de software | 61 |
| 7.11.1. Speech dinámico | 61 |
| 7.11.2. Interfaz para ciegos | 62 |
| 7.11.3. Módulos | 63 |
| 8. Evaluación de usabilidad y jugabilidad | 65 |
| 8.1. Muestra | 65 |
| 8.2. Procedimiento | 66 |
| 8.3. Resultados | 66 |
| 8.4. Análisis | 69 |
| 8.4.1. Análisis de resultados de usabilidad | 69 |
| 8.4.2. Análisis de resultados de experiencia de juego | 74 |
| 8.5. Correcciones de interfaz de usuario y jugabilidad | 75 |
| 8.6. Conclusiones | 77 |
| 8.6.1. Resumen | 77 |
| 8.6.2. Objetivos alcanzados y no alcanzados | 77 |
| 8.6.3. Aspectos a mejorar y trabajo futuro | 78 |
| Bibliografía | 80 |
| Anexo | 85 |

1. Introducción

En la actualidad se está constantemente trabajando en la inclusión del ser humano con *discapacidades* de todo tipo, ayudando a que estos puedan llevar a cabo una vida más equitativa, justa y digna en comparación al resto de las personas, sobretodo en el ámbito laboral. Una de estas tantas formas de inclusión es el trabajo e investigación con personas ciegas para facilitar la forma en cómo ellos interactúan con el mundo ayudándose de las herramientas tecnológicas de nuestro tiempo.

Según la Organización Mundial de la Salud (OMS [29]), una persona ciega es aquella que tiene una visión menor al 10% en su mejor ojo. Esto significa que dentro de las personas ciegas podemos encontrar también personas con resto visual, los que pueden distinguir colores, figuras, y con la corrección adecuada, distinguir incluso letras y palabras. Es importante tener en cuenta que al hablar de ciegos normalmente nos referiremos a personas con y sin resto visual.

Una de las más grandes problemáticas que suelen poseer las personas ciegas es referente a su escasa comprensión de la forma y el espacio que les rodea, lo que se traduce en problemas de orientación y movilidad (O&M) en su vida cotidiana [49], así como también en problemas del razonamiento y pensamiento geométrico.

Esta problemática suele ser aún mayor cuando nacieron con dicha discapacidad visual, ya que nunca recibieron ese estímulo visual y físico que todos los videntes experimentan desde desde el nacimiento. Estos estímulos apoyan el desarrollo de ciertas habilidades que ayudan a generar distintas rutas de movimiento o diferenciar lo que está lejos de lo que está cerca [20], por ejemplo.

Esta desventaja del entendimiento del espacio físico y su geometría, junto a la ceguera misma, ocasionan que las personas ciegas pierdan gran parte de su independencia en diferentes ámbitos de la cotidianidad. Es así como también se ve afectada la comodidad, autonomía y eficiencia de la interacción con ciertas tecnologías, incluyendo piezas de *software* y su interfaz intermediaria.

Los *softwares* para ciegos deben funcionar principalmente en base a lo que estas personas pueden percibir: estímulos físicos y sonidos principalmente [21], correspondientes al háptica y a la acústica respectivamente. Lo anterior se traduce en que, en una pieza de *software*, debe primar el uso de estímulos sonoros y táctiles. Estos estímulos suelen darse en forma de efectos de sonido, música, voz y vibración.

Desafortunadamente las tecnologías actuales, o al menos las más masivas, no suelen poseer mayor háptica que la vibración. Es por esto que el sonido ha pasado a ser la

fuente principal para transmitir información al usuario ciego [21], sin mencionar su enorme eficacia gracias al uso de sonidos de voz.

El sonido posee una variedad de matices que pueden ser tomadas en consideración al momento de diseñar una interfaz o videojuego, como el volumen, tono, timbre, calidad, posición, etc, propiedades que han sido usadas y estudiadas con anterioridad [48]. Cada sonido puede decir algo diferente, y es fundamental que las personas ciegas se enteren de la información contenida en un *software* para que puedan usarlo de forma satisfactoria alcanzando el objetivo de este.

Dentro del háptica, los estímulos físicos que el humano puede sentir son: vibración, temperatura, dureza, humedad, rugosidad, texturas en general, entre otros. Lamentablemente, no todo esto puede ser reproducido por la tecnología actual, a excepción de las tecnologías más modernas como las pantallas braile [5], las que no están al alcance del público general. De momento, el mejor esfuerzo que se puede hacer es enfocarse en la vibración y las diversas formas de cómo esta se puede emplear en un *software*.

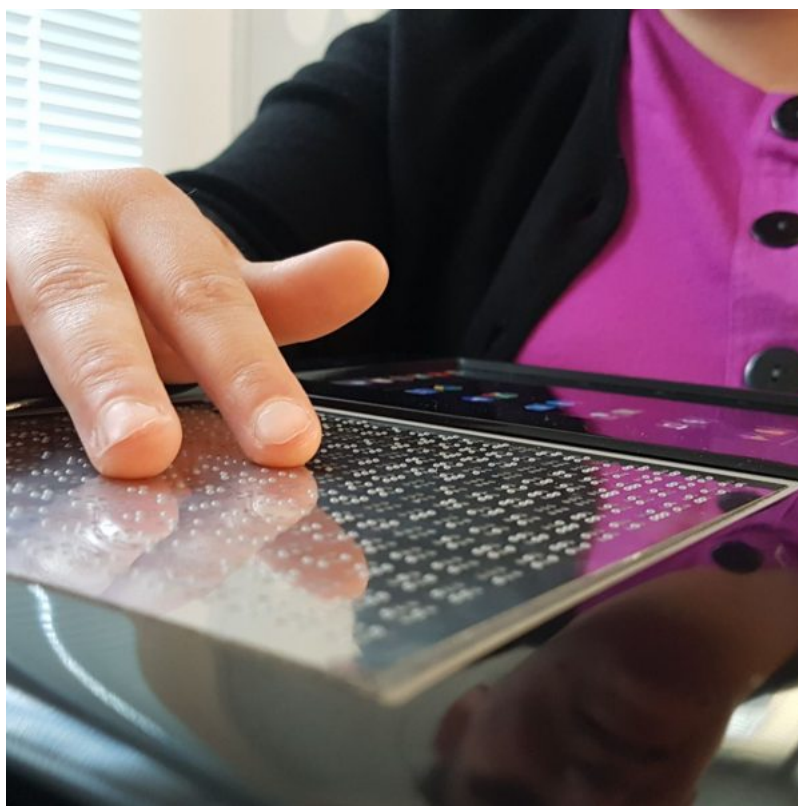


Figura 1: Tablet táctil Blitab [5], ideal para usuarios ciegos pero lejos del alcance del público general.

Frente a la problemática de las personas ciegas ante la disminución de la independencia, de la eficacia y del entendimiento en el uso de las tecnologías, en adición a la falta

de apoyo al pensamiento geométrico, a la comprensión del espacio y de la forma, se propone el videojuego educativo StarAudioGeoxies. Este videojuego hace uso de diferentes desafíos geométricos dentro de un plano cartesiano a través de una pantalla táctil. Estos desafíos geométricos están diseñados a partir en la malla curricular de matemática de la enseñanza básica, por lo que el videojuego se enfoca en niñas y niños de entre 5to y 8vo año básico.

El objetivo del videojuego educativo StarAudioGeoxies es que los usuarios con problemas visuales puedan utilizarlo como una herramienta educativa guiados por un profesor, o como un pasatiempo jugándolo de forma autónoma tal como harían normalmente con otros videojuegos.

Por todo lo anterior, el diseño y desarrollo de la aplicación fue dirigido por la usabilidad, la interacción humano-computador (HCI) y la experiencia de usuario, consiguiendo que el videojuego pueda ser utilizado eficientemente y de manera autónoma, y manteniendo el foco y motivación de los jugadores tal como lo haría otro videojuego.

2. Objetivos

2.1. Objetivo General

El objetivo de esta memoria fue el desarrollo de un videojuego educativo para dispositivos móviles, enfocados en usuarios ciegos de entre 11 y 14 años. Este videojuego fue diseñado para poder ser jugado de forma autónoma por parte de los usuarios, es decir, sin un especialista que guíe o dirija la interacción del usuario con la interfaz, y que a su vez, pueda ser utilizado como herramienta educativa y de seguimiento del usuario ciego por parte diferentes especialistas o educadores de matemática.

2.2. Objetivos Específicos

1. Diseñar y desarrollar un videojuego con desafíos geométricos en un plano cartesiano para dispositivos móviles Android. Este videojuego debe poder ser jugado por personas con ceguera parcial o completa.
2. El videojuego debe ser lo suficientemente auto-explicativo para permitir al jugador utilizarlo de manera autónoma, sin requerir de un tercero para su uso y entendimiento.

3. Marco teórico

3.1. Usabilidad

La usabilidad hace referencia a qué tan *usable* es una herramienta o tecnología. Este concepto de “usable” contiene diferentes variables que indican ciertas cualidades de la herramienta o tecnología en cuestión, cualidades que hacen que la forma de uso sea mucho más sencillo y agradable [26].

La usabilidad, en conjunto con otras propiedades de una tecnología, permiten determinar si una tecnología puede o no ser utilizada para los objetivos para lo cual fue diseñada y creada, usabilidad que depende directamente del usuario final para quien fue diseñada. Un ejemplo sencillo es la ropa para adultos versus la ropa de niño: ambas fueron diseñadas para vestir, con todo lo que dicha acción conlleva, pero los adultos no pueden utilizar la ropa de niño (debido a su tamaño), y los niños que usen la ropa de adulto lo harán posiblemente de una forma incómoda y poco eficaz.

Suelen nombrarse 5 aspectos que definen la usabilidad de una tecnología, aspectos que permiten evaluar la calidad de uso que tiene dicha tecnología o herramienta [26]:

1. Facilidad de aprendizaje:

Este aspecto, tal como indica, alude a la facilidad con la que el usuario aprende a utilizar la tecnología evaluada antes de tener conocimiento alguno sobre ella, es decir, durante su primera interacción con dicha tecnología.

Es importante tener en cuenta que este aspecto se ve directamente afectado por la experiencia y conocimiento previo que posee el usuario que utiliza la tecnología por primera vez. Aquí también se debe considerar la intuición del usuario, que corresponde a un conocimiento aplicado automáticamente y de manera inconsciente en base a experiencias pasadas [25].

Dada la complejidad de algunas tecnologías, en ocasiones estas requieren de un periodo de apresto en donde un especialista realiza una capacitación de la tecnología y su interfaz al usuario. Parte del trabajo de este aspecto de usabilidad es justamente prescindir de dicho especialista, de forma que el usuario pueda utilizar la tecnología con nada más ni menos que él mismo.

2. Eficiencia

Dentro de la usabilidad, la eficiencia de una tecnología señala qué tan rápido el usuario puede utilizarla una vez tiene el conocimiento necesario hacerlo. La idea no es que el usuario utilice la tecnología de la forma más rápida que pueda, sino más bien utilizarla a una velocidad con la que se encuentre cómodo, en donde no tenga que esperar más de lo necesario ni apurarse innecesariamente.

3. Facilidad para recordar

Tecnologías que son más sencillas a nivel de interfaz, son más fáciles de recordar por la pequeña cantidad de componentes y fácil uso. Este hecho es de suma importancia para que el usuario no tenga que volver a aprender a utilizar la tecnología una vez esté un tiempo alejado de ella.

Normalmente un usuario nunca podrá recordar del todo el funcionamiento e interfaz de una tecnología, por lo que este aspecto debe ir de la mano con el aspecto referente a la facilidad de aprendizaje, de tal manera que si la memoria no es lo suficientemente aguda, el usuario pueda continuar utilizando la tecnología gracias a que es sencilla de aprender a usar.

4. Nivel de error

El nivel de error de una tecnología señala con qué frecuencia los usuarios cometen errores al usar dicha tecnología, qué tan serios son estos errores y con qué facilidad el usuario puede superarlos. No todos los usuarios tienen la misma experiencia ni conocimiento, por lo que algunos cometerán más errores que otros y es imposible evitar que alguno de ellos cometa algún error. Este aspecto de usabilidad no busca eliminar los errores, más bien busca disminuir su frecuencia y su impacto negativo en el uso de la tecnología.

5. Satisfacción

La satisfacción es un aspecto que indica cuán placentera o amigable es la tecnología. Este aspecto, al igual que algunos anteriores, depende de las experiencias previas del usuario y del conocimiento que este tiene.

De forma adicional, la satisfacción de un usuario respecto a una tecnología evoluciona a medida que este la utiliza, generando una mayor satisfacción a medida que se tiene un mayor conocimiento sobre el uso de la tecnología o bien disminuyéndola si esta, en algún punto, deja de ser amigable.

Estos 5 aspectos no solo nos permiten evaluar la usabilidad de una tecnología, sino que además pueden ser utilizados para el diseño de usabilidad en piezas de *software* e interfaz de usuario (UI) de una aplicación por medio de la interacción usuario-computador (HCI). Si una aplicación no es usable, esta no se utiliza, muy simple. Por ejemplo, si un sitio web no es usable o si las personas no entienden el servicio que se ofrece o lo que pueden realizar en el sitio web, se van. La usabilidad es indispensable para la supervivencia de una tecnología.

Para medir la usabilidad de una tecnología se tienen 3 requisitos [26]:

1. Usuarios representativos:

Se deben conseguir usuarios con un perfil similar al perfil del público objetivo al que estamos apuntando. Los diseñadores no pueden evaluar sus propias aplicaciones, ya que estos no representan al usuario final.

2. Tareas representativas:

Se debe solicitar a los usuarios representativos que realicen ciertas tareas definidas y representativas de la aplicación.

3. Observar:

El usuario debe ser observado mientras realiza las tareas representativas, identificando las componentes de la aplicación en dónde interactúan de forma satisfactoria y en dónde tienen dificultades para hacerlo.

Al momento de realizar ingeniería de *software*, es indispensable tomar en cuenta estos aspectos en el diseño de la interfaz, de otro modo la aplicación no será utilizada y el objetivo principal de la aplicación no será alcanzado.

3.2. Jugabilidad y experiencia de usuario

Uno de los desafíos más grandes de todas las formas de aprendizaje es mantener a los estudiantes lo suficientemente motivados para apegarse al proceso de aprendizaje sea cual sea la finalidad [31].

La motivación es el razón que las personas tienen para hacer algo, o aprender algo nuevo, ya sea netamente por gusto, por un objetivo o por una recompensa en específico. En el caso de los videojuegos, la motivación es el motor que impulsa a un persona a jugarlo, y más aún, es la encargada de mantener su interés en el videojuego para que éste último continúe jugando.

Para entender mejor el significado de la motivación en la tecnología es necesario mencionar el concepto de *experiencia de usuario* (UX). La experiencia de usuario es el conjunto de decisiones que llevan a un usuario hacia un resultado u objetivo a través de una herramienta interactiva, tanto como la productividad y satisfacción del proceso para alcanzar dicho resultado [36].

En otras palabras, la experiencia de usuario puede describirse como la capacidad para generar motivación al usuario durante la interacción de este con la tecnología hasta que alcanza cierto objetivo. Al entrar en el campo de los videojuegos, esta experiencia de usuario se transforma en experiencia de juego o *jugabilidad*, también conocida como *gameplay*.

La jugabilidad corresponde al conjunto de actividades y estrategias utilizadas para mantener a los jugadores enfocados y motivados a completar cada nivel de un videojuego [31]. Esta propiedad de un videojuego es lo que origina la motivación del individuo solo mediante el gusto de conectar con las ideas e información expuestas [31]. Dicha motivación suele verse afectada por factores como el miedo, el esfuerzo, la frustración, etc.

La jugabilidad crea motivación a partir de mantener el foco del usuario en el videojuego en cada instante que transcurre, lo que hace que cada segundo del videojuego sea un constante desafío para los diseñadores [31].

La entretención es la principal manera de conseguir que el jugador se mantenga motivado utilizando el videojuego [31] [32], esto es debido a que la entretención hace que volvamos a repetir la actividad una y otra vez de modo que obtengamos dicha satisfacción de manera reiterada [31]. Es por esto que uno de los focos principales a tomar en cuenta durante el diseño de la experiencia de juego es la entretención de las componentes que este posee. Este enfoque de diseño en la entretención no solo es útil para disfrutar un videojuego, sino que además es una herramienta esencial para la educación mediante el uso de estos últimos [32] [31].

El modelo Gameplay Enjoyment Model (GEM) [32] propone que existen 6 tipos de características de los cuales surge entretenimiento con los que el jugador puede conectar para conseguir disfrutar un videojuego en base a la entretención:

- **Desafío:**
Refiere al tipo de entretención que nace a partir de la superación de obstáculos desafiantes y de la dificultad para vencerlos y dominarlos. Para que un videojuego sea entretenido, el nivel de esta característica debe ser el adecuado. Un bajo nivel de desafío ocasionará aburrimiento, mientras que uno demasiado alto generará frustración [11] [9].
- **Compañerismo:**
El compañerismo hace referencia a la entretención que se obtiene por el videojuego mediante la existencia de más jugadores y la interacción colaborativa con estos.
- **Competición:**
La competición, al igual que el compañerismo, alude hacia la entretención mediante la interacción entre jugadores, la diferencia reside en que ya no hay una colaboración sino más bien una competencia con los demás jugadores, permitiendo comparar habilidades entre jugadores otorgando comúnmente reconocimiento público a los más talentosos.
La competición entre jugadores puede reducir la motivación en ciertos contextos, pero también se ha demostrado que puede impactarla positivamente en otros. Es importante detectar qué usuarios tienen afinidad con la competición para evitar generar una sensación desagradable a los jugadores.
- **Exploración:**
La exploración refiere a la entretención que trae el buscar entidades ocultas, descubrir eventos inesperados, explorar zonas desconocidas, experimentar con estrategias diferentes de juego y explorar el funcionamiento interno del videojuego.
- **Fantasía:**
Esta característica trae entretención mediante el uso de mundos fantásticos, criaturas imaginarias y personajes ficticios, dando vida a entidades y habilidades que no

están presentes en nuestro mundo real.

Se ha demostrado que los niños prefieren videojuegos educativos que componentes de fantasía frente a los que no. No solo eso, la fantasía puede ser un motivador de gran valor al momento de generar aprendizaje a través de un videojuego.

- **Fidelidad:**

La fidelidad alude a la entretención obtenida a partir de utilizar gráficos realistas, efectos de sonidos, animaciones y gráficos 3D.

3.3. Orientación y movilidad

Las personas ciegas poseen una capacidad reducida de orientación y movilidad en comparación al resto de las personas, lo que dificulta diversos de sus quehaceres cotidianos influyendo también en su autonomía y libertad sobre el entorno físico.

La *orientación* corresponde a la capacidad de conocer la posición propia en el espacio respecto a ciertos puntos de referencia, como lo puede ser el punto de llegada o destino o elementos del entorno [13].

Esta capacidad de orientación se origina luego de un complejo proceso cognitivo que utiliza la información sensorial percibida por nuestro cuerpo.

La *movilidad* refiere a la posibilidad de moverse libremente y de forma autónoma a través de diferentes entornos, públicos o privados, abiertos o cerrados [13].

Una persona con estas capacidades limitadas, ya sea por razones físicas y/o mentales, no podrá movilizarse de forma autónoma apropiadamente ni orientarse de la mejor manera. Esto podría mejorarse enfocándose en el uso de distintas tecnologías y educación cognitiva para reducir estas limitaciones.

Aquí tenemos como ejemplo personas con sus extremidades paralizadas, ciegos, sordos o cualquier persona con dificultades de movimiento u orientación causa de alguna condición mental.

3.4. Mapas mentales

El mapa mental es una visualización o esquema mental que asocia diferentes entidades entre sí con diferentes significados. Son utilizados por las personas y animales para

entender el entorno que les rodea.

Este es generado mediante el uso de la memoria de experiencias pasadas y las sensaciones inmediatas de nuestro cuerpo ante estímulos del mundo físico. La gran mayoría de la información requerida para generar este mapa mental es visual [22].

La generación de mapas mentales del entorno es sin duda uno de los pilares fundamentales de la orientación y movilidad de una persona [20]. Estos mapas mentales son lo que le dan significado a la orientación de la persona, permitiendo que esta escoja uno de los distintos trayectos generados en su mente antes de comenzar a moverse.

En pocas palabras, si queremos desarrollar la capacidad de orientación y movilidad de las personas con discapacidad visual, debemos enfocarnos en que ellos consigan mejorar su entendimiento del entorno, vale decir, que consigan generar y comprender los mapas mentales a cabalidad.

3.5. Educación y matemáticas en niños ciegos

Cuando hablamos de personas ciegas, hablamos también de un grupo vasto y heterogéneo de personas con esta condición, algunos nacidos completamente ciegos y otros afectados en algún punto de sus vidas, por algún accidente o enfermedad [1].

Por lo anterior, las personas tienen diferentes experiencias visuales del mundo, y los que tengan una mayor experiencia tendrán también una mayor habilidad para comprender conceptos físicos y geométricos [55]. No es lo mismo enseñarle a un ciego de nacimiento que a un ciego que en algún momento fue vidente.

El desarrollo de distintas habilidades y áreas en niños depende directamente del sentido espacial que posean [1], es por ello que no puede dejarse la matemática ni la geometría de lado al enseñar a niñas y niños ciegos.

Aprender matemáticas suele ser de las tareas más complejas para los niños ciegos, pero incluso ellos pueden aprender matemáticas y geometría a la par con sus compañeros videntes cuando se les enseña de la forma correcta [1], apoyándose principalmente de herramientas y recursos hápticos [28].

3.6. Aprendizaje y tecnología

Cuando se menciona que la tecnología ayuda al aprendizaje hay dos posibles situaciones: la tecnología innova a nivel de los procesos cognitivos del individuo o bien referente a las metodologías y estrategias utilizadas para alcanzar dichos procesos cognitivos [33]. Ambos casos sin lugar a dudas aportan al desarrollo cognitivo, ya sea de manera directa o indirecta, por lo que este tipo de tecnología debe ser considerada para fomentar el aprendizaje.

Lo que algunas personas piensan al escuchar tecnología es de inmediato en un computador o en un teléfono inteligente, pero la realidad es que la tecnología está en todo lo que nos rodea y evoluciona día a día, desde nuestra ropa hasta los computadores.

De lo anterior podemos concluir directamente que no toda la tecnología nos sirve para aprender, al menos no si no se utiliza correctamente, es por ello que debemos ser cuidadosos con cuál utilizar al momento de querer enseñar algo mediante su uso.

Dentro de los tipos de tecnología existen las tecnologías de información y comunicación (TIC), las que nos permiten captar, procesar, editar, enviar y generar información, y todo lo necesario que nos permite mantener contacto unos con otros [33].

Dentro de este tipo de tecnología está la televisión y el teléfono, pero estos ejemplos de TICs no son nada en comparación a las tecnologías que existen hoy en día: evoluciones de las TICs anteriores. Estas últimas nos permiten captar información que las tecnologías nos muestran de una infinidad de maneras, con una infinidad de interfaces e interacciones HCI.

Estas tecnologías poco a poco han tomado parte en nuestras vidas, llegando a ser instrumentos cotidianos en nuestro día a día. Estas han estado siendo introducidas poco a poco a las aulas de clases, puesto que las personas consiguieron ver lo beneficioso que son para las diferentes estrategias educativas [33].

El uso de tecnologías ha resultado ser de gran ayuda en el ámbito educativo, puesto que los estudiantes trabajan con más motivación en la escuela al utilizar una herramienta con la cual están familiarizados y acostumbrados, como lo es el computador o el teléfono inteligente [33].

Lo anterior nos indica que es de suma importancia utilizar tecnología con la que los estudiantes se sientan cómodos, de tal manera que dicha tecnología no sea contraproducente para el aprendizaje. De esta premisa nace el concepto de tecnología invisible [44].

Para conseguir un aprendizaje óptimo en el estudiante, el uso de tecnología invisible se traduce directamente en el uso de tecnologías cotidianas. Dichas tecnologías cotidianas se definen por nuestra sociedad y cultura, y también dependen de la edad de cada persona.

No es lo mismo si apuntamos al aprendizaje en estudiante de enseñanza básica que al aprendizaje de personas de 50 años, las tecnologías invisibles en estos usuarios puede potencialmente diferir. Por ejemplo, la persona de 50 años puede estar acostumbrada al uso cotidiano del computador debido a su trabajo, mientras el niño de enseñanza básica solo conoce los dispositivos móviles y las consolas de videojuegos.

El uso inadecuado y desmedido de la tecnología en el aprendizaje puede generar efectos contrarios a los deseados, consiguiendo generar frustración en los usuarios si no se realiza una correcta introducción a dichas tecnologías [11] [9], sin mencionar los efectos negativos del uso desmedido de la tecnología en las relaciones sociales y el detrimento de otras actividades educativas.

La tecnología es y será parte de nuestras vidas, lo queramos o no, aprenderemos a vivir con ella de una u otra manera. El rol de la tecnología en el aprendizaje no es indispensable, las personas han podido sobrevivir sin ella sin ningún problema, a pesar de ello les ha facilitado enormemente la calidad de vida, por lo que negar su importancia en el aprendizaje no es el camino a seguir.

Para generar aprendizaje, el *software* educativo respectivo de una tecnología puede pertenecer a cuatro categorías diferentes [33]: tutorial, cuticular, lúdica y de tratamiento de información.

Una tecnología tutorial provee al usuario información sobre una temática o contenido junto a preguntas simples con *feedback* inmediato respecto a la correctitud de una acción o respuesta, guiando al usuario para que este pueda utilizar el *software* de manera más autónoma y eficiente. La categoría cuticular en cambio, trabaja no solo los contenidos, sino que también procedimientos y metodologías relacionadas con el contenido de la manera más similar posible al trabajo en el aula.

La categoría más interesante para esta memoria es la lúdica, esta busca el aprendizaje haciendo uso de la motivación del usuario para utilizar el *software*. Aquí entran videojuegos y otro tipo de aplicaciones que son más llamativas al usuario, en donde comienza a resonar y tomar importancia el concepto de *jugabilidad*.

Las tecnologías de tratamiento de información, a diferencia de las otras, son herramientas para procesar información y elaborar todo tipo de documentos necesarios para educar y aprender, por lo que su rol en el aprendizaje es más bien indirecto.

Retomando la categoría lúdica, una de las palabras claves para poder generar aprendizaje mediante este tipo de tecnologías es la motivación, tal como se comentó en capítulos anteriores. El aprendizaje se da en la medida en que el individuo se siente involucrado, y en este sentido es que el ambiente mediado por las tecnologías provocan procesos de aprendizaje [2]. No es la tecnología la causante del aprendizaje, sino más bien el uso práctico y didáctico de esta.

Las TICs pueden ser utilizadas como herramientas de aprendizaje en diferentes ámbitos, pero esto no las hace automáticamente aptas para ser utilizadas por personas ciegas, las TIFLO [58] (o tiflotecnología), en cambio, son tecnologías que están diseñadas pensando en usuarios ciegos o con baja visión. Estas últimas corresponden a tecnologías de apoyo con instrumentos auxiliares o adaptaciones que posibilitan la correcta utilización de tecnologías a usuarios ciegos o con discapacidad visual. Por supuesto, ciertas TIFLOs también corresponden a TICs, por lo que pueden ser utilizadas para el aprendizaje.

3.7. Aprendizaje en personas ciegas

Una de las grandes ventajas de las TICs para el aprendizaje en las personas ciegas, específicamente estudiantes, fue su globalización. El hecho de que los estudiantes tengan computadoras y dispositivos móviles en sus hogares hace que el uso de éstas en el aula sea mucho más eficiente.

Esta globalización trae consigo la inclusión de las personas ciegas a la sociedad, ya que estos dispositivos pueden ser utilizados tanto por personas videntes como ciegas, permitiéndoles trabajar relativamente a la par, al momento de utilizar estos dispositivos en el aula.

Las tecnologías TICs para ciegos se caracterizan por utilizar principalmente canales sensoriales complementarios a la vista: el tacto y la audición, y se ha demostrado que las tecnologías que utilizan sonidos, tanto icónicos como verbales, pueden fomentar el aprendizaje y cognición en niños ciegos [38, 21].

Las interfaces hápticas pueden ser un gran complemento para las tecnologías basadas en audio. Mediante el uso de tecnologías hápticas el usuario puede identificar formas, objetos, posiciones y relaciones físicas, entre otros datos relevantes [43, 27, 59], por lo que en conjunto con el audio se pueden construir herramientas que impacten significativamente la cognición en los usuarios que utilicen estas tecnologías educativas.

3.8. Aprendizaje con videojuegos

Actualmente, los videojuegos son parte de la cotidianidad de muchos niños y jóvenes dentro de sus pasatiempos. Afortunadamente, los videojuegos no solamente son útiles para entretenerse, sino que también sirven como herramientas educativas [49, 12].

Diferentes tipos de videojuegos pueden impactar el desarrollo de diversas habilidades en las personas con discapacidad visual, por lo que estos pueden ser utilizados en una gran cantidad de áreas y especialidades que un humano puede desarrollar.

El gran potencial de la enseñanza a través de un videojuego, es que este también puede ayudar a transferir las habilidades cognitivas desarrolladas al entorno cotidiano [49]. Pero antes de esto es de suma importancia asegurar que estos videojuegos efectivamente fomentan el desarrollo cognitivo, para lo cual existen diversos tipos de herramientas.

Aludiendo a la importancia de las interfaces en el aprendizaje de la tecnología educativa, los videojuegos pueden exponerse al público en diversas interfaces de usuario; dentro de las más conocidas están el *joystick*, el *mouse*, el teclado, el *touchpad* y la pantalla táctil. Cada interfaz suele tener un público objetivo, generalmente debido a la accesibilidad y popularidad. Además, ciertas interfaces son más favorables que otras para ciertos tipos de videojuegos. Es por ello que se debe ser muy cauteloso al momento de escoger la interfaz y usabilidad de un videojuego.

Dentro de todas estas opciones de interfaz, destacamos la pantalla táctil, debido a su gran capacidad háptica, ya que permite generar videojuegos para usuarios con discapacidad visual con una gran versatilidad en la interacción táctil [6], y a su propiedad de ser una tecnología invisible en la sociedad actual.

4. Estado del Arte

4.1. Educación matemática en niños ciegos

Como se señaló en puntos anteriores, es de suma importancia exponerle los conceptos al niño ciego de una manera que él pueda entenderlos. Es por ello que en el aula de clases se utilizan diferentes herramientas didácticas basadas en háptica especializadas para estudiantes con discapacidad visual [28].

Dentro de estas herramientas para enseñar matemática y geometría se encuentra el geoplano, figuras geométricas tridimensionales/planas, el ábaco japonés, la calculadora parlante, reglas con relieve y el goniómetro para ciegos, entre varios otros [34].

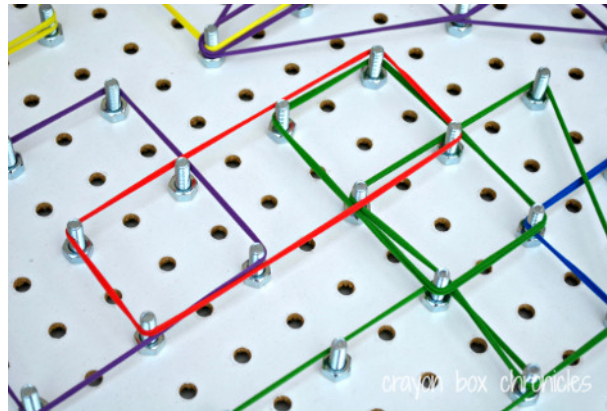


Figura 2: Geoplano para la educación de estudiantes ciegos.

4.2. Videojuegos para ciegos

El mundo de los videojuegos es bastante amplio, posee una gran variedad de géneros y por sobre todo existe una gran variedad de interfaces y modos de jugar. Una de estas tantas formas del videojuego es aquella que no contiene componente gráfica o que bien posee un rol secundario, enfocándose principalmente en el uso del sonido [3]. Estos últimos calzan perfectamente con el perfil del usuario con discapacidad visual.

Un ejemplo es el videojuego *A blind legend* [14]. Este videojuego de la plataforma Android funciona completamente por audio, reconociendo diferentes gestos sobre la pantalla y explicando cada uno de ellos en cada parte del videojuego mediante un *software* narrador, permitiendo que el jugador sepa en todo momento sus opciones de interacción y su objetivo.



Figura 3: Ejemplo de gestos de pantalla en *A blind legend* [14].

Un videojuego que se basa en audio no necesariamente fue realizado para personas ciegas, el no utilizar gráficos es simplemente una modalidad más en el mundo de los videojuegos. De la misma manera, este tipo de videojuegos no tiene porqué ser educativo.

4.3. Software y videojuegos educativos para ciegos

Existe una gran variedad de *software* enfocado en ayudar personas con discapacidad visual para que enfrenten al mundo de una mejor manera. Algunos de ellos funcionan con mucho éxito, como lo son los lectores de pantalla en sistemas operativos, tales como Jaws [37] para Windows, Orca [16] para Linux o Google TalkBack [18] para Android.

También existe *software* enfocado a la educación y al desarrollo cognitivo de las personas ciegas. Un breve listado de ellos se muestra a continuación:

- AbES (Audio-based Environment Simulator) [7]: Simulador de ambientes basado en audio que puede ser navegado a través del teclado.
- mAbES (mobile Audio-based Environment Simulator) [7]: Versión móvil de AbES, con *feedback* háptico e interfaz *touch*.

- HAGA (Haptic Audio Game Application) [24]: Simulador de ambientes para computador de escritorio similar a AbES. Utiliza *joystick* y/o teclado.
- AHM (Audio Haptic Maze) [43]: Laberinto virtual basado en háptica, sonido e imágenes. Utiliza la interfaz *Novint Falcon* [56].
- Wolfpack Haptic Virtual Environment [35]: Simulador de formas geométricas 3D utilizando *feedback* háptico con *Novint Falcon*.
- AudioDoom [40, 4]: Entorno 3D auditivo e interactivo basado en *hiperhistorias* [4]. Utiliza el teclado como interfaz.
- AudioChile [42, 45]: Videojuego de género RPG para aprender sobre la cultura y geografía de Chile en un entorno de simulación 3D.
- AudioMath [39]: Entorno virtual basado en audio para reforzar la memoria de corto plazo y el aprendizaje de la matemática.
- La granja de Theo y Seth [41]: Videojuego para reforzar la teoría de conjuntos a través de estímulos auditivos. Utiliza el teclado.
- AudioBattleship [48]: Videojuego similar a Combate Naval utilizando un plano con coordenadas para atacar. Se utiliza mediante teclado.
- GeoAudioWizard [30]: Videojuego de género RPG para reforzar la orientación, movilidad y las transformaciones geométricas mediante la navegación en un entorno virtual 3D y desafíos geométricos basados en vibración, *feedback* de voz y efectos de sonido. Se utiliza mediante un *joystick*.
- AudioMetro [53]: *Software* para el desarrollo de orientación y movilidad en el uso del Metro de Santiago. Se utiliza con teclado.
- AudiodMC [47]: Herramienta auditiva para la generación de mapas conceptuales que posteriormente pueden ser analizados y discutidos. Uso mediante teclado.
- Viaje a Anatomía [51]: Videojuego interactivo basado en audio para enseñar 4 diferentes sistemas del cuerpo humano: nervioso, digestivo, cardiovascular y respiratorio. Uso mediante teclado.
- AudioMemorice [46]: Herramienta para potenciar el uso de la memoria de corto y largo plazo, la que puede ser utilizada para apoyar diversas áreas del currículo escolar. Uso mediante teclado.
- AudioLink [50]: Videojuego interactivo estilo RPG en donde se puede navegar un entorno virtual 2D. Está enfocado en la resolución de problemas. Uso mediante teclado.
- AudioVida [45]: Videojuego para la resolución de problemas en un laberinto virtual. Uso mediante teclado.
- AudioMUD [52]: Este videojuego define un ambiente virtual multi-jugador que permite la colaboración entre diferentes usuarios. El ambiente, navegación e interacción con el videojuego se diseñó a partir de texto hablado
- Mova3D [54]: *Software* para potenciar las habilidades de orientación de movilidad mediante la navegación de espacios virtuales. Uso mediante teclado y una alfombra háptica.
- Audio Space Invaders [23]: Videojuego basado en un entorno virtual 3D que mezcla el uso del háptica, la audición y la visión, permitiendo generar la misma experiencia

de juego tanto en jugadores videntes como ciegos. Se utiliza mediante un computador y un *joystick* con vibración.

- AudioGeometria: Videojuego estilo RPG en un entorno virtual 3D enfocado en la resolución de problemas geométricos. Uso mediante pantalla táctil en tablet.

5. Especificación del problema

Analizando el estado del arte salen a luz principalmente 2 problemas relacionados con diseño de *software* que requieren ser solucionados para poder alcanzar los objetivos expuestos en esta memoria. Estos problemas son a nivel de usabilidad y a nivel de jugabilidad.

Adicionalmente, también destacamos el problema de desarrollo del pensamiento geométrico presente en las personas ciegas. Estos tres puntos se discuten a continuación.

5.1. Problema a nivel de software

Se ha demostrado que gran parte de los juegos expuestos anteriormente cumplen su objetivo educativo desarrollando la cognición de los usuarios ciegos que los utilizan [49, 35], pero parte de ellos no poseen un diseño de usabilidad lo suficientemente robusto como para conseguir un uso eficiente y autónomo.

Desafortunadamente, el desarrollo de *software* para usuarios ciegos no está dirigido por ninguna metodología específica, por lo que en general se aplica uno de los procesos de ingeniería de *software tradicional* [49], consiguiendo resultados peores de los deseados a nivel de usabilidad y experiencia de usuario.

Dentro de los aspectos débiles en el estado del arte está principalmente la falta de *feedback* y la interfaz escogida para el videojuego. En otras palabras, las aplicaciones actuales poseen un excelente contenido, pero este no logra transmitirse a cabalidad por malas decisiones a nivel de usabilidad y de la interfaz de usuario.

Un ejemplo de lo anterior es el videojuego educativo *GeoAudioWizard* [30]. Este videojuego posee un *feedback* bastante acertado, pero no lo suficiente para permitir a los usuarios jugar de forma autónoma, sin mencionar que utiliza una interfaz alejada del uso frecuente de las personas; un *joystick* de *Xbox360* en un computador con sistema operativo *Windows*.

Muchos niños no están familiarizados con esta interfaz, por lo que el público objetivo del videojuego disminuye. Los que sí están familiarizados, suelen estarlo solo mediante el uso de la consola de videojuego *Xbox360* y no mediante su uso en una computadora Windows, que requiere conexión y configuración adicional para utilizarse, logrando limitar aún más el uso del videojuego.

El día de hoy la mayor parte de las personas dispone de un *smartphone*, más específicamente el 81 % de los chilenos [8], incluyendo personas ciegas. Por esto es de suma importancia abarcar la mayor cantidad de personas sacando provecho de esta realidad, lo que nos lleva a enfocar el diseño de las aplicaciones a una pantalla táctil de pequeño tamaño, dejando de lado el teclado, el ratón y otras interfaces de acceso menos frecuente.

Respecto al *feedback* para que el usuario pueda entender y jugar un videojuego de forma consciente, el videojuego debe ser capaz de mostrarle al jugador (*feedback*) todo lo que está sucediendo, o al menos lo suficiente para que el usuario pueda desarrollarse en al menos un aspecto del videojuego. Lo que ocurre en este instante es que los videojuegos no están siendo lo suficientemente auto-explicativos, consiguiendo que el usuario no pueda jugar de forma independiente.

La independencia del usuario es fundamental para fomentar el uso de las aplicaciones, más aún cuando se trata de videojuegos con fines educativos, puesto que dicha falta de independencia solo conseguirá que la aplicación no pueda ser utilizada, disminuyendo la posibilidad de generar aprendizaje.

5.2. Problema a nivel de jugabilidad

Los videojuegos educativos no suelen resultar ser videojuegos, en el sentido de que no están enfocados en el *gameplay*, ya que se transforman de forma constante en una herramienta de estudio entre sesiones de juego, sin ningún tipo de esfuerzo por camuflar el aprendizaje de tal manera que el usuario se mantenga entretenido jugando, y no estudiando.

Los videojuegos son adictivos debido a que fueron diseñados con ese fin [31]. A diferencia de los videojuegos, la finalidad de la enseñanza es educar, proceso que requiere además de un esfuerzo no menor para llevar a cabo dicho aprendizaje.

Un ejemplo de lo anterior es el videojuego AudioGeometría. Este videojuego utiliza muy bien lo que es la orientación y movilidad a través de misiones en un mundo 3D virtual pero cae en lo anterior puesto que muestra explícitamente ejercicios de geometría y matemática en ciertas partes del videojuego.

Resumiendo, se requiere que los videojuegos educativos se enfoquen en el *gameplay*, haciendo que el aprendizaje sea una consecuencia indirecta al jugar, sin dejar de ser entretenidos, sin dejar de ser videojuegos y sin hacer pensar al jugador que está en una sesión de estudio.

5.3. Problema a nivel herramientas educativas

Ya se mencionó que las matemáticas y la geometría suelen ser las áreas más difíciles de enseñar a las personas ciegas, por lo que potenciar la forma de enseñar estas áreas con tecnologías de la época puede resultar ser de mucha utilidad tanto para profesores como para estudiantes. Es por ello que el desarrollo de esta memoria busca fomentar el uso y el manejo del plano cartesiano, mediante el reconocimiento de la recta numérica y ejercicios de transformaciones geométricas (traslación y reflexión), área y perímetro, proyección y definición de trayectorias.

5.4. Reducción del problema

Si queremos comenzar a generar aprendizaje mediante los videojuegos, debemos pensar en cómo poder llegar a las personas de la manera más sencilla posible, permitiendo a la vez entregarles una tecnología que cumpla con sus expectativas de entretenimiento (*gameplay*), siendo capaces a la vez de transmitir conocimiento de manera directa o indirecta.

Debemos enfocarnos además en las interfaces que el usuario utiliza día a día, de manera constante y de forma casi automática, debemos enfocarnos en aquella tecnología invisible que nos permitirá generar aprendizaje visible [44]: el teléfono inteligente [8].

Lo que queda por hacer es seguir mejorando la interacción humano-computador para así alcanzar los objetivos educativos de la tecnología de la mejor forma posible, lo que también es parte del videojuego desarrollado.

6. Metodología

La metodología se pensó detalladamente dejándonos llevar a cabo el desarrollo del proyecto de una forma estructurada y ordenada, lo que nos permitió definir tareas especí-

ficas que cumplan los objetivos propuestos de la memoria, dándonos un estado inmediato de los objetivos ya conseguidos y los que quedan por conseguir.

El proyecto tiene tres grandes etapas, las que definen cómo se llevará a cabo:

1. Diseño y desarrollo del prototipo
2. Evaluación de usabilidad y jugabilidad: Se realizará una evaluación constante sobre el desempeño de del videojuego diseñado a nivel de interfaz y jugabilidad en conjunto con usuarios ciegos y con resto visual. Esta evaluación dará origen a un nuevo diseño y desarrollo, mejorando el *software* hasta converger a un videojuego con buen desempeño a nivel de usabilidad y jugabilidad.
3. Análisis y conclusiones: Se analizarán los datos recopilados durante las evaluaciones de usabilidad para concluir sobre la calidad de la usabilidad y jugabilidad del videojuego desarrollado.

A nivel de ingeniería de *software* se utilizó una metodología ágil en donde el prototipo en desarrollo fue utilizado con profesores diferenciales y estudiantes ciegos de forma recurrente para lograr generar un prototipo lo más robusto posible antes de comenzar la evaluación de usabilidad con sus respectivas herramientas. Paralelamente a la evaluación de usabilidad, se continuó con esta metodología de *software* para el rediseño del videojuego dados los resultados de usabilidad y jugabilidad obtenidos de forma semanal.

6.1. Diseño y desarrollo del prototipo

El diseño es sin lugar a dudas el corazón de este proyecto, ya que es en esta etapa donde se define el futuro de uno de los objetivo más importante de esta memoria: el desarrollo de un videojuego para niños ciegos que pueda utilizarse de forma autónoma.

Para lograr esto, la etapa de diseño y desarrollo comenzó con una investigación y análisis de distintos tipos de *software* para usuarios ciegos y con resto visual. La idea de este análisis es poder identificar componentes y funcionalidades que le permitan al usuario utilizar la aplicación de forma autónoma y eficaz.

Luego de identificar los requisitos que debe cumplir una aplicación para ser utilizada de forma autónoma por el usuario, pudimos ser capaces de identificar las ventajas y desventajas de ciertas componentes de los distintos videojuegos analizados que influyen en esta autonomía.

El principal objetivo de este análisis fue detectar qué formas de interacción con la interfaz de usuario funcionan de manera eficiente y cuáles de forma deficiente. La idea

fue reutilizar las interfaces que ya funcionan desde el punto de vista de la usabilidad, evitando a grandes rasgos tener que generar una nueva interfaz desde cero y que deba ser evaluada por completo.

Posterior al análisis de usabilidad se procedió con el diseño conceptual del videojuego utilizando los resultados y conclusiones mencionados, obteniendo un diseño de las distintas interacciones deseadas a través de diferentes interfaces de usuario, las que contemplen un uso autónomo por parte del jugador ciegos o con resto visual.

El diseño conceptual del videojuego fue realizado entorno a un formato estándar de desarrollo de videojuego denominado *Game Design Document* o GDD [57], el cuál propone el diseño de un videojuego en base a ciertos conceptos como vistas o mecánicas, historia y personajes, interfaz y experiencia de usuario, entre otros.

Durante el diseño conceptual del videojuego se definió un conjunto de 6 desafíos geométricos que puedan ser llevados a cabo en un plano cartesiano discreto, los que fueron utilizados a lo largo del videojuego, tanto en el modo historia y como en el modo tutorial.

Estos 6 desafíos consisten de problemas geométricos y utilizan los siguientes aspectos de la malla curricular de los cursos de enseñanza básica desde quinto a octavo año:

1. Transformaciones geométricas: traslación y reflexión
2. Proyección
3. Área y perímetro
4. Trazado de trayectos entre dos puntos del plano cartesiano

El videojuego contiene 2 niveles de dificultad, los que hacen variar el funcionamiento de cada desafío geométrico así como el lenguaje técnico que utiliza el narrador para referirse a ciertos conceptos geométricos y espaciales (ej: *coordenada* y *cuadrícula* para referirse a un punto del plano cartesiano).

Hasta este punto solo tenemos un diseño a nivel conceptual de usabilidad e interfaz de usuario. El diseño a este nivel permite el uso de cualquier tecnología y lenguaje de programación para la implementación, siempre y cuando cumpla con los requisitos de la interfaz de usuario y plataforma escogidas.

Con todo lo anterior listo, diseñamos la aplicación a nivel de ingeniería de *software*, definiendo una arquitectura consistente, ordenada y escalable utilizando el motor de videojuego Unity3D. Durante el desarrollo de la aplicación se comenzó implementando todo lo relacionado a la interfaz de usuario, desde botones simples hasta un sistema completo de menús para ciegos reutilizable dentro de cada parte del videojuego.

Dado que este videojuego busca ser también utilizado como una herramienta educativa, al finalizar el desarrollo del prototipo se procedió con la implementación de un sistema de seguimiento para cada uno de los desafíos jugados por los usuarios. Este sistema de seguimiento registra todas las acciones realizadas por los usuarios durante cada uno de los niveles del juego y sus desafíos, de manera que un evaluador pueda analizar esta información para ver las debilidades y fortalezas de cada niño con discapacidad visual al momento de enfrentar los diferentes problemas geométricos.

6.2. Evaluación de usabilidad y jugabilidad

Durante esta evaluación, el prototipo realizado fue puesto a prueba por estudiantes con discapacidad visual de 5to a 8vo año básico, los que nos permitió obtener resultados del videojuego respecto a su usabilidad y jugabilidad luego de una sesión de juego de aproximadamente 20 minutos.

Para este análisis se utilizó una pauta de usabilidad obtenida mediante la adaptación de otra [19] utilizada en investigaciones de usabilidad previas con niños ciegos. Esta pauta de usabilidad nos permitió a su vez medir aspectos de jugabilidad del videojuego, permitiéndonos observar las debilidades y fortalezas de este. También se utilizó un segundo documento que corresponde a una pauta de observación que señala si los objetivos que éste define se concretaron o no de forma satisfactoria por el jugador durante la sesión de juego.

Cada evaluación nos dio resultados inmediatos que analizamos para obtener las componentes a corregir del videojuego respecto a su interfaz e interacción con el usuario. Para estas correcciones también se consideraron los resultados referente a la jugabilidad y a la motivación generada por el videojuego, otorgando una mejor experiencia a los jugadores durante el desarrollo de las sesiones posteriores.

La pauta consta de preguntas de usabilidad y jugabilidad que permiten la calificación del videojuego con puntuaciones de entre 1 y 10, en donde 1 significa *poco/desacuerdo* y 10 significa *mucho/de acuerdo*. Además posee preguntas de desarrollo en donde el usuario participante puede expresar libremente su opinión del videojuego tras haberlo utilizado.

Las preguntas de la pauta adaptada para esta memoria se clasificaron en preguntas de usabilidad, las que hacen referencia directa al uso de la interfaz y a su interacción con el usuario, y en preguntas de jugabilidad en base a los aspectos de entretenimiento y motivación generados por el videojuego, los ya mencionados en el marco teórico [32]. Las preguntas 7, 9, 10, 11, 13, 16, 17, 19-27 se refieren directamente a usabilidad, y las preguntas 1-6, 12, 14, 15 y 18 a la jugabilidad o *gameplay* (ver anexo).

La pauta de observación se divide en tareas, y cada tarea posee objetivos específicos que deben ser observados en el jugador y luego marcados como cumplidos, no cumplidos o cumplidos parcialmente. Esta división en la pauta de observación está detallada al comienzo del anexo.

En ella se evalúa el entendimiento de los menús, los desafíos en el plano cartesiano y el entendimiento de la interacción con dichas interfaces. Estas observaciones corresponden netamente a aspectos de usabilidad y entendimiento con la interfaz más allá de la jugabilidad.

6.3. Análisis y conclusiones

Al terminar con la evaluación previa se analizaron los resultados obtenidos y se expusieron las fortalezas y debilidades del videojuego referente a la interfaz diseñada y experiencia de juego. También se identificó el trabajo futuro a nivel de investigación que generó el desarrollo de este *software*.

6.4. Etapas metodológicas

1. Investigar, diseñar e implementar una interfaz de menús y opciones para usuarios videntes y no videntes.
2. Implementar un sistema de narración o habla dinámica.
3. Diseñar e implementar vistas principales y la navegación del juego utilizando la interfaz para ciegos y la narración dinámica.
4. Implementar una arquitectura extensible de desafíos dinámicos (módulos) con dos dificultades en un plano cartesiano utilizando la interfaz para usuarios ciegos:
 - (a) Implementar interfaz de módulos extensibles.
 - (b) Implementar interfaz de dificultades.
 - (c) Implementar módulo base y sus recursos correspondientes.
5. Extender el *framework* Unity3D para una sencilla generación de módulos y sus recursos correspondientes.
6. Diseñar e implementar diferentes módulos de desafíos geométricos con diversos objetivos cognitivos de aprendizaje e implementarlos utilizando la interfaz de módulos diseñada.
7. Extender el sistema de módulos para generar módulos tipo tutorial para explicar el funcionamiento de cada desafío generado.
8. Desarrollar un sistema de capítulos y niveles que permita mezclar múltiples módulos para generar una historia y experiencia de juego.

9. Crear una historia predefinida organizada en capítulos y niveles.
10. Implementar un sistema para realizar seguimiento a las acciones realizadas por los usuarios al momento de jugar los distintos niveles del videojuego.
11. Implementar una interfaz para visualizar y analizar el seguimiento guardado de cada nivel y sus módulos jugados
12. Evaluar la usabilidad y jugabilidad del videojuego y rediseñar cuando corresponda.

7. Diseño y desarrollo del videojuego

7.1. Aspectos técnicos

Para el desarrollo de este proyecto se requiere el uso y manejo de las siguientes herramientas técnicas:

- Interfaz: Smartphone/Tablet
- Sistema operativo: Android
- Framework: Unity3D
- Lenguaje de programación C#
- Objetos JSON
- Microsoft Visual Studio

Los aspectos técnicos de la solución se propusieron en base a la premisa del gran uso de *smartphones* o dispositivos móviles por parte de las personas, incluyendo escolares ciegos o de baja visión correspondientes al público objetivo, y a el gran potencial que posee el plano cartesiano para construir todo tipo de problemas geométricos y espaciales para el desarrollo del pensamiento y razonamiento en este ámbito.

La interfaz de usuario utilizada fue la pantalla táctil en su totalidad, aprovechando la vibración de los dispositivos móviles y dejando de lado cualquier otra interfaz que potencialmente disminuya el tamaño del público deseado.

El lenguaje de programación utilizado para la realización del proyecto fue C#. También se utilizaron objetos *JSON* para codificar información estática y dinámica del videojuego, por ejemplo, para generar los capítulos y niveles predefinidos, guardar la partida y generar el registro de seguimiento de cada nivel jugado.

7.1.1. Unity3D

El proyecto de memoria fue realizado en el *framework* de videojuegos Unity3D v5.1.2f1, ya que este permite compilar el *software* a múltiples plataformas, incluyendo Android: el sistema operativo más utilizado en *smartphones* y *tablets*, S.O. que fue escogido para el proyecto por tener una mayor cantidad de usuarios en Chile: un 81 % registrado el año 2018 [8].

Unity3D es un motor gráfico 3D para PC y Mac que viene empaquetado como una herramienta para crear juegos, aplicaciones interactivas, visualizaciones y animaciones en 3D. Unity puede publicar contenido para múltiples plataformas como PC, Mac, Flash, Xbox, PS2/3/4, Android, PSVita y iPhone. El motor también puede publicar juegos basados en web usando el *plugin* Unity web player.

El editor de Unity es lo principal en línea de producción de videojuegos, ofreciendo editor visual con todo lo necesario para crear juegos. El contenido del juego es construido desde el editor y el *gameplay* se programa usando un lenguaje de programación. Esto significa que los desarrolladores no necesitan ser unos expertos en C++ para crear juegos con Unity, ya que las mecánicas de juego son compiladas usando una versión de Javascript, C# o Boo, un dialecto de Python.

Un videojuego construido en Unity consta de un conjunto de *assets*, un conjunto de *scripts* con la lógica del videojuego y un conjunto de escenas que definen el flujo y secuencia del videojuego. Estas últimas hacen uso de los distintos *assets* y *scripts* dentro de una instancia de ejecución, permitiendo la comunicación entre estos recursos.

Los conceptos principales que se requieren entender para poder utilizar Unity3D de forma efectiva son los siguientes:

- **Assets:** Son recursos que funcionan como bloques constructivos de todo lo que Unity requiere en sus proyectos. Se guardan en forma de archivos de imagen, modelos del 3D, archivos de sonido o cualquier otro tipo de recurso que pueda ser consumido para su uso en la aplicación.
- **Componente:** Las componentes corresponden a recursos de la aplicación que poseen funcionalidad, funcionalidad que puede estar aislada o que bien puede ser compartida con otras componentes. Unity trae por defecto componentes que se utilizan como bloques para construir la aplicación, desde algunas muy básicas hasta otras más complejas. Como ejemplo, están los elementos de interfaz de usuario (UI), objetos o elementos 3D (personajes, terrenos, etc), luces e iluminación, lógica y simulación de física, entre otras. Estas componentes hacen uso de diversos *assets* para poder conseguir todo tipo de especialización, como lo puede ser un botón que utilice una fuente distinta a las por defecto, una imagen de fondo y un sonido al

presionarlo.

- **GameObject:** Un *game object* corresponde a una agrupación de componentes que está siendo usada en una escena de juego dentro del entorno de ejecución actual. Todo *GameObject* trae por defecto la componente *Transform* la cual dice al motor Unity la posición, rotación y la escala del objeto.
- **Script:** El *script* corresponde a un archivo que contiene lógica de ejecución. Este *script* puede utilizarse como una componente en un *gameobject* o llamada dentro de otro *script* más grande. Una vez que el *script* se ejecuta como una componente, puede comunicarse con todas las demás componentes del *gameobject* asociado, generando todo tipo de lógica deseada.
- **Prefab:** Un *prefab* corresponde a un *asset* de un *gameobject* pre-fabricado. Este se guarda como un recurso el que puede ser consumido por una componente o *script* para crear una instancia dentro de la escena en ejecución. Por ejemplo, un proyecto puede tener un único *prefab* de un personaje de un juego en primera persona, el que puede ser utilizado en diversas escenas del videojuego tan solo creando una copia cuando se es necesario. Del mismo modo pueden tenerse *prefabs* de plantas y árboles, permitiendo la reutilización de este recurso.

7.2. Contexto del videojuego

StarAudioGeoxies, el *software* desarrollado para la memoria, es un videojuego del género arcade basado en Space Invaders del mismo género.

Posee un enfoque futurista en donde el jugador debe comandar una nave espacial, batallar contra enemigos extraterrestres y superar diferentes obstáculos que se llevarán a cabo en diferentes galaxias.

Dentro de los objetivos del jugador está principalmente el derrotar enemigos, esquivar obstáculos dañinos y lanzar rayos láseres. El jugador además deberá interactuar con el entorno teniendo que realizar tareas más complejas como rodear planetas, generar trayectorias, transportar elementos de un lugar a otro, entre otras tareas espaciales.

7.3. Análisis de usabilidad de softwares para ciegos

Para la realización de este análisis se utilizó la siguiente pregunta base: ¿puede el niño ciego utilizar el videojuego de forma autónoma, es decir, sin una persona que lo acompañe ni entregue soporte sobre la usabilidad y los objetivos del videojuego? Luego de esta pregunta base, el análisis consistió en buscar principalmente componentes del

videojuego que disminuyeran el entendimiento del videojuego y por tanto disminuyendo también la autonomía al momento de jugar.

Para determinar qué aspectos del videojuego disminuyen la autonomía de este, se propusieron 3 requisitos mínimos necesario para que un videojuego pueda ser utilizado de forma autónoma. De esta forma se podría identificar de forma sencilla aquellos aspectos que dificultan la autonomía del videojuego.

Dichos requisitos propuestos son los siguientes:

- El usuario debe ser capaz de identificar en todo momento el o los objetivos actuales de la vista del videojuego en el que se encuentra.
- El usuario debe ser capaz de identificar las diferentes opciones que tiene para interactuar con el videojuego en cada una de sus vistas.
- Además de conocer las diversas interacciones, el usuario debe saber cómo interactuar con ellas mediante la interfaz de usuario y recibir *feedback* inmediato de cada una de sus acciones.

Con estos 3 requisitos sobre el videojuego y la interfaz de usuario, el jugador debería ser capaz de entender todo lo que sucede en el videojuego y de las opciones de interacción de posee, dejando libertad absoluta para que su mente se enfoque en la lógica principal del videojuego y no en entender cómo utilizar el *software*.

Estos requisitos no son un todo o un nada, también puede haber un punto intermedio en donde el videojuego otorgue cierto nivel de independencia al jugador, el que potencialmente quedará en un punto ciego en donde no sabrá qué hacer ni cómo interactuar con la aplicación.

Un ejemplo de lo anterior es una aplicación que dice el objetivo al usuario una única vez, pero que luego no hay modo de repetir dicha instrucción. El usuario podría no escuchar esta explicación o bien no entenderla la primera vez que la escucha, por lo que si no se repite él se quedará eternamente con la duda de lo que debe realizar, impidiendo utilizar la aplicación con autonomía.

Del análisis anterior de algunos videojuegos del estado del arte se llegó a una serie de conclusiones respecto a los aspectos positivos y negativos de cada uno de ellos.

7.3.1. Aspectos positivos

- Navegación por menú

La navegación por menú es un aspecto de usabilidad que trae enormes beneficios al ser utilizado con usuarios ciegos. El permitir a los usuarios navegar y recorrer las opciones disponibles evita que tengan que memorizar posiciones físicas de cada una de ellas y elimina la posibilidad de no encontrar una opción buscada.

Software como Orca y Jaws utilizan esta funcionalidad con el uso de la pantalla y/o el teclado, así como también el videojuego AudioSims que la utiliza con un *joystick*. La navegación por menú alude directamente al segundo requisito expuesto más arriba para generar autonomía en una aplicación.

- Navegación por gestos

Los gestos son una característica única de ciertos tipos de interfaces de usuario basados en háptica, las que son capaces que captar ciertos gestos del usuario mediante diferentes tipos de sensores. La pantalla táctil es una de ellas, esta capta diferentes gestos de las manos y dedos permitiendo diferentes tipos de acciones sin necesidad de añadir un gran número de botones.

A Blind Legend [14] es un excelente ejemplo de la utilización de gestos en pantalla. La navegación por gestos hace referencia al tercer requisito, que es el responsable de que el usuario sepa cómo comunicarse con la aplicación mediante la interfaz de usuario.

- Explicación de la interfaz de usuario

Explicar cómo utilizar la interfaz también forma parte de la interfaz, y esto es sin duda indispensable para los usuarios ciegos. Al no tener una visión que les permita ver, junto a la falta de experiencia en el uso de interfaces, hacen que su *intuición* [25] no sirva de mucho.

Si una interfaz no es lo suficientemente intuitiva, los usuarios no podrán utilizarla eficientemente sin una capacitación previa. Es más, incluso diseñando una aplicación intuitiva, ciertos usuarios con falta de experiencia en interfaces similares seguirán sin poder utilizarla correctamente. Esto es lo que sucede comúnmente cuando adultos alejados de la tecnología intentan hacer uso de ella, generando frustración y ocasionando un rechazo hacia dichos dispositivos.

- Narración

La tecnología de narración (o tecnología de discurso) es sin duda alguna una de las mejores formas de entregar un mensaje concreto a una persona ciega [15]. Este tipo de interfaz otorga una enorme accesibilidad a usuarios ciegos, sirviendo como una interfaz de propósito general sin importar el tipo de *software* o aplicación en la que se utilice.

Dado que la narración solo entrega información al usuario, esta solo da apoyo a los dos primeros requisitos expuestos para generar autonomía. Existen otras interfaces de usuario para el control de una aplicación mediante el uso de la voz y un micrófono, interfaz que no se utilizará ni se analizará en esta memoria.

- Contexto u objetivo

Una aplicación posee diferentes vistas o menús por los cuales el usuario puede navegar. Cada una de estas vistas corresponde a un contexto diferente, los cuales poseen sus propias opciones, objetivos y formas de interacción. El usuario debe saber en todo momento el objetivo del contexto actual, para así interactuar con la interfaz de una manera efectiva. En los videojuegos esto se consigue utilizando la tecnología de narración: el narrador señala al jugador el objetivo junto a las principales acciones que este tiene disponible.

La narración de contexto se relaciona de forma directa con el primer y segundo requisito propuesto para la autonomía del *software*, y es indispensable para poder utilizarlo.

- Re-narración de contexto

Cuando el usuario tarda demasiado tiempo en realizar una acción correcta, posiblemente es debido a que no haya entendido el contexto en el que se encuentra. Por ello algunos videojuegos vuelven a narrar reiteradamente el contexto de la vista actual luego de cierto periodo de tiempo.

Esta práctica en las aplicaciones mejora el nivel de autonomía generado por los dos primeros requisitos, ya que el usuario sabrá, tarde o temprano, los objetivos del juego y las opciones de interacción que posee.

Algunos videojuegos solo poseen esta re-narración mediante la interacción con un botón o menú. Esto puede traer problemas si el usuario no es capaz de encontrar o interactuar con dicha opción que le permitirá entender el objetivo o las opciones que está buscando.

- Sensibilidad en vibración táctil y tono de sonido

Las personas ciegas y con baja visión, por su condición, han desarrollado de mejor manera sus sentidos del tacto [17] y audición. Esto puede ser utilizado para dar diferentes niveles de *feedback* a los usuarios ciegos, basados por ejemplo en el tipo de vibración o el tono de un sonido.

7.3.2. Aspectos negativos

- Dependencia física de la interfaz en la pantalla táctil

Una interfaz en donde la interacción depende físicamente del espacio en la pantalla tiene grandes desventajas para un usuario ciego. Este, al no ver la pantalla, no puede determinar la posición de cada opción, por lo que se debe saturar al narrador innecesariamente para transmitir toda esta información espacial al usuario.

Además de ello, el jugador debe memorizar las posiciones de estas opciones, las que debe recordarlas cada vez que quiera utilizarlas, de no ser así, el narrador deberá repetir una gran cantidad de información una y otra vez. De forma adicional, si estas opciones no están bien posicionadas (por ejemplo, en una esquina o borde), el jugador ciego puede tener dificultades para encontrarlas, aún cuando el narrador especifica la posición exacta de la opción.

Esta dependencia física puede funcionar de buena manera, referente al nivel de

autonomía de la aplicación o videojuego, si se tiene una cantidad adecuada de botones en pantalla. Desafortunadamente, dado que no existe un estándar para el desarrollo de interfaces para ciegos ni para el posicionamiento de ciertos botones, el narrador está obligado a narrar esta información haciendo que la interacción con el videojuego sea ralentizada.

Otra desventaja relacionada con la dependencia física de la interfaz es que suele funcionar peor en pantallas táctiles de mayor tamaño. Esto es debido a que el usuario debe recorrer mayores distancias para detectar las opciones especificadas por el narrador si es que estas no crecen de forma proporcional con la pantalla.

Una solución del problema anterior es utilizar botones de gran tamaño que ocupen gran parte del área de la pantalla para disminuir la posibilidad de que el jugador no encuentre dichas opciones, pero no resuelve del todo el problema principal de la dependencia física mencionado en los párrafos anteriores.

- Narración obligatoria

La tecnología de narración es sin duda una excelente forma de transmitir información, pero puede tornarse una molestia si se hace de forma reiterada a usuarios que ya tienen conocimiento del *software* o videojuego.

La idea principal del narrador es informar al usuario de los objetivos del videojuego y/o de la interfaz de usuario, y aquí es donde surge una inquietud con el trabajo del narrador: ¿es necesario informar al usuario de forma reiterada y descontrolada de algo de lo que ya fue informado con anterioridad?.

Dentro de puntos anteriores mencionamos la importancia de repetir una instrucción, pero en este instante nos contradecimos diciendo que puede ser contraproducente. Si no repetimos la instrucción o contexto estamos originando una enorme posibilidad de que un usuario pierda el enfoque de lo que debe realizar, por otro lado, si repetimos demasiado estamos retrasando y frustrando al usuario con un discurso que no quiere continuar escuchando.

La narración obligatoria es una característica que está presente en varios de los videojuegos diseñados para personas ciegas, como por ejemplo el videojuego Geo-AudioWizard [30], en donde el usuario debe escuchar de forma obligatoria una y otra vez los mismos audios antes de poder comenzar a jugar, lo que también delimita la experiencia de usuario.

La solución encontrada a esta problemática es no utilizando un narrador obligatorio, dicho con otras palabras, utilizar un narrador que solo sea escuchado cuando el usuario lo desee, entregándole la posibilidad de omitir el narrador cada vez que lo considere necesario para continuar avanzando con el videojuego.

Esto puede realizarse de una infinidad de maneras, según el tipo de interfaz que posea la aplicación en cuestión. En el caso de dispositivos móviles, explícitamente celulares y *tablets* con pantallas táctiles, esto podría realizarse con un simple gesto o toque en la pantalla en el momento en el que el narrador comienza su discurso.

- *Feedback* incompleto

Si el *feedback* no es lo suficientemente claro y explicativo, entonces el jugador tendrá dificultades para entender la lógica del videojuego o su interfaz de usuario. Pero es aún peor cuando el videojuego no entrega *feedback* alguno de lo que está sucediendo en el espacio virtual.

Una aplicación puede tener un *feedback* de excelente calidad: claro, conciso y auto-explicativo, pero de no tenerlo en cada momento en el que se requiere no servirá de mucho para generar autonomía en el jugador que enfrente el videojuego.

Para verlo de mejor manera, tomemos como ejemplo un automóvil a medio terminar: el auto puede tener una calidad increíble tanto en su piezas como en su rendimiento, pero si al utilizar las llaves para encender el motor no obtenemos *feedback* alguno de que está encendido, no sabremos si podemos comenzar a manejar o no, o si hay algún otro conflicto de por medio que debe ser solucionado.

En el caso de los usuarios ciegos esto es mucho más notorio puesto que la falta de retroalimentación culmina de inmediato en una desorientación de lo que sucede en el videojuego. Aplicaciones que tengan esta problemática no están cumpliendo los primeros dos requisitos propuestos para un diseño autónomo, o bien los cumplen solo durante una parte de su funcionamiento. Es sumamente importante que el *feedback* sea entregado en todo momento, sin dejar espacio alguno para la desorientación del usuario.

Si una aplicación carece de retroalimentación frente a ciertos estímulos del usuario o frente al desarrollo de su propia lógica, el usuario todavía puede utilizarla de forma autónoma sin ningún tipo de desorientación o disminución en esta. Para que esto sea posible, el resto del *feedback* de la aplicación debe complementar la información no transmitida con anterioridad.

Para ejemplificar lo anterior, consideremos un jugador ciego que intenta abrir una puerta cerrada del videojuego utilizando un conjunto de llaves virtuales que encontró. Al intentar abrir, el jugador escucha el sonido de las llaves, pero no sabe si la puerta abrió o no puesto que faltó *feedback* de una puerta abriéndose. El jugador está desorientado, no sabe si la puerta está o no abierta, por lo que intenta pasar a través de ella dándose cuenta de inmediato de que sí lo está, gracias al *feedback* que las pisadas al caminar le dan. De forma análoga puede enterarse de inmediato que la puerta continúa cerrada si al intentar atravesarla se da cuenta de que está chocando con esta.

- Desuso del contraste y tamaño de fuente

Dado que estamos tratando con personas legalmente ciegas, algunas con cierto grado de visión, debemos tomar en cuenta la componente visual al momento de diseñar e implementar una aplicación o videojuego. Es por lo anterior que, para una completa inclusión de las personas ciegas, utilizamos colores de alto contraste y fuentes de gran tamaño para permitir a estas personas hacer uso de su resto visual [10].

Dentro de este mismo aspecto de inclusión al ciego con resto visual, entra también el uso de imágenes e iconos para llamar la atención de los usuarios sugiriendo una acción o transmitiendo información del contexto virtual de la aplicación. Al igual que el texto, los iconos e imágenes del videojuego también deben ser de gran tamaño para poder ser visualizados apropiadamente.

Los colores e imágenes no deben ser escogidos al azar, deben tener coherencia con la propuesta y entorno virtual del videojuego. Esto es dado que las imágenes (en conjunción con los sonidos de voz y efectos especiales), ayudan a generar una mejor inmersión en el videojuego potenciando la jugabilidad y experiencia de usua-

rio, por tanto, afectando la motivación del jugador y el aprendizaje sobre el uso de la interfaz.

- **Interfaz de usuario de poco acceso**

Una de las grandes desventajas de diversos videojuegos educativos actuales ha sido la interfaz de usuario que han escogido. No porque sean interfaces mal diseñadas o difíciles de usar, sino más bien por la experiencia que los usuarios tienen en ellas dado el escaso acceso que tenemos a dichas interfaces por el contexto social en el que vivimos. Esto lo mencionamos en puntos anteriores aludiendo a la tecnología invisible [44].

Para que un videojuego educativo pueda generar aprendizaje, este se debe utilizar. Para que este pueda ser utilizado se deben tener en cuenta ciertos aspectos de su interfaz, principalmente, el hecho de que el usuario tenga experiencia con la interfaz evitando generar frustración al momento de usarla.

Para el diseño de la interfaz de StarAudioGeoxies se tomaron todos estos aspectos en consideración, corrigiendo los negativos y potenciando los positivos. Con ello, se desarrolló un prototipo de videojuego que intenta plasmar de la mejor manera posible todo lo que está sucediendo en el videojuego, permitiendo que el jugador se centre en comprender el argumento del videojuego y no en cómo utilizarlo.

Existen otras ventajas y desventajas que surgieron de la evaluación de usabilidad de StarAudioGeoxies, evaluación que se expondrá en los puntos siguientes junto a dichas ventajas/desventajas y a las soluciones propuestas para corregir y potenciar estos aspectos.

7.4. Fortalezas del videojuego desarrollado

El diseño del videojuego está pensado de tal forma de que este tenga ciertas fortalezas y ventajas respecto a otros videojuegos educativos dentro del estado del arte. Estos últimos fueron analizados en diferentes ámbitos originando una serie de desventajas, las cuales fueron corregidas durante el diseño para dar origen al diseño que StarAudioGeoxies propone.

7.4.1. Diseño enfocado en la jugabilidad

El diseño del videojuego está enfocado en la experiencia de juego (*gameplay*). Esto quiere decir que StarAudioGeoxies busca que el jugador conecte con lo que está presenciando, generando la motivación suficiente para que este quiera continuar avanzando y resolviendo los problemas que se le presenten.

Ya se comentó que la motivación es de suma importancia al momento de generar aprendizaje [31] [32], por lo mantener la atención del jugador es el primer paso antes que el videojuego educativo comience a educar.

Este enfoque en la jugabilidad es lo que permite que el videojuego se pueda utilizar tanto dentro de la sala de clases como fuera de ella. Los niños ciegos pueden aprender geometría utilizando el videojuego mediado por el profesor o profesora dentro de la sala de clases, y reforzar con ejercicios al jugar en su tiempo libre fuera de la sala.

De no tener una experiencia de usuario lo suficientemente atractiva, el videojuego solo se utilizaría dentro de la sala de clases, sin garantía de que los estudiantes quieran realmente enfocar sus mentes en resolver los problemas que expone el profesor mediante el videojuego. Es muy importante que el videojuego no sea una “herramienta para estudiar” para los usuarios, sino que más bien sea un videojuego del que pueden disfrutar, tal como cualquier otro.

Para conseguir un nivel lo suficientemente atractivo de jugabilidad se tomaron en cuenta cuatro de los seis factores expuestos en el modelo *Gameplay Enjoyment Model* (GEM) [32]:

- Desafío: Aludiendo principalmente a los desafíos geométricos del plano cartesiano.
- Exploración: La exploración se utiliza permitiendo al jugador buscar en la pantalla diversas entidades durante un desafío geométrico.
- Fantasía: Esta característica alude al uso de un entorno virtual futurista y de fantasía llevado a cabo en el espacio exterior, con planetas, naves, alienígenas, campos de energía, asteroides, tormentas solares y todo tipo de entidades utilizadas con el fin de dar al usuario un entorno virtual más atractivo.
- Fidelidad: Para esto se utilizaron efectos de sonidos para fortalecer el ambientación fantástica expuesta y gráficos lo más acorde posible a dicho entorno, junto con animaciones y colores brillantes.

Además de estos cuatro factores para generar entretenimiento se propusieron 2 adicionales:

- Nivel de interacción con el videojuego: Dando al usuario opciones variadas para interactuar con el videojuego.
- Evitar el lenguaje educativo explícito: Evitar el lenguaje formal, técnico y explícito de matemática y geometría para ayudar a mantener al usuario inmerso en un entorno de fantasía, concentrado en el videojuego.

StarAudioGeoxies se diseñó con gran variedad de acciones con las que el jugador puede interactuar dentro de la componente principal del videojuego. Esto permite que

el jugador no tenga solo dos opciones para avanzar en el videojuego, la correcta y la errónea, sino que también puede tomar alternativas intermedias para llegar a la solución solicitando pistas del problema a resolver o consumiendo recursos (que debe acumular) para resolverlo de forma automática.

No se utilizó lenguaje educativo formal y técnico para no generar al jugador la sensación de que está “estudiando” en vez de utilizarlo con un videojuego. Por ejemplo, en vez de decir “Busca la coordenada (2, 1) en el plano cartesiano para resolver el problema” se dice “Encuentra al enemigo en la coordenada (2, 1) y atácalo para vencerlo”.

Para continuar fomentando la inmersión en el entorno virtual generado por el videojuego, se utilizaron sonidos acordes al contexto espacial que el videojuego expone. De forma adicional, las voces de los personajes fueron grabadas por personas reales y no por *software* de narración como lo hacen otras aplicaciones, lo que es mucho más llamativo para los niños ciegos. Los personajes poseen un rol dentro del videojuego, rol que se ve caracterizado e interpretado por las personas que dieron vida al personaje mediante el uso de su voz.

Dado que el videojuego contempla también a personas con resto visual, la componente gráfica de StarAudioGeoxies también posee un enfoque futurista/espacial, en donde se utilizan colores de *neon* e imágenes acordes al contexto espacial.

Finalmente, para conseguir una mejor experiencia de juego se tomaron en consideración los comentarios y evaluaciones de los usuarios durante el proceso de evaluación de usabilidad y jugabilidad. De esta evaluación surgieron diversos cambios para mejorar el interfaz de usuario y la experiencia de juego, las que se comentarán a detalle en los capítulos siguientes específicos de dicho proceso de evaluación.

7.4.2. Diseño enfocado en la usabilidad

Con StarAudioGeoxies buscamos que no sea necesario ningún especialista o profesor que dirija la sesión de juego, otorgando completa libertad al jugador para que utilice el videojuego cada vez que lo considere pertinente, tal como lo hace habitualmente con sus otros quehaceres o pasatiempos.

Es por lo anterior que este proyecto se ha enfocado fuertemente en el análisis de otros videojuegos y en el diseño de una interfaz usable, buscando que esta sea auto-explicativa e intuitiva para los usuarios ciegos.

De forma indirecta, se busca además que este gran enfoque en el diseño de una inter-

faz usable potencie el uso libre, independiente y autónomo del videojuego, abarcando una mayor cantidad de usuarios a través del diseño móvil pensado para el sistema operativo Android. Una mayor cantidad de usuarios, con un mayor tiempo de uso, permitirá un mayor tiempo de exposición a los distintos problemas geométricos del videojuego, abriendo la posibilidad a un desarrollo cognitivo mediante el uso de la aplicación.

Cabe señalar que la interfaz emula el funcionamiento de Jaws [37] utilizado en sistemas operativos como Windows, utilizado en las escuelas, por lo que usuarios que sepan utilizar dicha interfaz no tendrán problemas en utilizar este videojuego.

7.4.3. Plano cartesiano y variedad de problemas

El diseño del videojuego fue realizado junto con profesores de enseñanza a niños ciegos, con los definimos una serie de contenidos críticos en geometría que requieren apoyo, y que son necesarios para potenciar el entendimiento del mundo espacial que nos rodea en el día a día.

Estos contenidos, tal como se mencionó en la metodología de esta memoria, son:

- Transformaciones geométricas: traslación y reflexión
- Proyección
- Área y perímetro
- Trazado de trayectos entre dos puntos del plano cartesiano

Si bien solo existen 6 desafíos geométricos que abordan estos contenidos, la variedad de problemas que se pueden realizar en el plano cartesiano son muchísimos, por lo que este videojuego tiene un gran potencial por la versatilidad que tiene el uso del plano cartesiano.

El diseño de StarAudioGeoxies ha sido de tal manera que generar nuevos problemas geométricos es una tarea realmente sencilla, por lo que, desde el punto de vista educativo, puede usarse para definir cualquier desafío o problema geométrico que requiera de un plano cartesiano.

7.4.4. Repetición y aleatoriedad

Para entender la importancia de la aleatoriedad debemos comprender lo que es el aprendizaje [2]:

“El aprendizaje también es el puente entre el conocimiento y la experiencia, ya que cuando la experiencia es comprendida, apropiada, se convierte en una forma especial de conocimiento que genera capacidad para crear información y guiar la experiencia posterior.”

El aprendizaje solo es aprendizaje cuando a partir de una experiencia, generamos conocimiento suficiente para guiar una experiencia posterior. Si dicha experiencia posterior, es una experiencia idéntica a la anterior, entonces no estamos presenciando aprendizaje, sino más bien el simple uso de la memoria.

Es por ello que el videojuego de esta memoria no es un juego repetitivo: cada uno de los desafíos geométricos es diferente a los anteriores jugados, gracias al uso de la aleatoriedad. Esto evitará el uso de la memoria para resolver los problemas geométricos, forzando el uso de la mente y el pensamiento en cada uno de ellos.

7.4.5. Registro y seguimiento

Un sistema de registro no es indispensable, pero si muy útil para un educador que busca identificar las falencias del pensamiento geométrico de un niño ciego al momento de enseñarle contenidos de esta área.

El sistema de registro de este videojuego permite observar y analizar toda la actividad realizada por los jugadores en los distintos niveles jugados, entregando información de las distintas acciones realizadas por el usuario y el tiempo que tardó cada una de ellas.

Este análisis del registro permitirá a los profesores y educadores enfocarse en los aspectos débiles de los niños, para enseñarles de una manera más personalizada y eficiente, sin tener que estar evaluándolos de forma presencial al momento de utilizar el videojuego.

Cabe señalar que este registro se guarda de forma automática dentro del dispositivo móvil en el cuál se utiliza el videojuego, y solo se puede acceder a este desde el menú principal del juego.

7.5. Interfaz y usabilidad del videojuego

Tal como se mencionó en el análisis de usabilidad realizado previamente, parte del diseño de usabilidad de este videojuego consistió en la solución de los aspectos negativos de parte de las aplicaciones para ciegos existentes.

Recordemos que estos aspectos apuntan a que la aplicación pueda ser utilizada de manera independiente, según los tres requisitos de usabilidad que propusimos durante dicho análisis.

A continuación se mencionará una lista con los aspectos positivos de usabilidad utilizados y cómo se intentó solucionar los negativos.

Navegación por menú

La dependencia física de la interfaz se solucionó implementando la navegación por menú, lo que significa que el usuario no deberá encontrar ninguna opción físicamente en la pantalla, y que siempre tendrá acceso a ellas mediante sencillas acciones sobre cualquier parte de la pantalla.

Esta navegación por menú funciona de tal manera que solo una opción de las disponibles es mostrada en el centro de la pantalla a la vez, las que pueden ser navegadas o seleccionadas utilizando los gestos que se explicarán en el punto siguiente.

Esta no dependencia física del menú, utilizando la navegación, permite al jugador poder recorrer y consultar todas las opciones disponibles del videojuego.

Navegación por gestos

La navegación a través de la interfaz se realiza mediante 3 simples gestos sobre cualquier punto de la pantalla táctil:

- Deslizar el dedo en cualquier dirección (vertical, horizontal o diagonal) para avanzar o retroceder a través de las opciones disponibles.
- Tocar una vez para consultar la opción mostrada en pantalla.
- Tocar dos veces seguidas para accionar la opción en pantalla.

De forma adicional, cabe señalar que hay opciones que en ciertos momentos están desactivadas y que deben ser desbloqueadas, por lo que el usuario recibirá *feedback* con un sonido negativo al intentar accionarlas con el gesto pertinente.

Explicación de interfaz

La explicación de la interfaz y de cómo utilizarla es mencionada mediante un narrador cada vez que el jugador ingresa al menú principal del videojuego. Aquí se explica al usuario que debe utilizar los 4 gestos antes mencionados para utilizar el videojuego.

Dado que no todo el juego es un menú con una lista de opciones para escoger, el narrador de este videojuego explica cómo utilizar la interfaz cuando el jugador enfrenta los módulos geométricos al utilizar el plano cartesiano. En tal caso los gestos se mantienen, pero el resultado de dicha interacción depende de cada módulo geométrico.

Narración

Dado que la narración por voz ha resultado ser de enorme ayuda en todos los videojuego y aplicaciones donde se utiliza, este aspecto se mantuvo durante todo el videojuego, siendo la principal fuente de información para el usuario ciego.

Narración de contexto

El narrador además de mencionar cómo utilizar la interfaz, señala en todo momento el objetivo que tiene el jugador en la vista actual del videojuego. Esto se realiza cada vez que el usuario navega a una vista diferente, en donde el primer *feedback* del narrador es mencionarle al usuario el contexto en dónde se encuentra y qué es lo que debe realizar.

Específicamente hablando, el narrador señala al jugador:

- Lo que debe hacer en el menú/vista actual y/o las diferentes opciones que hay disponibles.
- La opción del menú que el usuario acaba de consultar o a la que acaba de navegar utilizando los gestos.

La idea es que este narrador entregue toda la información posible para que el jugador sepa exáctamente qué hacer en cada menú y de todas las opciones disponibles que tiene.

Re-narración

Con una frecuencia de 10 segundos, el narrador repite el contexto y objetivo de la vista actual en la que el jugador se encuentra para que este recuerde constantemente qué es lo que debe realizar en el videojuego.

Narración voluntaria

El jugador puede silenciar y omitir lo que el narrador está diciendo tocando la pantalla dos veces seguidas. Esto permitirá que el jugador pueda seguir avanzando y no tenga que escuchar el narrador si no desea hacerlo.

Esta funcionalidad está desactivada dentro del tutorial del videojuego, ya que en este caso el narrador explica a detalle lo que el jugador debe realizar para entender la lógica del videojuego y sus distintos módulos geométricos.

Contraste y fuente

StarAudioGeoxies utiliza colores de alto contraste entre sí para todos sus botones y opciones, lo que permitirá, junto al gran tamaño del texto, que los usuarios con resto visual puedan diferenciar algunas letras e incluso palabras completas.

Lamentablemente, algunas de las imágenes *open source* utilizadas no poseen un buen contraste, lo que puede causar potencialmente dificultades para distinguir ciertas entidades del videojuego durante los desafíos geométricos. Esto último lo dirá la evaluación de usabilidad, de la que se habla más adelante.

7.5.1. Vistas y menús

La gran mayoría de las vistas del videojuego corresponden a un menú deslizable basado en el funcionamiento de Jaws [37]. Al entrar a cualquier menú del videojuego un

narrador lee el objetivo del menú o el contexto actual del juego, de esta forma el usuario sabrá lo que debe hacer en cada parte del videojuego.

Las opciones poseen un texto de un máximo de 5 letras aproximadamente, debido a que se utilizó una fuente de gran tamaño pensando en los niños con restos visuales. La interacción con este tipo de menús es mediante los 3 gestos mencionados en los puntos anteriores: deslizar, tocar una vez y tocar dos veces seguidas la pantalla.

Menú principal

El menú principal tiene tres opciones, una para iniciar una nueva partida, otra para cargar la partida guardada y una última para jugar el tutorial. El narrador señalará las opciones disponibles al entrar a este menú.

Al cargar la partida se irá directamente a la vista de *capítulos y niveles*. Allí estarán disponibles todos los capítulos jugados y sus respectivos niveles.

Además, el menú principal posee un botón en la esquina superior derecha, el que permite a los educadores acceder al sistema de registro de los niveles jugados en el dispositivo móvil.



Figura 4: Menú principal.

Menú de dificultad

Este menú tiene dos opciones de dificultad: novato y avanzado, los que comenzarán el juego en el primer nivel del primer capítulo al escoger una de las opciones. Adicionalmente posee una opción para volver al menú principal.



Figura 5: Menú de dificultad.

Menú de tutorial

Aquí se exponen un total de 8 opciones, 6 para los tutoriales de cada uno de los módulos del videojuego, una opción adicional para jugar el tutorial completo, que explica tanto los módulos como el uso del plano cartesiano, y una última opción para volver al menú principal.



Figura 6: Menú de tutorial.

Menú de capítulos y niveles

En este menú el usuario puede navegar a través de los diferentes capítulos de la historia del videojuego (explicado más adelante) y escoger sus niveles para jugarlos.

Inicialmente este menú solo muestra los niveles del capítulo actual según en avance en la historia del videojuego. Además, la primera opción permite continuar el videojuego en el último nivel de el último capítulo desbloqueado.

Los niveles bloqueados se muestran en rojo y al seleccionarlos emiten un sonido negativo.



Figura 7: Menú de capítulos y niveles.

Interfaz principal de juego

El plano cartesiano es la interfaz principal del videojuego. Aquí se juegan y resuelven los diferentes módulos de desafíos geométricos que hay durante el tutorial y durante el modo historia.

Esta interfaz se compone principalmente por una cuadrícula cartesiana de tamaño 10x6, situando la coordenada (0,0) al centro de la pantalla.

Además de estar presente la nave espacial que controlará el usuario, en esta interfaz se muestran las diferentes entidades de cada desafío geométrico: enemigos, asteroides, campos energéticos y otros que se detallarán más adelante.

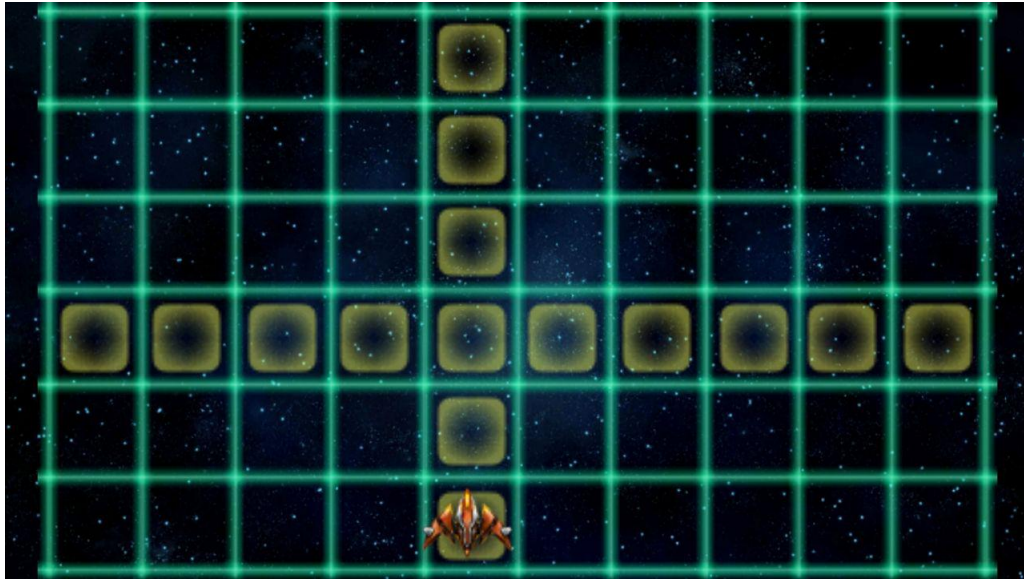


Figura 8: Interfaz de cuadrícula cartesiana para resolver problemas geométricos.

El usuario puede tocar cada una de las cuadrículas para preguntar su coordenada, y en el caso de haber un elemento en ella, el narrador lo señalará justo después de señalar la coordenada de la cuadrícula tocada.

Al deslizar el dedo por las cuadrículas, el usuario sentirá una vibración y un sonido característico dependiendo de si se está moviendo hacia los números positivos o negativos. También sentirá un sonido característico si al deslizar el dedo encuentra alguna de las entidades antes mencionadas.

Menú de acciones

Durante la interfaz del plano cartesiano existe un menú oculto al cual el usuario puede acceder tocando la pantalla con dos dedos de manera simultánea. Este menú posee diferentes opciones sumamente importantes que ayudarán al jugador a completar los diferentes módulos:

- **Objetivo:** Repite la última instrucción del narrador.
- **Refuerzo:** El jugador recibe ayuda del narrador para resolver el desafío actual. Consume 1 punto de utilidad al usar.
- **Control automático:** La nave realiza una acción de forma automática para resolver uno de los desafíos del nivel. Consume 3 punto de utilidad al usar.
- **Puntos de utilidad:** El narrador señala la cantidad de puntos de utilidad que posee el jugador. Los puntos de utilidad se consiguen al completar un nivel y se puede tener

una cantidad máxima de 9.

- Módulo: El narrador señala el módulo/desafío actual dentro del nivel.
- Nivel: El narrador señala el nivel actual dentro del capítulo.
- Salir: Opción para salir al menú de niveles.

Este menú se cierra de forma automática luego de un par de segundos, siempre y cuando el usuario no esté interactuando con este.



Figura 9: Menú de acciones del tutorial o modo historia.

7.6. Sistema de niveles

Para comprender mejor el funcionamiento del videojuego, sus niveles y módulos, es preciso señalar primero cómo se estructura un nivel del videojuego.

Cada capítulo del videojuego (definidos más adelante según la historia), posee cierta cantidad de niveles, y cada nivel posee cierta cantidad de módulos que el usuario debe superar para completar el nivel.

El nivel puede tener de 3 a 4 módulos, módulos que se definen según el avance del videojuego. Estos módulos aparecen en orden aleatorio pero de forma secuencial, se juega uno a la vez.

El usuario puede completar un nivel resolviendo correctamente al menos la mitad de los módulos, por lo que en caso de no poder resolver un desafío geométrico, el jugador

puede intentar resolver el siguiente.

Los niveles definidos por los capítulos tienen un único orden, y un nivel solo puede ser jugado si se completó de manera satisfactoria el nivel anterior. De la misma forma, los capítulos son desbloqueados cuando todos los niveles del capítulo anterior son completados.

7.7. Módulos geométricos

Como se señaló anteriormente, un módulo corresponde esencialmente a un desafío geométrico dentro del plano cartesiano, y actualmente existen 6 de ellos.

Los módulos se dividen en diferentes etapas que definen un flujo lineal dentro de este y los momentos en el que el usuario puede operar con el plano cartesiano. Estas etapas son las siguientes:

1. Inicio de módulo:

Aquí se explica al jugador a grandes rasgos cuál es el objetivo que debe cumplir para poder completar satisfactoriamente el módulo. Por ejemplo, se le señalará que hay un enemigo y que se debe dispararle con un rayo láser.

2. Inicio de ronda:

Se explicará específicamente el objetivo de cada una de las rondas del módulo. Por ejemplo, se señalará la cantidad exacta de asteroides que hay en la ronda actual para esquivarlos. En esta etapa del módulo se crean instancias de todas las entidades necesarias para construir un problema geométrico.

3. Periodo de exploración:

Corresponde a un periodo de tiempo en donde el jugador tendrá la posibilidad de explorar completamente el plano cartesiano, para así identificar las posiciones de cada una de las entidades presentes en la ronda actual del módulo. Esta etapa es crucial puesto que aquí el jugador deberá generar un mapa mental de todo el entorno para poder resolver el problema. No todas las rondas poseen periodo de exploración, pero la primera de ellas siempre la posee.

4. Periodo de acción:

Cuando termina la exploración de ronda, el jugador ya debería tener un mapa mental formado, por lo que durante este periodo debe ejecutar una o más acciones sobre las cuadrículas en orden de cumplir el objetivo señalado al comenzar. Algunas rondas son de acción única, como lo puede ser lanzar un rayo láser, lo que quiere decir que solo debe accionarse una cuadrícula del plano cartesiano para resolver el problema. De manera análoga, en las rondas de acción múltiple se deberá accionar un conjunto de cuadrículas para resolver el problema, como lo puede ser en el trazado de la trayectoria de un misil dirigido.

5. Periodo de respuesta:

Algunos módulos tienen la posibilidad de reaccionar a las acciones que ejecuta el jugador. Esta etapa se ejecuta justo después de la anterior en dichos casos. Por ejemplo, al lanzar un rayo láser, el enemigo se moverá a otra cuadrícula, lo que corresponde a la reacción de la ronda. Existen módulos sin reacción.

6. Fin de ronda:

En el fin de ronda se decide si aún hay rondas restantes en el módulo, por lo lo que se transfiere el flujo a una nueva ronda o al fin del módulo dependiendo de cual sea el caso.

7. Fin de módulo:

Periodo que comienza al terminar todas las rondas, aquí se determina y señala si el jugador completó satisfactoriamente el módulo o no. En cualquiera de los casos, de aquí se pasa al siguiente módulo de existir uno.

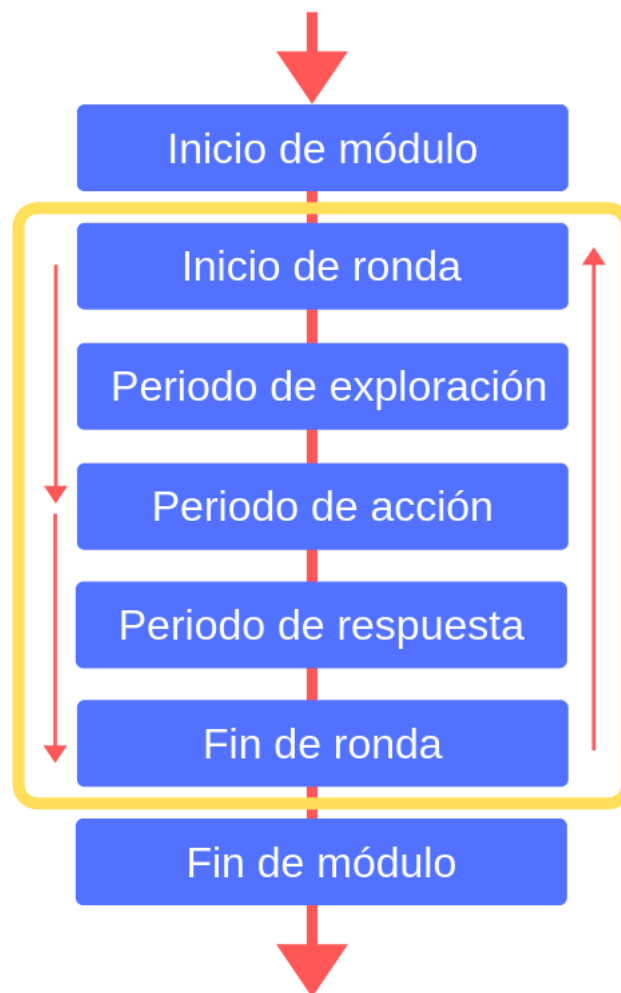


Figura 10: Estructura y flujo de un módulo geométrico.

Los módulos geométricos fueron diseñados para desarrollar un área específica de la geometría. El listado completo de módulos se lista a continuación.

7.7.1. Módulo 1: Ataque con rayo láser

El jugador debe mover la nave a la derecha o a la izquierda situando la nave justo debajo del enemigo que aparece en pantalla para atacarlo con un rayo láser vertical.



Figura 11: Módulo de rayo láser.

7.7.2. Módulo 2: Ataque con misil dirigido

El jugador debe trazar una trayectoria desde la nave hasta el enemigo para dispararle el misil, sin importar el trayecto que escoja.

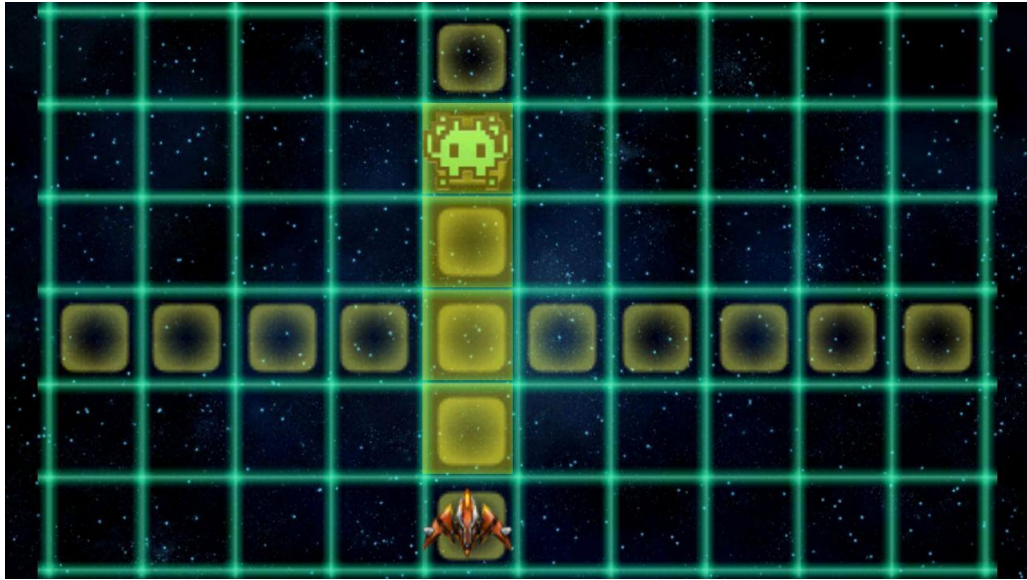


Figura 12: Módulo de lanzamiento de misil.

7.7.3. Módulo 3: Esquivar asteroides

El jugador debe mover la nave a una cuadrícula que deje la nave a salvo cuando los asteroides se muevan de manera vertical u horizontal (según diga el narrador).



Figura 13: Módulo de esquivar asteroides.

7.7.4. Módulo 4: Espiar planetas enemigos

El jugador debe navegar la nave por el perímetro del cuadrado que encierra todos los planetas del desafío. Puede hacerlo en sentido horario o antihorario.

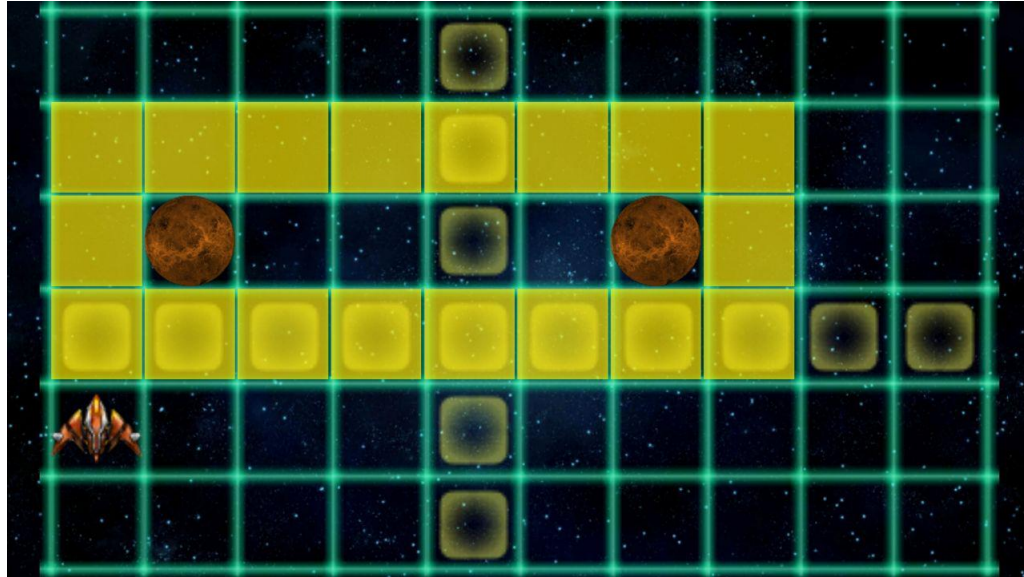


Figura 14: Módulo de espiar planetas enemigos.

7.7.5. Módulo 5: Extracción energética

El jugador debe extraer 3 unidades de energía del campo de energía y depositarlos en el planeta especificado. El desafío es que el narrador solo señalará la esquina superior derecha e inferior izquierda del campo de energía, por lo que el jugador debe ser capaz de identificar el área completa del campo para extraer la energía de sus cuadrículas.



Figura 15: Módulo de extracción energética.

7.7.6. Módulo 6: Esquivar tormentas solares

El jugador debe mover la nave a una cuadrícula que esté fuera del área de las tormentas solares. Nuevamente, el narrador solo identificará las tormentas solares por sus esquinas.

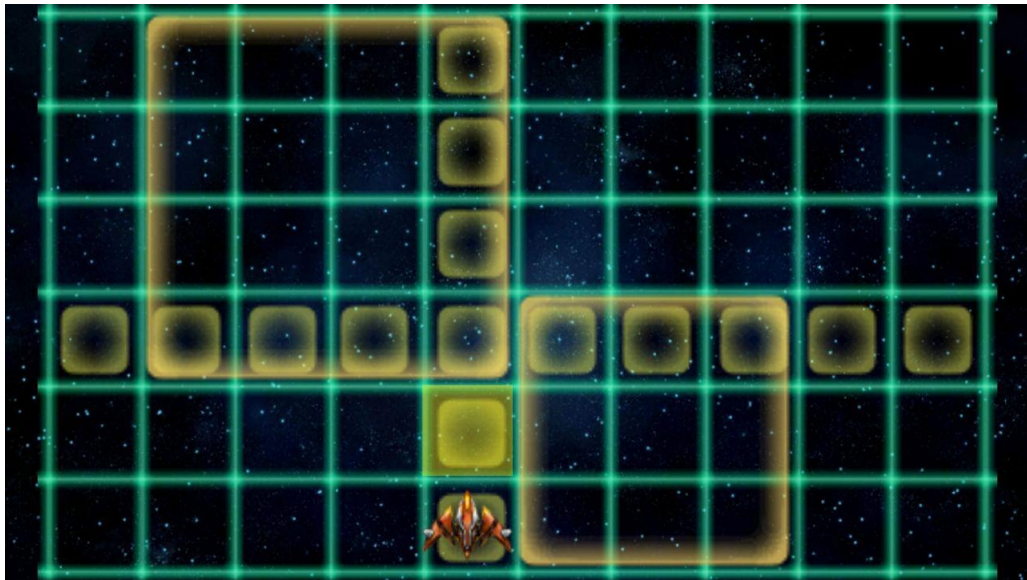


Figura 16: Módulo de esquivar tormentas solares.

7.8. Sistema de registro y seguimiento

El sistema de registros corresponden a dos vistas diseñadas para ser utilizadas por un especialista en enseñanza a niños ciegos. A diferencia de las demás partes del videojuego, estas vistas están enfocadas en personas videntes por lo que no hacen uso de la interfaz para ciegos diseñada en esta memoria.

Una de estas vistas muestra una lista resumida con todas las partidas jugadas, con la fecha y módulos geométricos jugados y superados correctamente. Esta vista global es útil para determinar con una simple mirada en cuáles módulos geométricos está teniendo dificultades un jugador en particular. Dado que los módulos se relacionan de forma directa con un contenido de la geometría, este registro dirá cuál es el contenido que debe reforzar cada niño en específico.

| Dificu. | # | Módulos Completados | Tiempo | Módulos jugados |
|---------|----|---------------------|--------|------------------------|
| HARD | 14 | 4 / 6 | 5:24 | Laser Asteroid Energy |
| HARD | 13 | 3 / 4 | 3:57 | Laser Asteroid Missile |
| HARD | 13 | 2 / 2 | 2:15 | Laser Missile |
| HARD | 12 | 2 / 6 | 5:24 | Laser Asteroid Energy |
| HARD | 11 | 2 / 6 | 5:24 | Laser Asteroid Energy |
| HARD | 10 | 2 / 6 | 5:24 | Laser Asteroid Energy |

SALIR

NIVEL 14

MÓDULOS

- Laser
- Asteroides
- Laser
- Extraer Energía
- Laser
- Asteroides

Figura 17: Vista de la lista de partidas jugadas registradas. Se muestran en verde los módulos que el jugador pudo completar correctamente y en gris los que no.

La vista de registro de un nivel permite ver a detalle todas las acciones realizadas por el jugador en los distintos módulos geométricos del nivel. Cada acción posee una coordenada y una categoría asociada que indican el tipo de acción que se realizó, y si esta fue o no una acción satisfactoria dentro de la lógica del videojuego.

El especialista puede navegar a través de los distintos módulos jugados, lo que le

permite enfocarse solo en aquellos que el jugador no pudo completar. Esto le permite identificar el punto exacto en que este comenzó a fallar al enfrentar cierto módulo geométrico.

Además de tener acceso al registro de acciones, el especialista tiene acceso a una recreación visual e interactiva de la partida jugada por el usuario. Por lo tanto, puede analizar las acciones del usuario de una manera más completa ya que tiene acceso a todo el contexto de la partida del jugador.

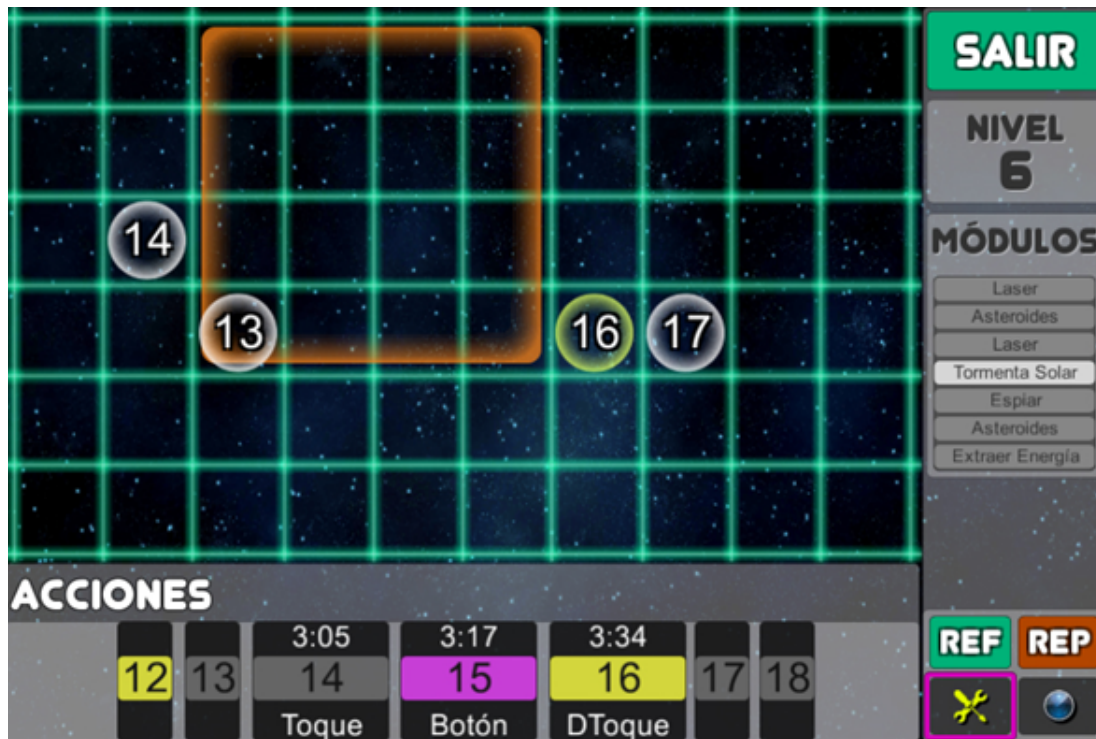


Figura 18: Vista del registro de un nivel. Abajo se muestra la secuencia de acciones realizadas por el jugador, distinguiéndolas por color. En gris los toques simples sobre la cuadrículas, en amarillo los toques dobles (acciones) y en morado las acciones sobre botones del menú. La vista principal muestra la recreación de la partida jugada y al costado derecho los módulos jugados y completados.

7.9. Historia del videojuego

“La Tierra está siendo explotada por extraterrestres de una lejana galaxia y la humanidad está a pocas décadas de desaparecer. Por ello la valiente joven Levy Wolfgang, comandante primera de la armada espacial, junto a su equipo deben hacer lo posible por llegar a la galaxia enemiga y detener a los extraterrestres de sus planes sobre la tierra.”

La historia de StarAudioGeoxies se divide en capítulos, cada capítulo se caracteriza mayormente con uno de los módulos (desafíos geométricos) del juego. Existe un total de 7 capítulos. Cada capítulo a su vez se divide en pequeños avances en la historia, que corresponden a un nivel del videojuego.

Personajes

- Levy Wolfgang: Género femenino. Comandante primera de la armada espacial y protagonista de la historia. La/el jugador debe identificarse con este personaje al jugar. Habla únicamente en el capítulo final, durante la historia de uno de los niveles.
- Drake Foxlite: Género masculino. Jefe de la armada y superior directo de Wolfgang. Él narra la historia al inicio de cada nivel para conseguir la victoria contra los extra-terrestres.
- Beatrix Mihawk: Género femenino. Asistente de comunicaciones en el centro espacial en el planeta Tierra. Narra las acciones que debe realizar la/el jugador (Levy) para completar cada uno de los módulos. Narra los menús, botones y opciones de todo el juego.

7.9.1. Narración de la historia

A continuación de muestran los diálogos narrados de la historia del videojuego en cada capítulo, en cada uno de sus niveles.

Capítulo 0: Entrenamiento galáctico

1. Drake Foxlite: "Levy Wolfgang, soy el líder Foxlite. Hoy comienza tu entrenamiento galáctico. ¡Prepárate! No será nada fácil."
2. Drake Foxlite: "Has sobrepasado la primera etapa de entrenamiento Wolfgang, no te confíes demasiado."
3. Drake Foxlite: "¡Disparar un rayo láser, esquivar asteroides, dirigir un misil o espiar planetas enemigos sólo son algunas de tus misiones, deben prepararte para todo eso y más!"
4. Drake Foxlite: "Wolfgang, llegó la hora. Termina esta última sesión y adéntrate al vacío del universo"

Capítulo 1: Entrada al vacío

1. Drake Foxlite: “¿Sientes una energía diferente?, ¡¿Ah?! Wolfgang?! Estás a punto de enfrentar desafíos reales. Continúa la expedición a través del vacío.”
2. Drake Foxlite: “¡Eh! ¡Wolfgang! resultaste ser todo un guerrero, pero todo guerrero necesita compañeros a su lado. Si no sabes qué hacer en alguna misión presiona el botón Refuerzo de tu nave, te ayudaremos en lo que nos sea posible.”
3. Drake Foxlite: “Aquí Foxlite. Wolfgang, ¿me copias? El rayo láser es un arma excelente contra nuestros enemigos. Utilízala con sabiduría, esta sólo funciona contra extraterrestres.”
4. Drake Foxlite: “Wolfgang no olvides que el planeta tierra está en tus manos. Dirígete hacia la parte densa de la galaxia.”

Capítulo 2: Densidad en la lejanía

1. Drake Foxlite: “Este denso lugar está plagado de peligrosos asteroides, navega el espacio con cuidado.”
2. Drake Foxlite: “Wolfgang, como te has dado cuenta, no puedes destruir asteroides con tu laser, así que esquivalos a toda costa, o las armas definitivas de tu nave serán dañadas.”

Capítulo 3: El arma definitiva

1. Drake Foxlite: “¿Has probado ya el misil dirigido Wolfgang? Puedes dispararlo desde el cañón de la izquierda, derecha o del de arriba de la nave. Repáralos con el botón Reparar si es necesario”.
2. Drake Foxlite: “Wolfgang ten mucho cuidado, estás en una zona con muchos enemigos. Recuerda que algunos de ellos evitarán tu misil con un escudo de energía”.
3. Drake Foxlite: “Muy bien Wolfgang, como esperaba. Sigue tu camino hacia los planetas lejanos, pero no te acerques demasiado. Tu próxima misión será allí.”

Capítulo 4: Espionaje indiscreto

1. Drake Foxlite: “¿Ves algo extraño en esos planetas Wolfgang? Es como piensas, son planetas enemigos. Sé sigiloso, deberás extraer información de utilidad de ellos.”
2. Drake Foxlite: “Aquí Foxlite. Wolfgang. Ahora sabemos que nuestros enemigos mantienen a los humanos con vida para robar su tecnología y conocimiento. Continúa con la misión de espionaje.”
3. Drake Foxlite: “¡Wolfgang! No tenemos tiempo. ¡Los extraterrestres roban energía de los planetas hasta hacerlos desaparecer! Ve en dirección al sol cercano.”

Capítulo 5: Vidas al borde de la destrucción

1. Drake Foxlite: “Los planetas están al borde de la destrucción. Los campos de energía cercanos ayudarán a reabastecer de vida los planetas. Wolfgang, si vez algún planeta cerca, tu misión será salvarlo.”
2. Drake Foxlite: “¡Wolfgang!, ¡has encontrado la ubicación del enemigo! Debes ir al más allá utilizando el agujero chrono-espacial que está cercano al sol.”

Capítulo 6: La amenaza solar

1. Drake Foxlite: “¡Wolfgang! Hemos detectado tormentas solares. Si encuentras alguna, evítala lo más rápido posible.”
2. Drake Foxlite: “Ya estás muy cerca del agujero chrono-espacial. Buen trabajo Levy. Sigue así.”
3. Drake Foxlite: “Wolfgang, sólo unos kilómetros más y estarás en el más allá para acabar con nuestros enemigos.”

Capítulo 7: En el más allá

1. Drake Foxlite: “¡¿Wolfgang?! ¿Puedes oírme? Soy Foxlite ¡¡Wolfgang!! ¡¡WOLFGANG!!”.
2. Levy Wolfgang: (Radio sin señal) – “Foxlite. Aquí Wolfgang, pronto estaré de vuelta. . .” (Radio sin señal) (Radio sin señal)
3. Beatrix Mihawk: “Comandante Wolfgang, ¿es usted? Soy Beatrix, los enemigos han abandonado el planeta, estamos a salvo.” – Levy Wolfgang: “Así que lo he conseguido. . . ¿Dónde está Foxlite?” – Beatrix: “Foxlite. . . luego de perder contacto contigo. . . perdió el control y fue a combatir al enemigo al espacio, no sabemos nada de él... De todas formas, felicitaciones Levy, misión cumplida.”

7.10. Unity 3D: recursos y arquitectura

Este proyecto consta con una serie de recursos que fueron estructurados de tal forma que el acceso a ellos dentro de la UI de Unity3D fuera de forma inmediata e intuitiva. Esta estructura es utilizada comúnmente en el desarrollo de videojuegos permitiéndole a cada quién trabajar con sus propios recursos sin entorpecer a los demás.

Los recursos utilizados están compuestos principalmente de *scripts* de lógica e inter-

faz, *prefabs*, escenas y recursos de sonido e imagen, los que se estructuran en diferentes carpetas, que detallaremos a continuación:

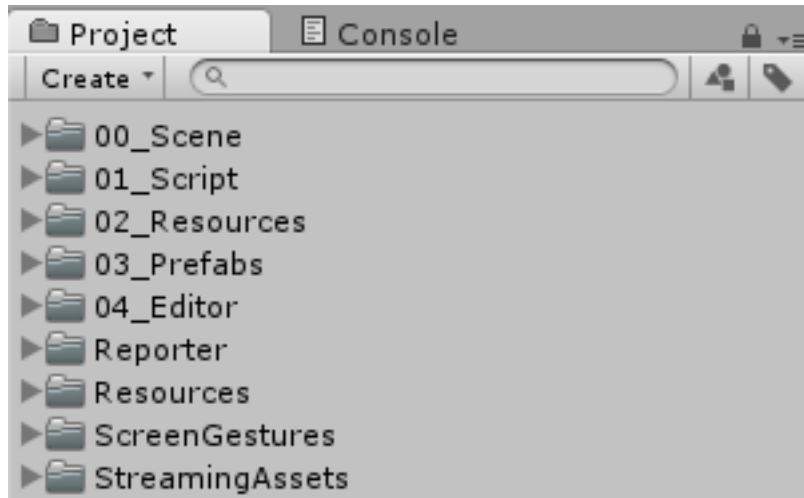


Figura 19: Estructura de carpetas en Unity3D.

- **Scripts:** Los *scripts* contienen la mayoría de la lógica de juego detrás de la aplicación. Aquí se encuentra la funcionalidad para instanciar componentes o recursos en el videojuego, sistema de módulos, capítulos y niveles, plano cartesiano y todas sus interacciones, persistencia de datos, registro del videojuego, animaciones y diversas utilidades.

Esta carpeta de recursos no contiene ningún tipo de lógica relacionada con el sistema de menús para ciegos, puesto que este se realizó en una carpeta aparte, con sus componentes aisladas para poder reutilizarla de manera sencilla en cualquier videojuego o aplicación.

- **Recursos:**

Esta carpeta contiene de forma ordenada todas las imágenes, audios, fuentes y música del videojuego. Nos aseguramos de solo utilizar recursos *open source* para no tener dificultades con la publicación futura del videojuego.

Aquí encontramos las imágenes de cada entidad del videojuego: asteroides, nave espacial, enemigos y planetas, así como también imágenes asociadas a la interfaz como los iconos y otros recursos necesarios para dibujar el plano cartesiano y la demás lógica del videojuego.

- **Escenas:**

Este videojuego posee un total de 8 escenas distintas, 5 de ellas correspondientes al videojuego, 2 al seguimiento y análisis de los jugadores y una última escena correspondiente a la introducción del videojuego. Las 7 primeras fueron descritas en el diseño del videojuego en capítulos anteriores.

Unity3D utiliza dichas escenas para que el usuario pueda navegar a través del videojuego interactuando con la interfaz expuesta. Cada escena contiene una serie de recursos base que utiliza para cargar una vista predeterminada con cierta lógica de juego definida por el desarrollador.

Como desarrollador, podemos acceder y manipular estas escenas abriéndolas con el editor de Unity3D, en donde podemos añadir todo tipo de recursos y la lógica de juego deseada en la escena. Utilizando esta misma interfaz de Unity3D, podemos ejecutar la escena obteniendo de forma inmediata los resultados de lo que estamos desarrollando, sin tener que pasar por un flujo de escenas anteriores para llegar a la actual, que es la que queremos *testear*.

No es necesario utilizar múltiples escenas en un videojuego, pero es muy útil para utilizar solo los recursos necesarios por una vista, desechando los que no lo son reduciendo el consumo de memoria y CPU del dispositivo.

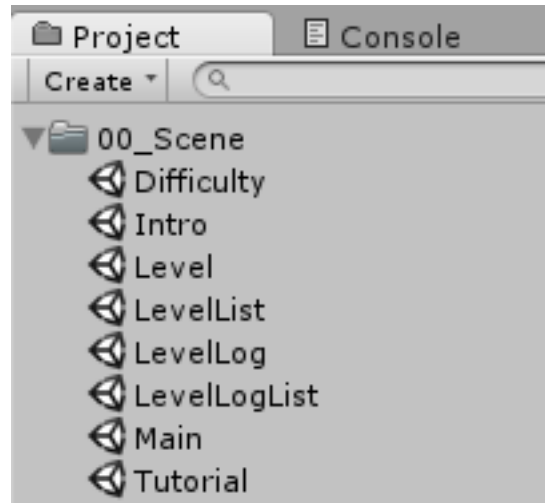


Figura 20: Escenas de StarAudioGeoxies en Unity3D.

- Prefabs:

Cuando hablamos de Unity3D en capítulos anteriores comentamos que este trabaja con *game objects*, que corresponden a un conjunto de componentes en tiempo de ejecución. Un *game object* puede utilizarse en más de una ocasión, por lo que es necesario crear un modelo pre-fabricado de este para instanciarlo cada vez que sea necesario, permitiendo su reutilización, optimizando recursos del dispositivo que ejecuta el videojuego.

StarAudioGeoxies reutiliza la gran mayoría de sus componentes, desde las más sencillas hasta las más complejas, es por ello que se creó una carpeta dedicada a los distintos *prefabs* que el proyecto posee. Algunos de ellos corresponden a las entidades que utilizan los módulos geométricos del videojuego, los que se reutilizan constantemente.

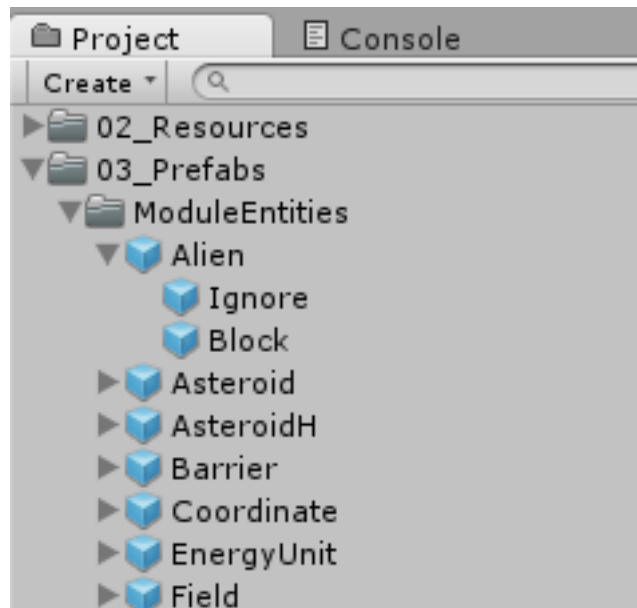


Figura 21: Lista reducida *prefabs* de entidades utilizadas en los módulos geométricos del videojuego.

Para comprender mejor el funcionamiento de los *prefabs* detallaremos el correspondiente a la entidad “alien” utilizada en uno de los módulos geométricos.

Tal como indica la figura siguiente, este *prefab* contiene 3 componentes que definen su funcionamiento y rol en el videojuego: *Rect Transform*, *Canvas Renderer* y *Area Entity (Script)*.

La primera componente es la componente fundamental y requerida por Unity3D en todos sus *game objects*. Esta permite dar al objeto una transformación, la que define la posición, rotación y escala del objeto. Con estas propiedades el *game object* puede ser *renderizado* en el espacio virtual tridimensional o bidimensional del videojuego. *Canvas Renderer* es una componente requerida para que el objeto pueda ser dibujado en un entorno virtual 2D.

La parte relevante de este *prefab* es el *script Area Entity*, el que contiene lógica de esta entidad en el videojuego, lógica que se utiliza dentro de otro *script* que se encarga de instanciar este *prefab* en el entorno de ejecución.

A simple vista se pueden observar las variables *x* e *y*, que definen la posición del alien en el plano cartesiano del videojuego. El *prefab* por defecto contiene los valores (1, 3), los que son irrelevantes pues se sobre-escriben al crear una nueva instancia del alien en el entorno de ejecución.

Aquí se pueden definir otros valores que son reutilizados por cualquier alien instanciado en el videojuego, como por ejemplo, las variables *Identification Speech* (otro *prefab*) y *Dynamic Name* (un *string*).

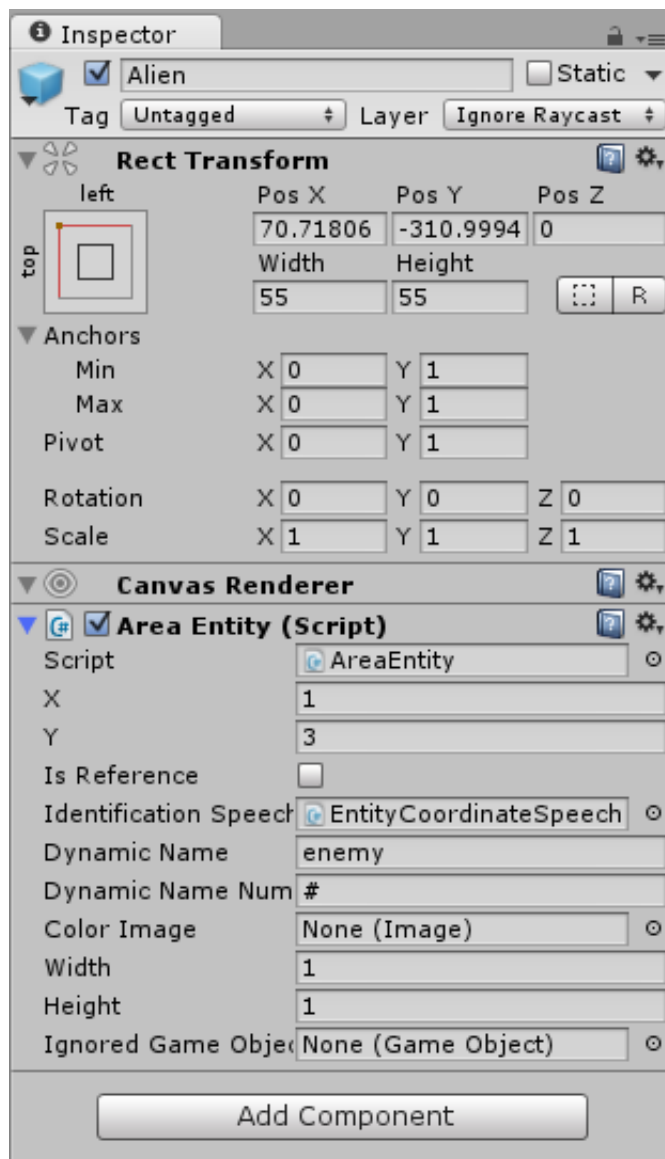


Figura 22: Vista de inspección del *prefab* de la entidad alien en Unity3D.

Un último detalle importante a recalcar es que un *prefab* puede contener otro *prefabs* con sus propias componentes y lógica de juego. En este caso el *prefab* de alien contiene dos *prefabs* adicionales, los que se encargan de la interacción con el usuario y dibujar la entidad en pantalla.

- Editor:

Esta carpeta tiene un significado especial para Unity3D. Aquí se incluyen todos los recursos y *scripts* que son utilizados para extender la funcionalidad del editor de Unity3D.

Estos recursos se utilizan solo en el entorno de desarrollo, por lo que, a diferencia de los demás recursos, estos se descartan al momento de construir la versión de producción del videojuego. Dado lo anterior, cualquier referencia que tengan los *scripts* de la lógica del videojuego a los *scripts* del editor hará que la compilación del videojuego falle.

- Otras carpetas:

Reporter es conjunto de *assets* que da soporte al desarrollo de un videojuego o aplicación creada por medio de Unity3D. Contiene un conjunto de recursos y *prefabs* que convierte el registro de mensajes y errores de consola en vista sencilla de texto dentro del videojuego que la utiliza. De esta manera, podremos acceder a estos mensajes desde la versión compilada del videojuego, desde un dispositivo Android.

Screen Gestures es uno de los recursos más importantes de este proyecto. Contiene el sistema de interfaz de usuario desarrollado durante esta memoria para usuarios ciegos. Esta carpeta está aparte pues sus recursos están encapsulados y pueden ser utilizados en cualquier aplicación o videojuego desarrollado con Android.

La carpeta *Streaming Assets*, al igual que la carpeta *Editor*, es una carpeta especial para Unity3D. Dentro de esta carpeta puede situarse cualquier recurso o archivo que requiera ser modificado en tiempo de ejecución del videojuego.

Esta carpeta es de suma importancia, puesto que, el archivo de instalación de videojuego de Android (con extensión *apk*), genera carpetas de solo lectura, impidiendo añadir información dinámica al videojuego, como guardar una partida. Es justamente esto lo que soluciona la carpeta anterior.

7.11. Arquitectura y diseño de software

El videojuego desarrollado consta de diferentes piezas de *software* que fueron diseñadas e implementadas de manera independiente, las que luego fueron puestas conjuntamente en escena para conseguir el videojuego implementado.

La comunicación entre las distintas componentes de este videojuego se realiza mediante el uso de *callbacks* asíncronos mediante suscripciones a ciertos eventos de cada una de las componentes.

A continuación se presentan las principales componentes del *software* diseñado.

7.11.1. Speech dinámico

El *Speech* es la componente de *software* que se encarga de la narración o discurso por voz mediante la unión de diversas unidades de discurso o *SpeechUnit*. Cada unidad de discurso es equivalente a un *clip* de audio, por lo que un *Speech* es un conjunto de audios que se ejecutan secuencialmente para dar origen al discurso.

El discurso (*Speech*) funciona como un simple *clip* de audio para el resto de las piezas de *software* que lo utilicen, a pesar de que internamente esté compuesto por un listado más pequeño de audios. Esto permite que un *Speech* pueda ser utilizado por cualquier aplicación o videojuego de Unity3D de una manera muy sencilla, tan solo definiendo la lista de audios a utilizar.

Un *Speech* permite tener cierto control sobre cada una de sus unidades, permitiendo recortar de forma automática los silencios de cada audio, o escogiendo en qué minuto o segundo del *clip* deben comenzar a reproducirse o en cuál terminar.

La componente de narración por si sola no es tan potente, simplemente equivale a una lista de audios más pequeños que se reproducen de forma secuencial como si fueran uno solo, salvo por el par de utilidades previamente mencionadas.

La parte realmente relevante de esta pieza de *software* está en la componente *DynamicSpeech* o discurso dinámico, clase que extiende un *Speech* y que permite variar las unidades de narración de forma dinámica. Esta componente se apoya de una reserva de unidades de audio, permitiendo construir en todo momento cualquier tipo de discurso utilizando cualquier combinación de dichas unidades audio.

Esta reserva de unidades de audio, llamada *SpeechUnitDictionary*, es utilizada por la componente de discurso dinámico para actualizar sus *clips* de audio por otros dentro de esta reserva, otorgando el dinamismo necesario para tener un nuevo discurso.

Un uso de esta componente en el videojuego es construir la narración para mencionar una coordenada de manera dinámica al momento de jugar los módulos geométricos. Un ejemplo es teniendo un *SpeechUnitDictionary* con los audios de numeración del 1 al 9, y construir un *DynamicSpeech* mediante la interfaz de Unity3D con los audios restantes necesarios para narrar una coordenada (“coordenada” y “coma”). Con estos dos audios y la reserva de audios numéricos podremos narrar cualquier coordenada al jugador.

7.11.2. Interfaz para ciegos

StarAudioGeoxies tiene dos principales interfaces de usuario diseñadas para ciegos: la interfaz de menú y la interfaz principal del videojuego. En este apartado solo será expuesta la primera de ellas, debido a que la interfaz de la segunda es muy similar y reutiliza casi todo el diseño de la primera.

La interfaz de usuario de este videojuego funciona principalmente con dos componentes: *ScreenHandler* y *SlideMenu*.

ScreenHandler es la componente que se encarga de recibir y procesar la interacción del usuario con la pantalla, percibiendo solo aquellos gestos que se desean otorgando la posibilidad de enviar dichas interacciones a otras piezas de *software*. De esta manera, *ScreenHandler* puede diferenciar un toque en pantalla de dos toques consecutivos, o el toque con un dedo a uno con dos o más dedos, filtrando los que sean necesarios.

SlideMenu es un menú simple que recibe una lista de opciones y acciones mostrándolas en pantalla utilizando un botón de Unity3D completamente personalizable. Esta componente tiene métodos públicos para avanzar a la siguiente opción o retroceder a la anterior, emitiendo ciertas señales a dichas opciones para consultarlas o accionarlas.

La comunicación entre estas dos componentes es sencilla, *SlideMenu* registra sus métodos públicos en *ScreenHandler* de forma que cuando el usuario realiza cierta acción, esta última componente la detecta, y en caso de que *SlideMenu* se haya registrado a dicha acción, acciona el método público registrado, navegando por el menú o realizando una acción sobre las opciones existentes: reproduciendo un *Speech/DynamicSpeech* o seleccionando la opción actual.

La componente *ScreenHandler* en conjunto con *SlideMenu* son los encargados de la interfaz de ciegos en este videojuego. *ScreenHandler* detectando toques simples, dos toques consecutivos y los deslices por la pantalla, mientras *SlideMenu* se encarga de responder a estas acciones, diciendo a las opciones que reproduzcan su sonido asociado según la acción realizada.

7.11.3. Módulos

Este *software* ha sido diseñado para poder construir módulos geométricos de una manera sencilla y escalable. La componente *Module* define todo lo esencial que debe tener un módulo geométrico para poder funcionar, basándose en el ciclo de vida de un módulo según lo especificado en el punto 7.7. Esta componente permite definir la cantidad de rondas, las entidades del módulo y la lógica para instanciarlas, los tipos de acciones disponibles, duración de la exploración o periodo de acción, la lógica de respuesta de cada ronda, etc, así como también todos los elementos de narración (*Speech/DynamicSpeech*) asociados a cada uno de estos eventos.

Cada uno de los módulos existentes extiende a *Module* para añadir lógica especializada de cada problema geométrico y todos los recursos específicos a dicho módulo. Los módulos existentes son los siguientes:

- LasserModule
- AsteroidModule

- MissileModule
- SpyModule
- EnergyModule
- SolarStormModule

Cada una de estas componentes utiliza módulos de dificultad que especifican aspectos de la lógica de juego de cada módulo geométrico, como la cantidad de enemigos, tipos de reacciones del enemigo, tamaño de las entidades, etc.

La vida de un módulo funciona en base a *callbacks*, trasladando el módulo de un estado a otro según la lógica definida por cada uno de ellos.

Tenemos un ejemplo de lo anterior en el siguiente código de la clase *Module* escrita en C#:

```
protected virtual void OnReactionEnd()
{
    RemoveRoundWaste ();
    if(ReactionEnd != null)
        ReactionEnd();
    if (roundsLeft == 0)
    {
        nextAction = OnModuleEnd;
        ModuleFinish();
    }
    else
    {
        nextAction = OnRoundInitEnd;
        StartRound();
    }
}
```

Por defecto, luego de realizar una acción, el módulo (de estar definido en la lógica como en el caso de *AsteroidModule*) debe reaccionar a esta acción: por ejemplo una animación de los asteroides moviéndose y una verificación de colisión con sus sonidos correspondientes. Al terminar este proceso, se ejecuta la función *OnReactionEnd* que señala el término de la reacción.

Al terminar, se remueven los desperdicios de la ronda (eliminación de asteroides para liberar recursos) y se ejecutan todas las acciones registradas en *ReactionEnd*. Posteriormente se inicia una nueva ronda si es que aún quedan rondas pendientes, y en el caso contrario, se pasa a la finalización del módulo.

Todos los estados del módulo operan de una manera similar. El código con todo este detalle puede encontrarse en el anexo.

8. Evaluación de usabilidad y jugabilidad

8.1. Muestra

Esta evaluación se realizó con estudiantes ciegos de 5to a 8vo año básico, de entre 11 a 14 años de edad. Todos estos estudiantes fueron evaluados durante su estadía en el colegio de ciegos Santa Lucía en la comuna de La Cisterna en la Región Metropolitana. Dentro del total de 11 evaluados, 9 de ellos presentan ceguera parcial y solo 2 ceguera

total.

8.2. Procedimiento

Para llevar a cabo la evaluación del *software*, se solicitó explícitamente a los usuarios jugar el tutorial del videojuego y escuchar al narrador atentamente.

Además de esto realizamos una pequeña introducción del videojuego para generar motivación e interés, señalando además que se trataba de una actividad completamente voluntaria y sin ningún tipo de responsabilidad académica ni moral, evitando generar presión y estrés a los usuarios.

Cada sesión de evaluación duró aproximadamente entre 10 a 30 minutos según el nivel de conocimiento y resto visual de los usuarios del videojuego. El tutorial fue diseñado para ser completado en alrededor de 20 minutos.

Durante el periodo de la sesión se tomaron apuntes sobre distintas observaciones de la interacción entre el usuario y la interfaz del videojuego, detectando deficiencias en el *software* para las mejoras futuras.

Al finalizar cada sesión se procedió con la evaluación de usabilidad y jugabilidad a través de las preguntas expuestas en el documento del punto 6.2 de esta memoria.

8.3. Resultados

A continuación exponemos los resultados obtenidos realizando el procedimiento señalado con los 11 usuarios mencionados. Las preguntas fueron calificadas con un valor mínimo de 1 y un máximo de 10 por los distintos usuarios que utilizaron el videojuego.

| | Promedio |
|--|----------|
| 1 Me gusta el videojuego | 8,5 |
| 2 El videojuego es entretenido | 8,2 |
| 3 El videojuego es desafiante | 8,4 |
| 4 El videojuego me hace estar activo | 9,6 |
| 5 Volvería a jugar con el videojuego | 8,9 |
| 6 Recomendaría este videojuego a otros niños/jóvenes | 8,7 |
| 7 Aprendí a jugar con este videojuego | 8,8 |
| 8 El videojuego tiene distintos niveles de dificultad | 7,1 |
| 9 Me sentí controlando las situaciones del videojuego | 7,6 |
| 10 El videojuego es interactivo | 8,7 |
| 11 El videojuego es fácil de usar | 7,6 |
| 12 El videojuego es motivador | 8,3 |
| 13 El videojuego se adapta a mi ritmo | 8,1 |
| 14 El videojuego me permitió entender nuevas cosas | 8,6 |
| 15 Me gustan los sonidos del videojuego | 9,5 |
| 16 Los sonidos del videojuego son claramente identificables | 8,5 |
| 17 Los sonidos del videojuego me transmiten información | 9,0 |
| 18 Me gustan las imágenes del videojuego | 9,1 |
| 19 Las imágenes del videojuego son claramente identificables | 9,6 |
| 20 Las imágenes del videojuego me transmiten información | 9,2 |
| 21 Entiendo los menús del videojuego | 8,0 |
| 22 Puedo identificar claramente las opciones de los menús | 8,2 |
| 23 La cantidad de botones y opciones es adecuada | 8,9 |
| 24 Puedo identificar cuando paso de un menú a otro | 8,9 |
| 25 Entiendo lo que debo hacer en cada parte del videojuego | 8,0 |
| 26 Sé cuando una acción es realizada o no lo es | 8,7 |
| 27 Es posible jugar sin un asistente | 9,4 |

Figura 23: Tabla del promedio de las evaluaciones obtenidas del documento de evaluación de usabilidad por 11 usuarios ciegos y con resto visual.

Para observar esta información de una manera más legible la separaremos en las dos categorías mencionadas en capítulos anteriores: usabilidad y jugabilidad.

De los anterior generamos dos gráficos de resultados, esta vez utilizando preguntas específicas a la usabilidad o a la jugabilidad.

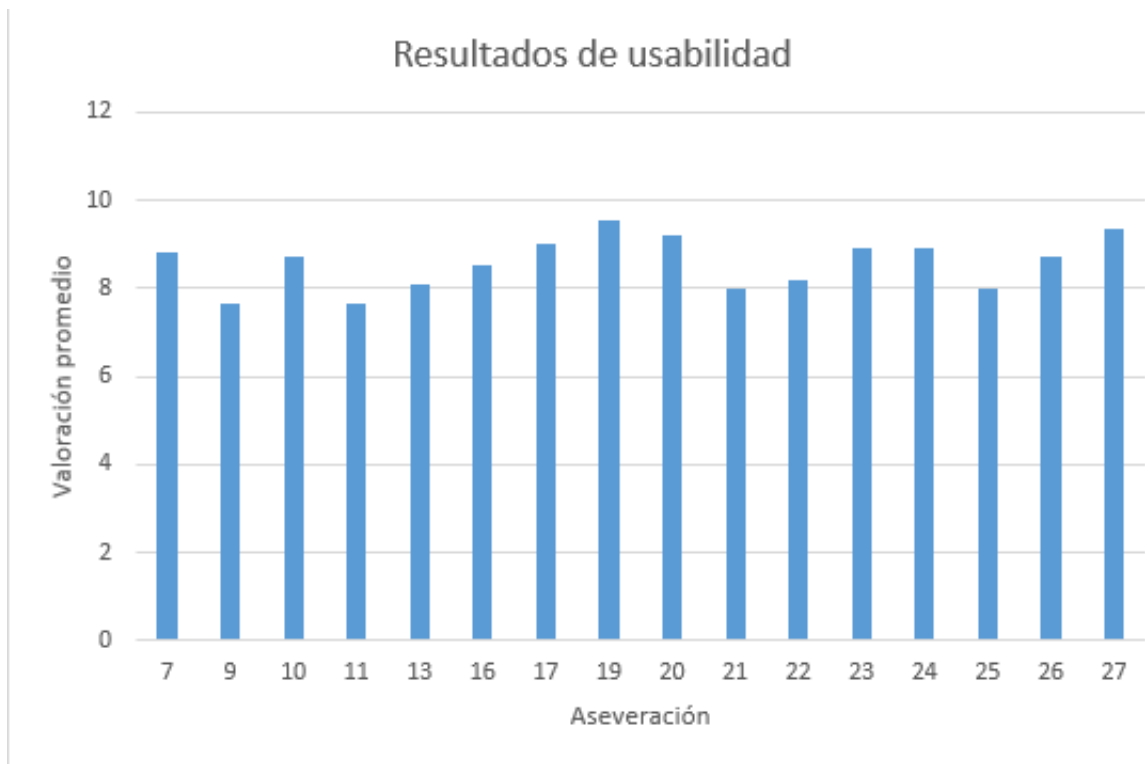


Figura 24: Resultados promedio de la evaluación sobre las aseveraciones de usabilidad

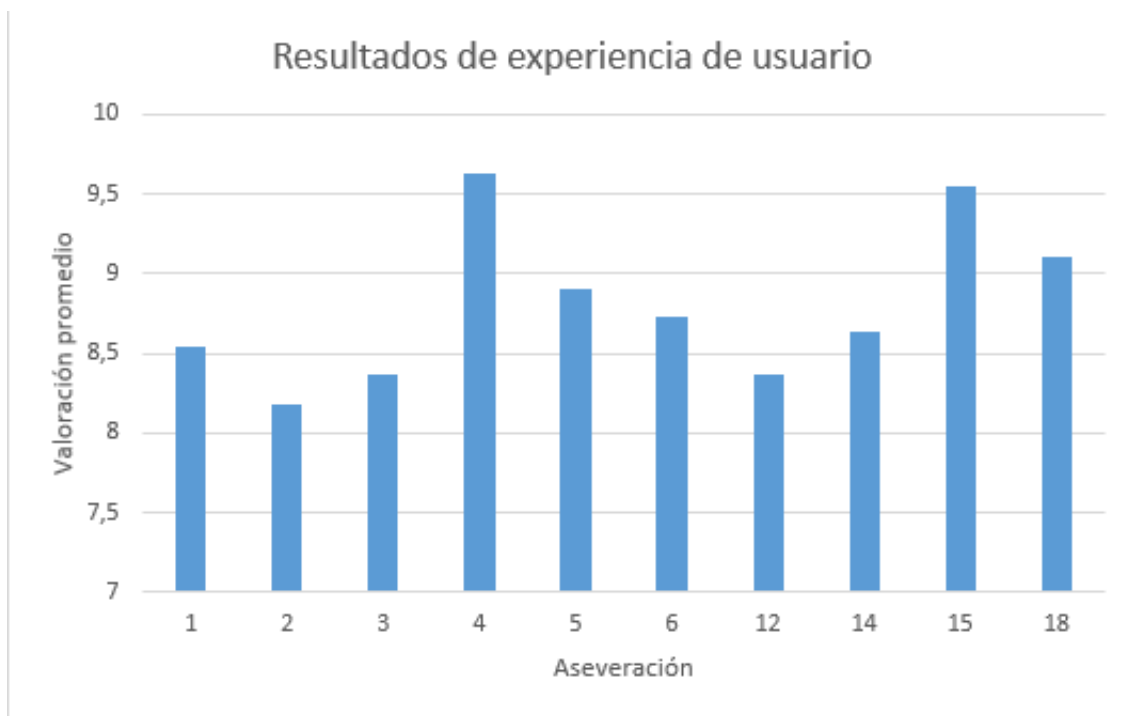


Figura 25: Resultados promedio de la evaluación sobre las aseveraciones de experiencia de juego

8.4. Análisis

El criterio de aceptación que definimos para el análisis de resultados fue una valoración mínima de 7,5. Las aseveraciones valoradas sobre este valor se considerarán logrados, y bajo este valor serán considerados no logrados. A pesar de que valoraciones bajo 7,5 no tienen porqué señalar que dicho aspecto no se logró, será considerado de esa manera ya que se esperaba un buen rendimiento del videojuego por nuestro enfoque en su usabilidad y jugabilidad.

El videojuego tuvo una valoración general promedio de 8,6 lo que significa que hubo una buena aceptación de los usuarios respecto a este, y por lo tanto un buen entendimiento de este y una buena experiencia de juego en promedio. Este valor supera por bastante el criterio de aceptación de 7,5, pero aún así puede ser mejorado con ciertos cambios sobre la interfaz y el videojuego en general.

Para profundizar en el análisis revisaremos detalladamente los resultados de las aseveraciones más relevantes según nuestro propio criterio. Habrá una análisis de los resultados de usabilidad y otro de la jugabilidad (*gameplay*).

8.4.1. Análisis de resultados de usabilidad

Dentro de lo que es usabilidad, el videojuego obtuvo una valoración general de 8,6, lo que nos indica que los usuarios pudieron entender el funcionamiento de la interfaz, cómo interactuar con ella y los objetivos del videojuego.

Dado nuestro criterio de aceptación, el videojuego cumple con lo mínimo para que la aplicación pueda ser usable por los usuarios ciegos, pero antes de dar el veredicto revisaremos más a fondo aspectos puntuales de ciertas aseveraciones de usabilidad.

La valoración mínima obtenida fue en las aseveraciones 9 y 11, con un valor de 7,6, las que corresponden respectivamente a *Me sentí controlando las situaciones del videojuego* y *El videojuego es fácil de usar*.

Si bien estas preguntas son parte de la usabilidad del videojuego, observaciones realizadas a los jugadores durante la evaluación de usabilidad y sus propios comentarios sobre el videojuego indican que estas aseveraciones se ven directamente afectadas por la experiencia de juego y el entendimiento de la lógica geométrica.

La totalidad de los usuarios pudieron desenvolverse sin ningún problema a través de

los distintos menús de opciones del videojuego, pero no así durante el plano cartesiano. En los módulos geométricos (la vista principal del videojuego), solo 4 de 11 usuarios tuvieron dificultades para interactuar con la interfaz, la que consiguieron usar eficientemente luego de unos minutos de interacción con el videojuego.

Con esto, volviendo a las aseveraciones 9 y 11, los usuarios pensaron que el videojuego no era fácil de usar y que perdían el control aún cuando interactuaban con la aplicación sin problemas, debido principalmente a la frustración de no entender el problema geométrico ni las coordenadas del plano cartesiano. Por lo anterior, a pesar de tener una aceptación al margen de lo mínimo requerido, las valoraciones a un nivel práctico de estas aseveraciones son mayores de las aparentes.

Las dos siguientes valoraciones más bajas se obtuvieron en las aseveraciones 21 y 25, *Entiendo los menús del videojuego* y *Entiendo lo que debo hacer en cada parte del videojuego*. Solo 4 de los usuarios no fueron capaces de entender los menús que se le presentaban, debido a que no escucharon lo que el narrador les estaba diciendo al momento de comenzar a utilizar el videojuego. Al no escuchar el narrador, no entienden que tienen un menú en frente, ni tampoco el objetivo del menú.

En los menús del videojuego, esto se solucionó señalando a los jugadores que escuchan el narrador con atención y que no se apresuren en interactuar con la interfaz puesto que aún no saben cómo utilizarla y estaban silenciando las repeticiones del narrador. En 1 de estos 4 casos, un usuario, a pesar de escuchar atentamente el narrador y entender el objetivo del menú, no entendió como interactuar con la interfaz puesto que no comprendió la instrucción de “tocar dos veces la pantalla”, este la reconoció como “tocar la pantalla con 2 dedos”. Aquí podemos ver la importancia del narrador y el lenguaje que este utiliza.

Según las observaciones de la interacción de los usuarios con la interfaz, un aproximado del 50 % de los usuarios pensó que los módulos geométricos también caen dentro de la categoría de menú. Es por ello que, el no entender la componente principal de juego y sus problemas geométricos, hacía creer a los jugadores que no estaban entendiendo los menús ni sus objetivos. En este caso tenemos un uso de lenguaje técnico que los usuarios no tienen porqué manejar, que también afectan los resultados de la evaluación

La siguiente aseveración peor evaluada es la número 13 “El videojuego se adapta a mi ritmo” con un valor de 8,1. Los jugadores con resto visual comúnmente podían jugar sin ningún problema, avanzando a una velocidad bastante rápida, lo que hacía que el juego fuera lento para ellos. Los niños con menos visión señalaron que el juego tenía una velocidad adecuada.

La aseveración 22 “Puedo identificar claramente las opciones de los menús” cae dentro del mismo problema que la aseveración 21: los usuarios veían el videojuego constante-

mente como un menú con opciones dado la falta de conocimiento de lenguaje técnico. Esta obtuvo una valoración de 8,2, lo cual resulta ser más que suficiente según el criterio de aceptación, pero hay que recalcar que este dato no está limpio del todo puesto que sus peores calificaciones aludían al no entendimiento de la lógica de los desafíos geométricos y de cómo resolverlos.

Para terminar el análisis de usabilidad, revisaremos a las aseveraciones 24 y 27 que resultan ser las más importantes de las que aún no se han comentado.

La aseveración 24 "Puedo identificar cuando paso de un menú a otro" alude a cuando el usuario cambia de contexto a uno diferente. Este cambio de contexto trae consigo un cambio en el menú, un cambio en los objetivos y un cambio en las opciones, por lo que es sumamente importante que el jugador esté consciente de dicho cambio para prestar atención al nuevo contexto al que se enfrenta.

Esta aseveración tuvo una valoración de 8,9, con lo que el criterio de aceptación se supera con un muy bien resultado. La valoración más baja que tuvo esta aseveración fue un 1, por el usuario C (Ver el anexo). A pensar de que este usuario valoró esta aseveración de usabilidad con la nota mínima (en conjunto con la 22), este no tuvo problemas para utilizar la interfaz ni para darse cuenta del cambio de menú ni sus opciones, con lo que nuevamente esta valoración en la práctica es más alta de lo que aparenta.

Una de las aseveraciones más importante del análisis de usabilidad de esta memoria es la 27 "Es posible jugar sin un asistente, la segunda aseveración mejor valorada, obteniendo una valoración de 9,4. Dado el puntaje de aceptación, esta aseveración consiguió un excelente resultado, pero aquí hay un gran detalle que es importante señalar antes de simplemente quedarnos con este valor.

Si bien la valoración más baja de esta aseveración fue de 7, debemos señalar que el contexto en el cual se elaboró este análisis de usabilidad resulta ser un tanto beneficioso para el videojuego evaluado. Esto es debido a que, dado que se requería que los usuarios utilizaran la interfaz para poder evaluarla de una forma más completa, esos fueron orientados en puntos claves de la aplicación en donde encontraron dificultades para retomar el juego de buena manera.

Aunque esta orientación haya sido mínima, marca la diferencia frente a un usuario que puede utilizar el videojuego frente a uno que no. Esta orientación se realizó mediante mensajes como: "debes escuchar atentamente al narrador", "presta atención a lo que el narrador está solicitando", "¿qué cuadrícula te solicitó encontrar el narrador?", "¿qué significa tocar la pantalla dos veces seguidas?". Todas estas intervenciones del especialista disminuyen la frustración y estrés del jugador, lo que hace que no pierda el control ni el enfoque sobre el videojuego, permitiéndole continuar avanzando de forma "autónoma".

También hubo otras intervenciones del especialista que permitieron que el jugador pudiera continuar utilizando el videojuego. Estas acciones fueron principalmente tres:

- Re-abrir el videojuego cuando el usuario salió de este regresando a la interfaz del sistema operativo por tocar erróneamente uno de los botones del teléfono móvil.
- Reiniciar el videojuego cuando ciertos usuarios, apresuradamente, accedían desde el menú principal al modo historia del videojuego antes de saber cómo utilizarlo.
- Explicar, con otras palabras, el problema geométrico del videojuego y orientar al jugador con la resolución dicho problema.

Dado todo lo anterior, volviendo a la aseveración 27 “Es posible jugar sin un asistente“, debemos cuestionar qué tan realista es este resultado, considerando que deseamos que el videojuego sea utilizado sin ningún tipo de soporte adicional más que los recursos propios del juego. Pero a pensar de ello, puesto que las intervenciones fueron mínimas, referente a casos borde y los problemas geométricos, se considera que la valoración de 9,4 obtenida para esta aseveración representa una buena calidad de la interfaz diseñada en este videojuego.

Durante la evaluación de usabilidad también se realizaron una serie de preguntas abiertas, las que, junto con las observaciones de la interacción del usuario, dieron origen a un nuevo diseño de la aplicación, las que se comentarán en el punto siguiente.

A continuación revisamos algunas respuestas relevantes sobre las preguntas abiertas a la evaluación de usabilidad.

- Pregunta 1: ¿Qué te gustó del videojuego?

Un total 6 jugadores señalaron explícitamente el agrado con la interacción de entidades durante la resolución de problemas geométricos. “Dirigir la nave”, “salvar planetas”, “tocar y esquivar los asteroides” fueron algunas de las respuestas de los jugadores.

8 jugadores mencionaron satisfacción completa por el videojuego, aludiendo principalmente a los sonidos, a la entretención que genera y lo desafiante que es.

Cabe mencionar que a pesar de que algunos mencionaron aprobación global con el videojuego, mencionaron ciertos aspectos que se pueden mejorar. Esto se comenta en la pregunta siguiente.

- Pregunta 2: ¿Qué no te gustó del videojuego?

De los 11 usuarios, 5 de ellos hicieron referencia a componentes gráficas del videojuego: las imágenes y los colores. Afortunadamente, estos comentarios sobre la gráfica del videojuego no fue por una componente de usabilidad, sino más bien una netamente estética acorde a los gustos de cada jugador.

Otras de las respuestas mencionadas fue referente a los narradores utilizados. Tres usuarios estuvieron de acuerdo en que la voz masculina se entendía menos que

la femenina, tanto a nivel de pronunciación como de volumen. Otros dos acordaron que en ocasiones no se entendía la voz femenina a pesar de que tenía un volumen adecuado.

También, dentro de esta respuesta, 3 usuarios mencionaron no entender en ocasiones lo que había que hacer. Respuestas que se relacionan con lo comentado durante el análisis de las aseveraciones de usabilidad evaluadas: los usuarios no entendían los problemas geométricos, por tanto, tampoco cómo enfrentarlos a través de la interfaz.

A pesar de que solo 3 jugadores hicieron este último comentario, las observaciones indican que aproximadamente un 75 % de los usuarios tuvo dificultades para entender uno o más de los problemas geométricos y cómo resolverlos.

Esta dificultad para entender los problemas geométricos fue principalmente al poco y nulo conocimiento sobre el plano cartesiano y las coordenadas cartesianas que tenían los usuarios. Los usuarios entendían cómo interactuar con las coordenadas, cómo consultarlas y cómo realizar una acción, pero no entendían la componente geométrica/matemática. Por ejemplo, no entendían qué significa cuando el narrador se refiere a “la cuadrícula $(-2, 0)$ ”. Dos de los usuarios evaluados comentaron no saber lo que era un número negativo.

- **Pregunta 3: ¿Qué agregarías al videojuego?**

La mitad de los participantes señaló que no agregarían nada al videojuego o bien no hicieron comentarios al respecto, ya que estuvieron muy satisfechos con lo jugado.

Tres usuarios señalaron que les gustaría que el videojuego tuviera más variedad de gráficos, personajes y sonidos, es decir, un mayor atractivo a nivel de experiencia de juego. Las respuestas fueron “poner planetas de fondo”, “distintos mundos, aspectos y personajes” y “más sonidos”.

- **Pregunta 4: ¿Para qué crees que te puede servir el videojuego?**

Esta pregunta obtuvo respuestas bastante variadas, que dejaron en claro la aprobación del videojuego por parte de los usuarios de la evaluación de usabilidad.

Tres jugadores señalaron que el juego puede servir para compartirlo con otras personas y entretenerse en conjunto a pesar de no tener una componente multijugador. Cinco comentaron que el videojuego sirve directamente para divertirse y otros cinco mencionaron que el videojuego sirve para aprender matemática, aunque ninguno de ellos utilizó el término “geometría”.

Una respuesta notable fue la del usuario D (ver anexo), el que respondió “para la concentración, enfocarse y tranquilizarse”. Durante el diseño se pensó que la dificultad de los problemas fueran tal que el usuario deba enfocarse casi por completo en el videojuego para resolverlos.

Todos estos resultados mencionados serán detallados en el anexo de esta memoria, al final de este documento.

8.4.2. Análisis de resultados de experiencia de juego

La valoración promedio obtenida para las aseveraciones relacionadas con experiencia de juego fue de 8,8 de un total de 10. Tal como los resultados de usabilidad, este valor supera por bastante el criterio de aceptación utilizado.

A diferencia de los resultados de usabilidad, la valoración mínima obtenida para las aseveraciones de experiencia de usuario tuvo un valor de 8,2. Esto nos señala que el videojuego tuvo una aceptación de la jugabilidad aún mayor que la de usabilidad.

La aseveración con el valor de 8,2 corresponde a “El videojuego es entretenido”, el que se refiere directamente a la experiencia de juego, que es la encargada de generar motivación y entretenimiento en el jugador. El hecho de los jugadores hayan considerado el juego entretenido luego de haber estado jugando entre 10 y 30 minutos nos dice de manera directa que el *software* ha resultado ser satisfactorio en este ámbito.

Para determinar el nivel de inmersión logrado por el videojuego se utilizó la aseveración 4 “El videojuego me hace estar activo”. El hecho de que el usuario esté activo refiere a que esté concentrado y enfocado en lo que sucede dentro del espacio virtual del videojuego.

El hecho de que los usuarios estuvieran activos y concentrados durante la sesión de juego proviene principalmente de la experiencia de juego que este presenta, y la gran concentración que requerían los problemas geométricos para ser resueltos.

El usuario A (ver anexo), a pesar de tener más dificultades con el uso del videojuego en comparación a sus pares, evaluó la aseveración mencionada con valor 8, correspondiente al valor más bajo de todos los usuarios. El videojuego requiere de cierta concentración para ser jugado, y la experiencia de juego que este entrega hizo que los usuarios hayan estado dispuestos a enfocarse en el videojuego.

Las dos componentes mejor evaluadas del videojuego respecto a la experiencia de usuario fueron las imágenes y los sonidos, pero por sobre todo estos últimos. La aseveración 15 “Me gustan los sonidos del videojuego” tuvo una valoración promedio de 9,6 mientras que la 18 “Me gustan las imágenes del videojuego” un promedio 9,1.

Dentro de las opiniones de los usuarios, se recalcó en varias ocasiones el gusto por los sonidos del videojuego y la ambientación que estos generaban. Sin lugar a dudas los sonidos de interfaz y efectos de sonidos fueron unos de los principales responsables de la gran experiencia de juego que presenta el videojuego.

Cabe destacar que el videojuego resultó ser atractivo incluso para usuarios que fre-

cuentemente suelen utilizar consolas de videojuego, dejándonos claro que el videojuego puede ser utilizado sin problemas para la entretención.

8.5. Correcciones de interfaz de usuario y jugabilidad

Dadas las observaciones y comentarios obtenidos de las evaluación de usabilidad y jugabilidad, se realizaron los siguientes cambios a la interfaz del videojuego.

- Interfaz funciona con múltiples dedos
Al comienzo la interfaz para ciegos solo funcionaba cuando el jugador tocaba la pantalla con un único dedo. Esto fue corregido para que la pantalla funcione con cualquier cantidad de toques en pantalla para evitar el bloqueo involuntario de la pantalla por parte del jugador.
Algunos usuarios tocaban la pantalla directamente con más de un dedo para utilizarla, otros la tocaban más de una vez con su segunda mano mientras afirmaban el teléfono móvil.
- Primera narración obligatoria:
Dado que el narrador es voluntario, como se mencionó durante el diseño del videojuego, algunos jugadores, apresuradamente, lo silenciaban antes de escuchar la primera explicación del narrador sobre la interfaz y los objetivos del menú inicial. Es por ello que la primera narración de cada menú pasó a ser obligatoria y solo las sucesivas pueden ser silenciadas.
- Límite del menú deslizable:
En un comienzo la sensibilidad del menú deslizable era bastante poca, los jugadores deslizaban su dedo en la pantalla pero no era suficiente para navegar a la opción adyacente. Esta sensibilidad se incrementó consiguiendo de inmediato que los usuarios pudieran utilizar el menú de manera más eficiente.
- Botón “atrás” de Android:
Por defecto, el botón “atrás” de Android hace que la aplicación actual quede en segundo plano volviendo a la actividad anterior del sistema operativo. Esto ocasionaba que los jugadores salieran del videojuego, incluyendo aquellos que sabían utilizar bien el botón y deseaban volver al menú anterior. Esto se corrigió durante la evaluación de usabilidad.
- Botón “volver” versus “salir”:
Algunos usuarios entendían que el botón “salir” del menú durante los módulos geométricos era para salir del menú, sorprendiéndose ya volvían al menú principal cerrando el avance actual de juego. Este botón se mantuvo pero se cambió su texto y mensaje narrado a “salir al menú principal”.
- Bordes del plano cartesiano:
Al jugar los desafíos geométricos, los usuarios con menos visión no sabían cuando estaban al borde del plano cartesiano, en otras palabras, en el límite de la pantalla.

Esto ocasionaba que toquen involuntariamente los botones del sistema operativo cerrando el videojuego, o bien el borde de la pantalla en donde no hay interacción alguna.

Para corregir esto se añadió sonidos de “colisión” al llegar a los bordes del plano cartesiano, lo que tuvo muy buenos resultados.

- Menú oculto durante módulos geométricos

El menú oculto de los módulos geométricos se cierra de forma automática y el *feedback* que lo señalaba no era claro. Del mismo modo, cuando los usuarios abrían este menú, los usuarios no se daban cuenta que la interfaz había cambiado.

Esto se corrigió mejorando el *feedback* mediante el uso del narrador, el que ahora señala explícitamente cuando este menú se abre y se cierra.

- Volumen de sonido de acción en módulos geométricos:

Para informar al jugador que el videojuego está listo para recibir sus acciones durante los desafíos geométricos, se utiliza un sonido de motor de la nave espacial que el narrador se encarga de explicar durante el tutorial.

Este sonido poseía un volumen demasiado bajo que impedía en ocasiones a los jugadores escucharlo. El volumen de este sonido fue incrementado y los usuarios que lo probaron recalcaron una atracción por este sonido que antes no se escuchaba.

- Tiempo de re-narración de objetivo:

La componente más difícil del videojuego, según el análisis de usabilidad, es entender el problema geométrico que se expone y el objetivo de lo que se debe realizar para solucionarlo.

Por esta razón los jugadores tardaban bastante tiempo pensando qué era lo que debían hacer, tiempo en el cual olvidaban también lo mencionado por el narrador. Es por ello que se redujo el tiempo de la re-narración del objetivo de 1 minuto a 30 segundos, lo que entregó muy buenos resultados para los usuarios posteriores a este cambio, ya que antes de comenzar a frustrarse por no saber qué hacer, el narrador volvía a explicar el problema.

- Soporte del problema geométrico:

El videojuego, como se mencionó en el diseño, trae dos opciones para dar soporte al usuario referente a los problemas geométricos que enfrenta. Dado que muchos jugadores no sabían resolver ciertos problemas geométricos, además de la re-narración de objetivo, cuando el jugador realiza demasiadas acciones incorrectas el narrador señala al usuario cómo utilizar dicho soporte.

- Superposición de voces y sonidos:

En la primera versión del videojuego, la voz del narrador se superponía con una segunda voz ocasionada por la interacción de usuario con la interfaz, ocasionando que ninguna de las dos voces fuera escuchada.

Por este motivo, cuando un narrador habla, se desactivan las demás opciones para que el jugador no pueda accionar un segundo narrador y pueda escuchar el que ya estaba hablando sin ninguna interferencia.

Esto pareciera ser una desventaja en ciertos casos, pero recalcamos que el narrador puede ser anulado tocando la pantalla dos veces seguidas, lo que retomará la

interacción con el resto de la interfaz permitiendo al jugador realizar cualquier otra acción.

8.6. Conclusiones

8.6.1. Resumen

A pesar del buen resultado de evaluación obtenido, que entregó una valoración de 8,6 de un total de 10, y el buen desempeño que consiguió el videojuego a nivel de interfaz de usuario, el análisis de usabilidad ha dejado en evidencia diversos aspectos de diseño que necesitan ser mejorados en este videojuego, y que es importante considerarlos en futuros trabajos de ingeniería de *software* para niños ciegos.

No solo ha quedado en evidencia que el diseño del videojuego de esta memoria debe ser mejorado, sino que además se deben mejorar las herramientas de evaluación de usabilidad y jugabilidad utilizadas puesto que en ocasiones resultan ser imprecisas por depender de diversas variables que no están consideradas dentro de estas mismas.

8.6.2. Objetivos alcanzados y no alcanzados

El principal objetivo de esta memoria era diseñar e implementar un videojuego educativo para ciegos que pueda ser utilizado de forma autónoma, tal como se menciona al comienzo de este documento.

Los resultados de usabilidad señalan que los usuario pueden utilizar el videojuego por sí solos, aseveración con una valoración promedio de 9,4 de un máximo 10. A pesar de esto, un análisis más profundo señaló que los usuarios efectivamente pueden hacerlo debido al contexto de esta investigación, pero que en la práctica pueden ocurrir complicaciones que los llevarían a no poder utilizar el videojuego. Afortunadamente, este primer resultado es un gran paso puesto que demuestra que el usuario entendió el videojuego y su interfaz.

El segundo objetivo de esta memoria hace referencia a la jugabilidad del videojuego, es decir, si este genera la motivación y entretenimiento suficiente en los jugadores para que continúen jugando. Este aspecto obtuvo una valoración de 8,8 sobre 10 en base a un criterio de aprobación de 7,5. Esta valoración es alta a pesar de la dificultad que trae la falta de formación en matemática y geometría en los participantes y la frustración que esto generó durante las sesiones de juego.

8.6.3. Aspectos a mejorar y trabajo futuro

Uno de estos aspectos es que no todos los usuarios quieren escuchar el narrador, los jugadores se apresuran, ya sea por ansiedad, porque lo encuentran aburrido o por extroversión. Algunas de las cosas que podría realizarse sería generar un narrador más atractivo, con una fuerte personificación e interpretación de un personaje ficticio, atractivo para cualquier usuario, tal como se hace en las películas. Un narrador al cual el usuario quiera escuchar, uno que entregue una mayor experiencia de juego.

Siguiendo con la lógica del narrador, otro aspecto a mejorar es la forma de cómo se silencia este narrador, ya que algunos usuarios silenciaban el narrador por error mediante el uso constante del gesto principal para realizar una acción justo en el instante en que este comenzaba a repetir la instrucción. Investigaciones futuras podrían incluir el uso de gestos en pantalla más completos para realizar este tipo de acciones.

Otros de los aspectos débiles de este videojuego fueron volumen y la pronunciación de los narradores, que en ocasiones causaba que los usuarios no entendieran el objetivo del videojuego, disminuyendo notablemente la usabilidad de la aplicación. Además de esto debemos señalar que el narrador no debe exponer demasiada información en un corto periodo de tiempo, menos si expone múltiples datos que los usuarios deben recordar ya que terminan siendo olvidados en el corto plazo.

El lenguaje de un videojuego también es un gran aspecto a considerar para el diseño de usabilidad. Algunos usuarios no entienden la diferencia entre “tocar dos veces la pantalla” y “tocar la pantalla con dos dedos”, lo cual nos indica que debemos enfocarnos en utilizar un lenguaje más exacto para obtener interfaces más usables. Además, las palabras “volver”, “salir” y “atrás” en la interfaz han causado confusión.

De lo anterior surgen preguntas como: ¿es posible utilizar un lenguaje único y global que permita a los usuarios de la misma lengua entender lo que la interfaz quiere expresar? Parece ser una tarea muy difícil la que requiere de investigación futura para ser respondida.

Uno de los aspectos más críticos resultó ser el cómo se vio afectada la usabilidad del videojuego debido a la dificultad de los problemas geométricos que los usuarios enfrentaban. Aquí se debe señalar cómo la frustración frente a no entender el problema puede afectar al usuario en la realización de otras tareas dentro de la interfaz de una aplicación.

Siguiendo con lo anterior, un trabajo que queda pendiente es responder si los problemas geométricos de este videojuego son demasiado complejos, o si bien la interfaz es demasiado débil y debemos encontrar mejores formas de exponer dichas problemáticas.

El trabajo futuro que ha originado esta memoria referente a usabilidad y jugabilidad es el siguiente:

1. Mejorar las herramientas de evaluación de usabilidad y jugabilidad para que sean más precisas, evitando que se vean afectadas por otros aspectos como el uso de lenguaje técnico o la dificultad de la lógica de los problemas del *software*.
2. Continuar con la evaluación de usabilidad del videojuego, corrigiendo todos los aspectos comentados con anterioridad para que los usuarios logren jugar, en un espacio cotidiano y espontáneo, de manera independiente.
3. Expandir la evaluación de usabilidad y jugabilidad a la totalidad del videojuego, ya que solo se realizó en la versión tutorial de este. Esta evaluación permitirá extraer más datos relevantes para el futuro de este videojuego y otras aplicaciones.
4. Analizar futuras tecnologías e interfaces que sean más aptas para exponer problemas geométricos dentro de un plano cartesiano a los jugadores ciegos. Una de ellas son las pantallas con relieve dinámico las cuales aún no salen al mercado, las que pueden estar disponibles dentro de unos años.
5. Realizar la evaluación de impacto correspondiente de este videojuego para analizar el impacto cognitivo que genera el videojuego en los usuarios que lo utilicen.

Bibliografía

- [1] A., Michael y Mariella B.: *Making Geometry Meaningful to Blind Students*. Journal of Theory and Practice in Education, 2011.
- [2] Aedo, Raúl R. Fernández: *Aprendizaje con el uso de nuevas tecnologías*. Revista Iberoamericana De Educación, 25(1), 1-9, 2001.
- [3] AudioGames: *AudioGames.net*. <https://audiogames.net/>, visitado el 2019-08-20.
- [4] Bermejo, F., V. Lunati y C. Arias: *Audiojuego con sonidos envolventes: una experiencia preliminar con personas ciegas y con visión normal*. Investigaciones en Discapacidad. 5. p71-80, 2016.
- [5] Blitab: *Tablet táctil*. <https://blitab.com/>, visitado el 2019-08-20.
- [6] Buraga, Sabin: *THCI 2018 Natural User Interfaces. Ubiquitous Computing*. 2018.
- [7] C Connors, Erin, Lindsay A Yazzolino, Jaime Sánchez y Lotfi Merabet: *Development of an Audio-based Virtual Gaming Environment to Assist with Navigation Skills in the Blind*. Journal of visualized experiments : JoVE, 73, Marzo 2013.
- [8] CADEM: *El Chile que viene*, 2018. <https://www.cadem.cl/encuestas/el-chile-que-viene-abril-2018/>, visitado el 2019-09-05.
- [9] Carrie, Heeter, Winn Brian, Winn Jillian y Bozoki Andrea: *The Challenge of Challenge: Avoiding and Embracing Difficulty in a Memory Game*. páginas 1–3, 2008.
- [10] Collinge, Robyn: *How to Design for Color Blindness*, 2017. <https://usabilla.com/blog/how-to-design-for-color-blindness/>, visitado el 2019-08-20.
- [11] Csikszentmihalyi, Mihaly: *Flow: The Psychology of Optimal Experience*. 1990.
- [12] Darin, T. y J. Sánchez: *Dimensions to Analyze the Design of Multimodal Videogames for the Cognition of People Who Are Blind*. 2015.
- [13] D'Atri, E., C. Medaglia, E. Panizzi y A. D' Atri: *A system to aid blind people in the*

mobility: A usability test and its results. 2nd International Conference on Systems, ICONS 2007, 2007.

- [14] Dowino: *A blind legend*. <http://www.ablindlegend.com/en/home-2/>, visitado el 2019-08-20.
- [15] Freitas, Diamantino y Georgios Kouroupetroglou: *Speech technologies for blind and low vision persons*. Technology and Disability, páginas 135–156, 2008.
- [16] Gnome: *Orca*. <https://wiki.gnome.org/Projects/Orca>, visitado el 2019-08-20.
- [17] Goldreich, Daniel y Ingrid M. Kanics: *Tactile Acuity is Enhanced in Blindness*. The Journal of neuroscience : the official journal of the Society for Neuroscience, 23 8, 2003.
- [18] Google: *TalkBack*. <https://www.androidcentral.com/what-google-talk-back>, visitado el 2019-08-20.
- [19] J., Sánchez: *Pauta de Evaluación de Usabilidad de Usuario Final*. Asignatura Taller de Interacción Humano-Computador CC6501. Carrera de Ingeniería Civil en Computación, Universidad de Chile, 2015.
- [20] Lahav, O. y D. Mioduser: *Multi-sensory virtual environment for supporting blind persons' acquisition of spatial cognitive mapping, orientation, and mobility skills*. 2000.
- [21] Lahav, O., D.W. Schloerb, S. Kumar y M.A. Srinivasan: *Aid: A Learning Environment for Enabling People Who are Blind to Explore and Navigate Through Unknown Real Spaces*. Pro-ceedings Virtual Rehabilitation 2008 Conference, Vancouver, Canada, pp.193- 197, 2008.
- [22] Lynch, Kevin: *The Image of The City*. 1971.
- [23] Mccrindle, Rachel y D Symons: *Audio space invaders*. Proceedings of the Third International Conference on Disability, Virtual Reality and Associated Technologies, ICDVRAT 2000, Enero 2000.
- [24] Merabet, Lotfi y Jaime Sánchez: *Development of an Audio-Haptic Virtual Interface for Navigation of Large-Scale Environments for People Who Are Blind*. 9739:595–606, Julio 2016.

- [25] Naumann, Anja: *Intuitive Use of User Interfaces: Defining a Vague Concept*. p128-136, 2007.
- [26] Nielsen, J.: *Usability 101: Introduction to Usability*. All Usability, 9:1–10, Enero 2003.
- [27] Nikolakis, G., D. Tzovaras y M. Strintzis: *Object Recognition For The Blind*. 13th European Signal Processing Conference (EUSIPCO2005), Antalya, Turkey, September 2005, pp. 1-4, 2005.
- [28] O., Carmen Molina: *Integración Del Invidente En La Clase De Matemáticas. Agenda De Investigación Desde La Teoría De Las Situaciones Didácticas*. <https://www.ugr.es/~jgodino/siidm/cangas/invidentes.htm>, visitado el 2019-08-20.
- [29] OMS: *Organización Mundial de la Salud*. <https://www.who.int/countries/chl/es/>, visitado el 2019-08-20.
- [30] Pardo, M. y J. Sánchez: *Videojuego Educativo Para El Aprendizaje De Geometría En Niños No Videntes*. 2017.
- [31] Prensky, Marc: *The Motivation of Gameplay or, the REAL 21st century learning revolution*. On the Horizon. 10. p1-p10, 2002.
- [32] Quick, John M., Robert K. Atkinson y Lijia Lin: *The Gameplay Enjoyment Model*. International Journal of Gaming and Computer-Mediated Simulations, 4:64–80, 2012.
- [33] R., Raúl y Pedro M.: *El Aprendizaje Con El Uso De Las Nuevas Tecnologías De La Información Y Las Comunicaciones*. Revista Iberoamericana de Educación, 25(1):1–9, ene. 2001. <https://rieoei.org/RIE/article/view/2912>.
- [34] Red, Formación en: *Educación inclusiva. Personas con discapacidad visual*. http://www.ite.educacion.es/formacion/materiales/129/cd/unidad_4/m4_matematicas.htm, visitado el 2019-08-20.
- [35] Rocha, A. y J. Sánchez: *OM Virtual Environments For End-users Who Are Blind: A Systematic Review*. 2018.
- [36] Roth, R.: *User Interface and User Experience (UI/UX) Design*. Geographic Information Science & Technology Body of Knowledge, 2017.

- [37] Scientific, Freedom: *Jaws*. <https://www.freedomscientific.com/Products/Blindness/JAWS>, visitado el 2019-08-20.
- [38] Sánchez, J.: *User Centered Technologies for Blind Children*. Human Technology Journal, 45(2), November 2008, pp. 96-122, 2008.
- [39] Sánchez, J. y H. Flores: *AudioMath: blind children learning mathematics through audio*. Virtual Reality & Assoc. Tech. 4., 2004.
- [40] Sánchez, J. y M. Lumbreras: *Ambientes Interactivos con Sonido Espacializado para Estimular la Cognición de Niños Ciegos*. 2019.
- [41] Sánchez, J. y M. Sáenz: *La granja de Theo y Seth, aprendizaje de las matemáticas en niños con discapacidad visual a través de audio*. 2003.
- [42] Sánchez, J. y M. Sáenz: *AudioChile: Navegación y cognición a través de espacios 3D para niños ciegos*. 2004.
- [43] Sánchez, J. y A. Tadres: *Audio and Haptic Based Virtual Environments for Orientation and Mobility in People Who are Blind*. Proceedings of the 12th International ACM SIGACCESS Conference on Computers and Accessibility, ASSETS 2010, Orlando, FL, USA, 2010.
- [44] Sánchez, Jaime: *Aprendizaje visible, tecnología invisible*. Enero 2001.
- [45] Sánchez, Jaime: *Three-Dimensional virtual environment for blind children*. Cyber-Psychology & Behavior, 9:200–206, Abril 2006.
- [46] Sánchez, Jaime: *Audiomemorice: Desarrollo de la memoria de niños con discapacidad visual a través de audio*. Octubre 2019.
- [47] Sánchez, Jaime, Paola Alarcón y Héctor Flores: *Diseño Centrado en el Usuario de una Herramienta para que Usuarios no Videntes Construyan Mapas Conceptuales*. Enero 2006.
- [48] Sánchez, Jaime, Nelson Baloian, Tiago Hassler y Heinz Ulrich Hoppe: *AudioBattleship: Blind Learners Collaboration through Sound*. páginas 798–799, Enero 2003.
- [49] Sánchez, Jaime y Ticciane Darin: *Videojuegos Multimodales para la cognición de*

personas ciegas: Tendencias y Cuestiones.

- [50] Sánchez, Jaime y Miguel Elías: *Science Learning in Blind Children through Audio-Based Games*, páginas 1–16. Octubre 2008.
- [51] Sánchez, Jaime y Zulma Guevara: *Impacto de la integración curricular de un videojuego en el desarrollo de habilidades cognitivas de orden superior*. Enero 2015.
- [52] Sánchez, Jaime y Tiago Hassler: *AudioMUD: A Multi-User Virtual Environment for Blind People*. Enero 2006.
- [53] Sánchez, Jaime y Mauricio Sáenz: *Assisting the mobilization through subway networks by users with visual disabilities*. Septiembre 2019.
- [54] Sánchez, Jaime, Mauricio Sáenz y Miguel Ripoll: *Usability of a multimodal videogame to improve navigation skills for blind children*. páginas 35–42, Enero 2009.
- [55] T., Dick y Kubiak E.: *Issues and Aids for Teaching Mathematics To the Blind*. Journal of Theory and Practice in Education, 1997.
- [56] Technologies, Novint: *Novint Falcon*. https://en.wikipedia.org/wiki/Novint_Technologies#Novint_Falcon, visitado el 2019-08-20.
- [57] Wikipedia: *Game Design Document*. https://en.wikipedia.org/wiki/Game_design_document, visitado el 2019-08-20.
- [58] Wikipedia: *Tiflotecnología*. <https://es.wikipedia.org/wiki/Tiflotecnología>, visitado el 2019-08-20.
- [59] Yu, W. y Brewster: *Comparing Two Haptic Interfaces for Multimodal Graph Rendering*. Proceedings of the 10th Symposium on Haptic Interfaces for Virtual Environment and Tele-operator Systems (Haptics 2002), Florida, USA, IEEE, pp. 3-9, 2002.

9. Anexo

9.1. Documentos de evaluación de usabilidad

| | Poco | | | | Mucho |
|---|------|---|-----|---|-------|
| | 1 | 2 | ... | 9 | 10 |
| Me gusta el videojuego | | | | | |
| El videojuego es entretenido | | | | | |
| El videojuego es desafiante | | | | | |
| El videojuego me hace estar activo | | | | | |
| Volvería a jugar con el videojuego | | | | | |
| Recomendaría este videojuego a otros niños/jóvenes | | | | | |
| Aprendí a jugar con este videojuego | | | | | |
| El videojuego tiene distintos niveles de dificultad | | | | | |
| Me sentí controlando las situaciones del videojuego | | | | | |
| El videojuego es interactivo | | | | | |
| El videojuego es fácil de usar | | | | | |
| El videojuego es motivador | | | | | |
| El videojuego se adapta a mi ritmo | | | | | |
| El videojuego me permitió entender nuevas cosas | | | | | |
| Me gustan los sonidos del videojuego | | | | | |
| Los sonidos del videojuego son claramente identificables | | | | | |
| Los sonidos del videojuego me transmiten información | | | | | |
| Me gustan las imágenes del videojuego | | | | | |
| Las imágenes del videojuego son claramente identificables | | | | | |
| Las imágenes del videojuego me transmiten información | | | | | |
| Entiendo los menús del videojuego | | | | | |
| Puedo identificar claramente las opciones de los menús | | | | | |
| La cantidad de botones y opciones es adecuada | | | | | |
| Puedo identificar cuando paso de un menú a otro | | | | | |
| Entiendo lo que debo hacer en cada parte del videojuego | | | | | |
| Sé cuando una acción es realizada o no lo es | | | | | |
| Es posible jugar sin un asistente | | | | | |

Figura 26: Preguntas de calificación del videojuego de la pauta de usabilidad y jugabilidad.

| |
|---|
| ¿Qué te gustó del videojuego? |
| ¿Qué no te gustó del videojuego? |
| ¿Qué agregarías al videojuego? |
| ¿Para qué crees que te puede servir el videojuego? ¿Qué otros usos le darías? |

Figura 27: Preguntas de desarrollo de la pauta de usabilidad.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|--|----|-----|----|-------------|
| Desliza el dedo para navegar el menú | | | | |
| Toca dos veces para seleccionar una opción | | | | |
| Toca una vez para consultar una opción | | | | |

Figura 28: Pauta de observación. Tarea 1: Reconocimiento de interfaz.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Reconoce el menú tutorial | | | | |
| Escoge la opción tutorial tocando dos veces | | | | |
| Escoge el tutorial completo tocando dos veces | | | | |

Figura 29: Pauta de observación. Tarea 2: Reconocimiento de tutorial.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Reconoce cuadrículas al tocarlas una vez | | | | |
| Realiza una acción tocando dos veces | | | | |
| Reconoce el menú oculto tocando con ambos dedos la pantalla | | | | |
| Selecciona el objetivo en el menú oculto tocando dos veces | | | | |
| Reconoce la cuadrícula (0,0) | | | | |
| Realiza una acción tocando la cuadrícula (0,0) | | | | |
| Reconoce que hay una nave espacial tocándola una vez | | | | |

Figura 30: Pauta de observación. Tarea 3: Reconocimiento del plano cartesiano y el objetivo.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Escoge la opción tutorial tocando dos veces | | | | |
| Escoge el tutorial de Rayo Láser tocando dos veces | | | | |
| Reconoce que hay un enemigo y lo busca | | | | |
| Encuentra al enemigo y lo toca dos veces | | | | |
| Busca la cuadrícula bajo el enemigo para disparar | | | | |
| Toca dos veces debajo del enemigo para disparar el rayo láser | | | | |

Figura 31: Pauta de observación. Tarea 4: Jugar tutorial de Rayo Láser.

9.2. Resultados de evaluación

La siguiente tabla muestra las valoraciones de cada usuario a cada una de las aseveraciones del documento de evaluación de usabilidad.

Los 9 usuarios con resto visual se identificaron desde la letra A hasta la J. Los usuarios con ceguera total corresponden a CA y CB.

| Aseveraciones | A | B | C | D | E | F | H | I | J | CA | CB | Promedio |
|--|----|----|----|----|----|----|----|----|----|----|----|----------|
| 1 Me gusta el videojuego | 7 | 7 | 5 | 10 | 10 | 9 | 10 | 10 | 9 | 8 | 9 | 8,55 |
| 2 El videojuego es entretenido | 8 | 8 | 4 | 10 | 9 | 8 | 10 | 7 | 9 | 8 | 9 | 8,18 |
| 3 El videojuego es desafiante | 5 | 1 | 7 | 10 | 10 | 10 | 10 | 10 | 10 | 9 | 10 | 8,36 |
| 4 El videojuego me hace estar activo | 8 | 10 | 10 | 10 | 10 | 9 | 10 | 10 | 9 | 10 | 10 | 9,64 |
| 5 Volvería a jugar con el videojuego | 8 | 10 | 4 | 10 | 10 | 9 | 9 | 10 | 9 | 10 | 9 | 8,91 |
| 6 Recomendaría este videojuego a otros r | 6 | 10 | 5 | 10 | 10 | 10 | 9 | 10 | 9 | 8 | 9 | 8,73 |
| 7 Aprendí a jugar con este videojuego | 10 | 10 | 7 | 10 | 10 | 9 | 9 | 10 | 5 | 8 | 9 | 8,82 |
| 8 El videojuego tiene distintos niveles de d | 4 | 10 | 5 | 10 | 10 | 10 | 6 | 6 | 9 | 5 | 3 | 7,09 |
| 9 Me sentí controlando las situaciones del | 6 | 10 | 3 | 9 | 10 | 8 | 9 | 8 | 6 | 8 | 7 | 7,64 |
| 10 El videojuego es interactivo | 7 | 10 | 5 | 10 | 10 | 7 | 10 | 10 | 10 | 9 | 8 | 8,73 |
| 11 El videojuego es fácil de usar | 10 | 10 | 7 | 6 | 9 | 7 | 7 | 8 | 8 | 5 | 7 | 7,64 |
| 12 El videojuego es motivador | 5 | 8 | 4 | 9 | 10 | 9 | 10 | 10 | 9 | 9 | 9 | 8,36 |
| 13 El videojuego se adapta a mi ritmo | 3 | 10 | 10 | 9 | 10 | 9 | 8 | 7 | 6 | 8 | 9 | 8,09 |
| 14 El videojuego me permitió entender nue | 1 | 10 | 10 | 10 | 10 | 10 | 8 | 10 | 9 | 9 | 8 | 8,64 |
| 15 Me gustan los sonidos del videojuego | 9 | 10 | 10 | 10 | 10 | 7 | 10 | 10 | 10 | 10 | 9 | 9,55 |
| 16 Los sonidos del videojuego son claram | 8 | 10 | 8 | 10 | 9 | 9 | 8 | 10 | 7 | 7 | 8 | 8,55 |
| 17 Los sonidos del videojuego me transmi | 9 | 10 | 2 | 10 | 9 | 9 | 10 | 10 | 10 | 10 | 10 | 9,00 |
| 18 Me gustan las imágenes del videojuegc | 8 | 10 | 4 | 10 | 10 | 10 | 10 | 10 | 10 | | | 9,11 |
| 19 Las imágenes del videojuego son clara | 10 | 10 | 7 | 10 | 10 | 9 | 10 | 10 | 10 | | | 9,56 |
| 20 Las imágenes del videojuego me transi | 10 | 10 | 6 | 10 | 10 | 9 | 10 | 9 | 9 | | | 9,22 |
| 21 Entiendo los menús del videojuego | 8 | 8 | 3 | 10 | 10 | 8 | 8 | 10 | 7 | 8 | 8 | 8,00 |
| 22 Puedo identificar claramente las opcior | 10 | 9 | 1 | 10 | 10 | 10 | 9 | 9 | 7 | 8 | 7 | 8,18 |
| 23 La cantidad de botones y opciones es | 10 | 10 | 8 | 3 | 10 | 9 | 10 | 10 | 8 | 10 | 10 | 8,91 |
| 24 Puedo identificar cuando paso de un m | 10 | 10 | 1 | 10 | 10 | 10 | 9 | 10 | 8 | 10 | 10 | 8,91 |
| 25 Entiendo lo que debo hacer en cada pa | 10 | 10 | 4 | 8 | 10 | 9 | 8 | 8 | 6 | 8 | 7 | 8,00 |
| 26 Sé cuando una acción es realizada o r | 10 | 10 | 3 | 6 | 8 | 10 | 10 | 10 | 10 | 10 | 9 | 8,73 |
| 27 Es posible jugar sin un asistente | 10 | 10 | 7 | 10 | 10 | 9 | 10 | 10 | 10 | 8 | 9 | 9,36 |

Figura 32: Resultados de evaluación de los 11 usuarios durante el proceso de evaluación de usabilidad y jugabilidad.

9.2.1. Usuario A

| Parámetros de observación | No | +/- | Sí | Comentarios |
|--|----|-----|----|-------------|
| Desliza el dedo para navegar el menú | | | X | |
| Toca dos veces para seleccionar una opción | | | X | |
| Toca una vez para consultar una opción | | | X | |

Figura 33: Observaciones Tarea 1: Reconocimiento de interfaz - Usuario A.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Reconoce el menú tutorial | | | X | |
| Escoge la opción tutorial tocando dos veces | | | X | |
| Escoge el tutorial completo tocando dos veces | | | X | |

Figura 34: Observaciones Tarea 2: Reconocimiento de tutorial - Usuario A.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Reconoce cuadrículas al tocarlas una vez | | | X | |
| Realiza una acción tocando dos veces | | | X | |
| Reconoce el menú oculto tocando con ambos dedos la pantalla | | | X | |
| Selecciona el objetivo en el menú oculto tocando dos veces | | | X | |
| Reconoce la cuadrícula (0,0) | | | X | |
| Realiza una acción tocando la cuadrícula (0,0) | | | X | |
| Reconoce que hay una nave espacial tocándola una vez | | | X | |

Figura 35: Tarea 3: Reconocimiento del plano cartesiano y el objetivo - Usuario A.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Escoge la opción tutorial tocando dos veces | | | X | |
| Escoge el tutorial de Rayo Láser tocando dos veces | | | X | |
| Reconoce que hay un enemigo y lo busca | | | X | |
| Encuentra al enemigo y lo toca dos veces | | X | | |
| Busca la cuadrícula bajo el enemigo para disparar | | | X | |
| Toca dos veces debajo del enemigo para disparar el rayo láser | | | X | |

Figura 36: Tarea 4: Jugar tutorial de Rayo Láser - Usuario A.

9.2.2. Usuario B

| Parámetros de observación | No | +/- | Sí | Comentarios |
|--|----|-----|----|-------------|
| Desliza el dedo para navegar el menú | | | X | |
| Toca dos veces para seleccionar una opción | | | X | |
| Toca una vez para consultar una opción | | X | | |

Figura 37: Observaciones Tarea 1: Reconocimiento de interfaz - Usuario B.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Reconoce el menú tutorial | | | X | |
| Escoge la opción tutorial tocando dos veces | | | X | |
| Escoge el tutorial completo tocando dos veces | | | X | |

Figura 38: Observaciones Tarea 2: Reconocimiento de tutorial - Usuario B.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Reconoce cuadrículas al tocarlas una vez | | | X | |
| Realiza una acción tocando dos veces | | | X | |
| Reconoce el menú oculto tocando con ambos dedos la pantalla | X | | | |
| Selecciona el objetivo en el menú oculto tocando dos veces | X | | | |
| Reconoce la cuadrícula (0,0) | | | X | |
| Realiza una acción tocando la cuadrícula (0,0) | | | X | |
| Reconoce que hay una nave espacial tocándola una vez | | | X | |

Figura 39: Tarea 3: Reconocimiento del plano cartesiano y el objetivo - Usuario B.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Escoge la opción tutorial tocando dos veces | | | X | |
| Escoge el tutorial de Rayo Láser tocando dos veces | | | X | |
| Reconoce que hay un enemigo y lo busca | | | X | |
| Encuentra al enemigo y lo toca dos veces | | X | X | |
| Busca la cuadrícula bajo el enemigo para disparar | | | X | |
| Toca dos veces debajo del enemigo para disparar el rayo láser | | | | |

Figura 40: Tarea 4: Jugar tutorial de Rayo Láser - Usuario B.

9.2.3. Usuario C

| Parámetros de observación | No | +/- | Sí | Comentarios |
|--|----|-----|----|-------------|
| Desliza el dedo para navegar el menú | | | X | |
| Toca dos veces para seleccionar una opción | | | X | |
| Toca una vez para consultar una opción | | | X | |

Figura 41: Observaciones Tarea 1: Reconocimiento de interfaz - Usuario C.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Reconoce el menú tutorial | | | X | |
| Escoge la opción tutorial tocando dos veces | | | X | |
| Escoge el tutorial completo tocando dos veces | | | X | |

Figura 42: Observaciones Tarea 2: Reconocimiento de tutorial - Usuario C.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Reconoce cuadrículas al tocarlas una vez | | | X | |
| Realiza una acción tocando dos veces | | | X | |
| Reconoce el menú oculto tocando con ambos dedos la pantalla | X | | | |
| Selecciona el objetivo en el menú oculto tocando dos veces | X | | | |
| Reconoce la cuadrícula (0,0) | | | X | |
| Realiza una acción tocando la cuadrícula (0,0) | | | X | |
| Reconoce que hay una nave espacial tocándola una vez | | | X | |

Figura 43: Tarea 3: Reconocimiento del plano cartesiano y el objetivo - Usuario C.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Escoge la opción tutorial tocando dos veces | | | X | |
| Escoge el tutorial de Rayo Láser tocando dos veces | | | X | |
| Reconoce que hay un enemigo y lo busca | X | | | |
| Encuentra al enemigo y lo toca dos veces | X | | | |
| Busca la cuadrícula bajo el enemigo para disparar | | | X | |
| Toca dos veces debajo del enemigo para disparar el rayo láser | | | X | |

Figura 44: Tarea 4: Jugar tutorial de Rayo Láser - Usuario C.

9.2.4. Usuario D

| Parámetros de observación | No | +/- | Sí | Comentarios |
|--|----|-----|----|-------------|
| Desliza el dedo para navegar el menú | | | X | |
| Toca dos veces para seleccionar una opción | | | X | |
| Toca una vez para consultar una opción | | | X | |

Figura 45: Observaciones Tarea 1: Reconocimiento de interfaz - Usuario D.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Reconoce el menú tutorial | | | X | |
| Escoge la opción tutorial tocando dos veces | | | X | |
| Escoge el tutorial completo tocando dos veces | | | X | |

Figura 46: Observaciones Tarea 2: Reconocimiento de tutorial - Usuario D.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Reconoce cuadrículas al tocarlas una vez | | | X | |
| Realiza una acción tocando dos veces | | | X | |
| Reconoce el menú oculto tocando con ambos dedos la pantalla | | | X | |
| Selecciona el objetivo en el menú oculto tocando dos veces | | | X | |
| Reconoce la cuadrícula (0,0) | | | X | |
| Realiza una acción tocando la cuadrícula (0,0) | | | X | |
| Reconoce que hay una nave espacial tocándola una vez | | | X | |

Figura 47: Tarea 3: Reconocimiento del plano cartesiano y el objetivo - Usuario D.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Escoge la opción tutorial tocando dos veces | | | X | |
| Escoge el tutorial de Rayo Láser tocando dos veces | | | X | |
| Reconoce que hay un enemigo y lo busca | | | X | |
| Encuentra al enemigo y lo toca dos veces | | | X | |
| Busca la cuadrícula bajo el enemigo para disparar | | | X | |
| Toca dos veces debajo del enemigo para disparar el rayo láser | | | X | |

Figura 48: Tarea 4: Jugar tutorial de Rayo Láser - Usuario D.

9.2.5. Usuario E

| Parámetros de observación | No | +/- | Sí | Comentarios |
|--|----|-----|----|-------------|
| Desliza el dedo para navegar el menú | | | X | |
| Toca dos veces para seleccionar una opción | | | X | |
| Toca una vez para consultar una opción | | | X | |

Figura 49: Observaciones Tarea 1: Reconocimiento de interfaz - Usuario E.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Reconoce el menú tutorial | | | X | |
| Escoge la opción tutorial tocando dos veces | | | X | |
| Escoge el tutorial completo tocando dos veces | | | X | |

Figura 50: Observaciones Tarea 2: Reconocimiento de tutorial - Usuario E.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Reconoce cuadrículas al tocarlas una vez | | | X | |
| Realiza una acción tocando dos veces | | | X | |
| Reconoce el menú oculto tocando con ambos dedos la pantalla | X | | | |
| Selecciona el objetivo en el menú oculto tocando dos veces | X | | | |
| Reconoce la cuadrícula (0,0) | | | X | |
| Realiza una acción tocando la cuadrícula (0,0) | | | X | |
| Reconoce que hay una nave espacial tocándola una vez | | | | |

Figura 51: Tarea 3: Reconocimiento del plano cartesiano y el objetivo - Usuario E.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Escoge la opción tutorial tocando dos veces | | | X | |
| Escoge el tutorial de Rayo Láser tocando dos veces | | | X | |
| Reconoce que hay un enemigo y lo busca | | | X | |
| Encuentra al enemigo y lo toca dos veces | | | X | |
| Busca la cuadrícula bajo el enemigo para disparar | | | X | |
| Toca dos veces debajo del enemigo para disparar el rayo láser | | | X | |

Figura 52: Tarea 4: Jugar tutorial de Rayo Láser - Usuario E.

9.2.6. Usuario F

| Parámetros de observación | No | +/- | Sí | Comentarios |
|--|----|-----|----|-------------|
| Desliza el dedo para navegar el menú | | | X | |
| Toca dos veces para seleccionar una opción | | | X | |
| Toca una vez para consultar una opción | | | X | |

Figura 53: Observaciones Tarea 1: Reconocimiento de interfaz - Usuario F.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Reconoce el menú tutorial | | | X | |
| Escoge la opción tutorial tocando dos veces | | | X | |
| Escoge el tutorial completo tocando dos veces | | | X | |

Figura 54: Observaciones Tarea 2: Reconocimiento de tutorial - Usuario F.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Reconoce cuadrículas al tocarlas una vez | | | X | |
| Realiza una acción tocando dos veces | | | X | |
| Reconoce el menú oculto tocando con ambos dedos la pantalla | | | X | |
| Selecciona el objetivo en el menú oculto tocando dos veces | | | X | |
| Reconoce la cuadrícula (0,0) | | | X | |
| Realiza una acción tocando la cuadrícula (0,0) | | | X | |
| Reconoce que hay una nave espacial tocándola una vez | | | X | |

Figura 55: Tarea 3: Reconocimiento del plano cartesiano y el objetivo - Usuario F.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Escoge la opción tutorial tocando dos veces | | | X | |
| Escoge el tutorial de Rayo Láser tocando dos veces | | | X | |
| Reconoce que hay un enemigo y lo busca | | | X | |
| Encuentra al enemigo y lo toca dos veces | | | X | |
| Busca la cuadrícula bajo el enemigo para disparar | | | X | |
| Toca dos veces debajo del enemigo para disparar el rayo láser | | | X | |

Figura 56: Tarea 4: Jugar tutorial de Rayo Láser - Usuario F.

9.2.7. Usuario G

| Parámetros de observación | No | +/- | Sí | Comentarios |
|--|----|-----|----|-------------|
| Desliza el dedo para navegar el menú | | | X | |
| Toca dos veces para seleccionar una opción | | | X | |
| Toca una vez para consultar una opción | | | X | |

Figura 57: Observaciones Tarea 1: Reconocimiento de interfaz - Usuario G.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Reconoce el menú tutorial | | | X | |
| Escoge la opción tutorial tocando dos veces | | | X | |
| Escoge el tutorial completo tocando dos veces | | | X | |

Figura 58: Observaciones Tarea 2: Reconocimiento de tutorial - Usuario G.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Reconoce cuadrículas al tocarlas una vez | | | X | |
| Realiza una acción tocando dos veces | | | X | |
| Reconoce el menú oculto tocando con ambos dedos la pantalla | | | X | |
| Selecciona el objetivo en el menú oculto tocando dos veces | | | X | |
| Reconoce la cuadrícula (0,0) | | | X | |
| Realiza una acción tocando la cuadrícula (0,0) | | | X | |
| Reconoce que hay una nave espacial tocándola una vez | | | X | |

Figura 59: Tarea 3: Reconocimiento del plano cartesiano y el objetivo - Usuario G.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Escoge la opción tutorial tocando dos veces | | | X | |
| Escoge el tutorial de Rayo Láser tocando dos veces | | | X | |
| Reconoce que hay un enemigo y lo busca | | | X | |
| Encuentra al enemigo y lo toca dos veces | | | X | |
| Busca la cuadrícula bajo el enemigo para disparar | | | X | |
| Toca dos veces debajo del enemigo para disparar el rayo láser | | | X | |

Figura 60: Tarea 4: Jugar tutorial de Rayo Láser - Usuario G.

9.2.8. Usuario H

| Parámetros de observación | No | +/- | Sí | Comentarios |
|--|----|-----|----|-------------|
| Desliza el dedo para navegar el menú | | | X | |
| Toca dos veces para seleccionar una opción | | | X | |
| Toca una vez para consultar una opción | | | X | |

Figura 61: Observaciones Tarea 1: Reconocimiento de interfaz - Usuario H.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Reconoce el menú tutorial | | | X | |
| Escoge la opción tutorial tocando dos veces | | | X | |
| Escoge el tutorial completo tocando dos veces | | | X | |

Figura 62: Observaciones Tarea 2: Reconocimiento de tutorial - Usuario H.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Reconoce cuadrículas al tocarlas una vez | | | X | |
| Realiza una acción tocando dos veces | | | X | |
| Reconoce el menú oculto tocando con ambos dedos la pantalla | X | | | |
| Selecciona el objetivo en el menú oculto tocando dos veces | X | | | |
| Reconoce la cuadrícula (0,0) | | | X | |
| Realiza una acción tocando la cuadrícula (0,0) | | | X | |
| Reconoce que hay una nave espacial tocándola una vez | | | X | |

Figura 63: Tarea 3: Reconocimiento del plano cartesiano y el objetivo - Usuario H.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Escoge la opción tutorial tocando dos veces | | | X | |
| Escoge el tutorial de Rayo Láser tocando dos veces | | | X | |
| Reconoce que hay un enemigo y lo busca | | X | | |
| Encuentra al enemigo y lo toca dos veces | | X | | |
| Busca la cuadrícula bajo el enemigo para disparar | | | X | |
| Toca dos veces debajo del enemigo para disparar el rayo láser | | | X | |

Figura 64: Tarea 4: Jugar tutorial de Rayo Láser - Usuario H.

9.2.9. Usuario I

| Parámetros de observación | No | +/- | Sí | Comentarios |
|--|----|-----|----|-------------|
| Desliza el dedo para navegar el menú | | | X | |
| Toca dos veces para seleccionar una opción | | | X | |
| Toca una vez para consultar una opción | | | X | |

Figura 65: Observaciones Tarea 1: Reconocimiento de interfaz - Usuario I.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Reconoce el menú tutorial | | | X | |
| Escoge la opción tutorial tocando dos veces | | | X | |
| Escoge el tutorial completo tocando dos veces | | | X | |

Figura 66: Observaciones Tarea 2: Reconocimiento de tutorial - Usuario I.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Reconoce cuadrículas al tocarlas una vez | | | X | |
| Realiza una acción tocando dos veces | | | X | |
| Reconoce el menú oculto tocando con ambos dedos la pantalla | | | X | |
| Selecciona el objetivo en el menú oculto tocando dos veces | | X | | |
| Reconoce la cuadrícula (0,0) | | X | | |
| Realiza una acción tocando la cuadrícula (0,0) | | | X | |
| Reconoce que hay una nave espacial tocándola una vez | | | X | |

Figura 67: Tarea 3: Reconocimiento del plano cartesiano y el objetivo - Usuario I.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Escoge la opción tutorial tocando dos veces | | | X | |
| Escoge el tutorial de Rayo Láser tocando dos veces | | | X | |
| Reconoce que hay un enemigo y lo busca | | X | | |
| Encuentra al enemigo y lo toca dos veces | | X | | |
| Busca la cuadrícula bajo el enemigo para disparar | | | X | |
| Toca dos veces debajo del enemigo para disparar el rayo láser | | | X | |

Figura 68: Tarea 4: Jugar tutorial de Rayo Láser - Usuario I.

9.2.10. Usuario J

| Parámetros de observación | No | +/- | Sí | Comentarios |
|--|----|-----|----|-------------|
| Desliza el dedo para navegar el menú | | | X | |
| Toca dos veces para seleccionar una opción | | | X | |
| Toca una vez para consultar una opción | | | X | |

Figura 69: Observaciones Tarea 1: Reconocimiento de interfaz - Usuario J.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Reconoce el menú tutorial | | | X | |
| Escoge la opción tutorial tocando dos veces | | | X | |
| Escoge el tutorial completo tocando dos veces | | | X | |

Figura 70: Observaciones Tarea 2: Reconocimiento de tutorial - Usuario J.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Reconoce cuadrículas al tocarlas una vez | | | X | |
| Realiza una acción tocando dos veces | | | X | |
| Reconoce el menú oculto tocando con ambos dedos la pantalla | X | | | |
| Selecciona el objetivo en el menú oculto tocando dos veces | | | X | |
| Reconoce la cuadrícula (0,0) | | | X | |
| Realiza una acción tocando la cuadrícula (0,0) | | | X | |
| Reconoce que hay una nave espacial tocándola una vez | | | X | |

Figura 71: Tarea 3: Reconocimiento del plano cartesiano y el objetivo - Usuario J.

| Parámetros de observación | No | +/- | Sí | Comentarios |
|---|----|-----|----|-------------|
| Escoge la opción tutorial tocando dos veces | | | X | |
| Escoge el tutorial de Rayo Láser tocando dos veces | | | X | |
| Reconoce que hay un enemigo y lo busca | | | X | |
| Encuentra al enemigo y lo toca dos veces | | | X | |
| Busca la cuadrícula bajo el enemigo para disparar | | | X | |
| Toca dos veces debajo del enemigo para disparar el rayo láser | | | X | |

Figura 72: Tarea 4: Jugar tutorial de Rayo Láser - Usuario J.

9.3. Código clase *Module*

```
using UnityEngine;
using System.Collections;
using System;
using System.Collections.Generic;

[RequireComponent (typeof (AudioSource))]
public class Module : MonoBehaviour, IModuleRegistryProvider
{
    protected const float EVENT_DELAY = 1f;
    private Level level;
    private EntitySpawner spawner;
    protected Board board;

    [Header("Specifics")]
    public float actionTime;
    public int roundsLeft;
    public bool justOneExploration;
    public bool justOneAction;
    public bool reactsAtEnd;

    [Header("Tutorial")]
    public bool tutorial;

    private int totalRounds = 0;
    private int actionLeftTries;
    private float actionLeftTime;
    protected bool isStillExplorable = true;
    private bool isExploring = false;
    private bool countdown = false;
    private bool actionSent = false;
    private bool autoControl = false;
    protected bool lastRoundCompleted = false;
    protected Vector2 autoAction;

    protected AudioSource audioSource;

    [Header("Speechers")]
    public ModuleSpeecher[] moduleSpeechers;
    protected ModuleSpeecher moduleSpeecher;
    private OnEndEventHandler nextAction;
    private OnEndEventHandler savedNextAction;
    private Speech currentSpeech;
    protected Speech toolTipSpeech;

    public delegate void OnEndEventHandler();
    public event OnEndEventHandler GoalEnd;
    public event OnEndEventHandler RoundInitEnd;
```

```

public event OnEndEventHandler ExplorationEnd;
public event OnEndEventHandler ActionPerformed;
public event OnEndEventHandler ActionEnd;
public event OnEndEventHandler ReactionEnd;
public event OnEndEventHandler ModuleEnd;
public event OnEndEventHandler Timeout;

public ModuleRegistryManager registryManager;
[HideInInspector]
public ActionType actionType;
[HideInInspector]
public int actionX;
[HideInInspector]
public int actionY;

public AudioSource AudioSource
{
    get
    {
        return audioSource;
    }
}

public int CurrentRound
{
    get { return totalRounds - roundsLeft; }
}

public virtual bool SuccessfulCompleted() {
    return true;
}

public bool BaseSuccessfulCompleted()
{
    return tutorial || SuccessfulCompleted();
}

public void Awake()
{
    level = Level.instance;
    board = Board.instance;
    spawner = EntitySpawner.instance;
    audioSource = GetComponent();
    InitSpeecher();
    registryManager = new ModuleRegistryManager(this);
    totalRounds = roundsLeft;
}

public void Start() {

```



```

}

public virtual void Run() {
    moduleSpeaker.SetAudioSource(audioSource);
    PlayGoal();
}

protected void InitSpeaker()
{
    moduleSpeaker = tutorial?moduleSpeakers[0]:LoadSpeaker();
    moduleSpeaker.Start();
    toolTipSpeech = moduleSpeaker.GoalSpeech;
}

protected virtual ModuleSpeaker LoadSpeaker()
{
    return moduleSpeakers[0];
}

protected virtual T EasyModuleDifficulty<T>() where T : IModuleDifficulty
{
    return default(T);
}

protected virtual T MediumModuleDifficulty<T>()
{
    return default(T);
}

protected virtual T HardModuleDifficulty<T>()
{
    return default(T);
}

protected virtual Type DifficultyInterface()
{
    return null;
}

protected T LoadDifficulty<T>() where T : IModuleDifficulty
{
    switch (PersistentData.instance.GetDifficulty())
    {
        case Game.Difficulty.EASY:
            moduleSpeaker = moduleSpeakers[0];
            return EasyModuleDifficulty<T>();
        case Game.Difficulty.MEDIUM:
            moduleSpeaker = moduleSpeakers[1];
            return MediumModuleDifficulty<T>();
        case Game.Difficulty.HARD:

```

```

        moduleSpeaker = moduleSpeakers[2];
        return HardModuleDifficulty<T>();
    default:
        moduleSpeaker = moduleSpeakers[0];
        return EasyModuleDifficulty<T>();
    }
}

public virtual void SpawnEntities (){

}

protected virtual void RoundInit(){
    if (moduleSpeaker.RoundSpeech != null &&
        !moduleSpeaker.RoundSpeech.IsEmpty())
        toolTipSpeech = moduleSpeaker.RoundSpeech;
    PlaySpeech(moduleSpeaker.RoundSpeech, OnRoundInitEnd);
}

public void ForceNextState()
{
    if (tutorial && isExploring && !moduleSpeaker.ExplorationSpeech.Ended) {
        return;
    }
    if(level.CoordinateSpeech != null)
        level.CoordinateSpeech.CancelInvoke();
    audioSource.Stop();
    CancelInvoke();
    if (currentSpeech != null) {
        currentSpeech.CancelInvoke();
    }
    RunNextAction();
}

public void ForceNextStateDelayed(float dt)
{
    Invoke("ForceNextState", dt);
}

protected virtual int GetActionTries(){
    return 10;
}

protected virtual bool IsCorrectAction (int i, int j){
    return true;
}

public bool Action (int i, int j){

```

```

    actionX = i;
    actionY = j;
    if ((actionSent && justOneAction) || actionLeftTries == 0 || autoControl) {
        actionType = ActionType.Rejected;
        if(ActionPerformed != null)
            ActionPerformed();
        return false;
    }

    if (!IsCorrectAction (i, j)) {
        if (tutorial)
            return false;
        actionLeftTries--;
        actionType = ActionType.Incorrect;
        if(actionLeftTries == 0) {
            ForceNextStateDelayed(2f);
        }
        if(ActionPerformed != null)
            ActionPerformed();
        return false;
    }
    if(justOneAction)
        countdown = false;

    actionType = ActionType.Correct;
    if(ActionPerformed != null)
        ActionPerformed();
    actionSent = true;
    ProcessAction (i, j);
    return true;
}

/// <summary>
/// Processes the user action.
///
/// This methods must use OnActionEnd() or OnModuleEnd() methods.
/// </summary>
/// <param name="i">The index.</param>
/// <param name="j">J.</param>
protected virtual void ProcessAction(int i, int j){
    OnActionEnd();

}

public virtual void ForceProcessAction()
{
    OnActionEnd();
}

protected virtual void Reaction(){

```

```

        PlaySpeech(moduleSpeecher.ReactionSpeech, OnReactionEnd);
    }

public virtual void ModuleFinish(){
    if (BaseSuccessfulCompleted ()) {
        PlaySpeech(moduleSpeecher.SuccessSpeech, ModuleSuccessfulFinished,
            false);
    } else {
        PlaySpeech(moduleSpeecher.FailureSpeech, ModuleFailed, false);
    }
}

protected virtual void ModuleSuccessfulFinished(){
    PlaySpeech(moduleSpeecher.EndSpeech, OnModuleEnd);
}

protected virtual void ModuleFailed(){
    PlaySpeech(moduleSpeecher.EndSpeech, OnModuleEnd);
}

public void StartReaction(){
    Reaction ();
}

public virtual void PlayGoal(){
    PlaySpeech(moduleSpeecher.GoalSpeech, OnGoalEnd);
}

public void PlayTooltip() {
    if (isExploring && moduleSpeecher.ExplorationSpeech.Ended)
    {
        audioSource.Stop();
        if (tutorial)
        {
            moduleSpeecher.TooltipSpeech = moduleSpeecher.ExplorationSpeech;
            toolTipSpeech = moduleSpeecher.ExplorationSpeech;
        }
        PlaySpeech(toolTipSpeech);
    }
    else if (IsActioning && moduleSpeecher.RoundStartSpeech.Ended) {
        audioSource.Stop();
        if (tutorial)
        {
            moduleSpeecher.TooltipSpeech = moduleSpeecher.RoundStartSpeech;
            toolTipSpeech = moduleSpeecher.RoundStartSpeech;
        }
        PlaySpeech(toolTipSpeech);

        //PlaySpeech(moduleSpeecher.TooltipSpeech);
    }
}

```

```

    }
    else
        audioSource.PlayOneShot(BlindUISound.instance.disabledSoundClip);
}

public void StartRound(){
    roundsLeft--;
    lastRoundCompleted = false;
    actionLeftTries = GetActionTries();
    Spawner.ResetVibrationMode(); ;
    RoundInit ();
}

public virtual void StartExploration(){
    isExploring = true;
    if (!isStillExplorable) {
        OnExplorationEnd();
        return;
    }
    if (justOneExploration)
        isStillExplorable = false;
    PlaySpeech(moduleSpeecher.ExplorationSpeech);
    if(!tutorial)
        InvokeNextAction(OnExplorationEnd, ExplorationDuration());
}

public virtual float ExplorationDuration(){
    return 300;
}

public virtual void Reinforcement() {
    moduleSpeecher.ReinforcementSpeech.Play();
}

public void AutomaticControl() {
    autoControl = true;
    countdown = false;
    float dt = AutomaticAction();
    if(dt >= 0)
        Invoke("StartAutoAction", dt);
}

public void StartAutoAction() {
    autoControl = false;
    Action((int)autoAction.x, (int)autoAction.y);
}

public virtual float AutomaticAction() {
    return 0f;
}

```

```

public void StartTimeoutCountdown(){
    countdown = true;
    actionLeftTime = tutorial?90: actionTime;
}

protected virtual void RemoveModuleWaste(){

}

protected virtual void RemoveRoundWaste(){

}

public void UpdateTimeout(){
    actionLeftTime -= Time.deltaTime;
}

protected void FixedUpdate(){
    if (countdown) {
        UpdateTimeout ();
        if(actionLeftTime <= 0){
            CancelInvoke();
            OnTimeoutEnd();
        }
    }
}

public void MoveSpaceship (Vector2 offset){
    board.MoveEntity (ShipEntity, offset);
    spawner.ResetEntityPosition (ShipEntity);
}

public void MoveSpaceshipToPosition (int i, int j){
    board.MoveEntityToPosition (ShipEntity, i, j);
    spawner.ResetEntityPosition (ShipEntity);
}

protected void MoveAreaEntity(AreaEntity areaEntity, Vector2 offset){
    board.MoveEntity (areaEntity, offset);
    spawner.ResetEntityPosition (areaEntity);
}

protected void MoveAreaEntityToPosition(AreaEntity areaEntity, int i, int j){
    board.MoveEntityToPosition (areaEntity, i, j);
    spawner.ResetEntityPosition (areaEntity);
}

protected void DestroyAreaEntity(AreaEntity areaEntity) {
    board.RemoveAreaEntity(areaEntity);
}

```

```

        areaEntity.DestroyImmediate();
    }

    protected void PlaySpeech(Speech speech)
    {
        if (speech != null)
            speech.Play();
    }

    protected void PlaySpeech(Speech speech, OnEndEventHandler callback)
    {
        if (speech != null)
        {
            currentSpeech = speech;
            nextAction = callback;
            speech.Play(new Speech.OnEndEventHandler(callback));
        }
        else
            callback();
    }

    protected void PlaySpeech(Speech speech, OnEndEventHandler callback, bool
        replace) {
        if(speech != null) {
            currentSpeech = speech;
            nextAction = replace?callback:nextAction;
            speech.Play(new Speech.OnEndEventHandler(callback));
        } else
            callback();
    }

    protected void SaveNextAction() {
        savedNextAction = nextAction;
    }

    protected void RecoverNextAction() {
        nextAction = savedNextAction;
    }

    protected void InvokeNextAction(OnEndEventHandler action, float v)
    {
        nextAction = action;
        Invoke("RunNextAction", v);
    }

    private void RunNextAction()
    {
        if(level.CoordinateSpeech != null)
            level.CoordinateSpeech.CancelInvoke();
        level.AudioSource.Stop();
    }

```

```

        nextAction();
    }

    /* Events */

    protected virtual void OnGoalEnd()
    {
        if (GoalEnd != null)
            GoalEnd();
        if (roundsLeft == 0)
        {
            nextAction = OnModuleEnd;
            ModuleFinish();
        }
        else
        {
            nextAction = OnRoundInitEnd;
            StartRound();
        }
    }

    protected virtual void OnRoundInitEnd(){
        nextAction = OnExplorationEnd;
        this.StartExploration();
        if (RoundInitEnd != null)
            RoundInitEnd();
    }

    protected virtual void OnExplorationEnd()
    {
        isExploring = false;
        actionSent = false;
        nextAction = OnActionEnd;
        PlaySpeech(moduleSpeecher.RoundStartSpeech);
        StartTimeoutCountdown ();
        if(ExplorationEnd != null)
            ExplorationEnd();
    }

    protected virtual void OnActionEnd()
    {
        countdown = false;
        autoControl = false;
        if (ActionEnd != null)
            ActionEnd();
        if (roundsLeft == 0 && !reactsAtEnd)
        {
            OnReactionEnd();
        }
        else

```



```

    {
        nextAction = OnReactionEnd;
        StartReaction();
    }
}

protected virtual void OnReactionEnd()
{
    RemoveRoundWaste ();
    if(ReactionEnd != null)
        ReactionEnd();
    if (roundsLeft == 0)
    {
        nextAction = OnModuleEnd;
        ModuleFinish();
    }
    else
    {
        nextAction = OnRoundInitEnd;
        StartRound();
    }
}

protected virtual void OnModuleEnd()
{
    RemoveModuleWaste ();
    if(ModuleEnd != null)
        ModuleEnd();
}

protected virtual void OnTimeoutEnd()
{
    if (Timeout != null)
        Timeout();
    if (tutorial)
    {
        actionLeftTime = tutorial ? 90 : actionTime;
        return;
    }
    countdown = false;
    autoControl = false;
    if (roundsLeft == 0 && !reactsAtEnd)
    {
        OnReactionEnd();
    }
    else
    {
        PlaySpeech(moduleSpeecher.TimeoutSpeech, StartReaction);
        nextAction = OnReactionEnd;
    }
}

```

```

}

string IModuleRegistryProvider.Name() {
    return name;
}

bool IModuleRegistryProvider.Completed() {
    return SuccessfulCompleted();
}

bool IModuleRegistryProvider.CompletedRound() {
    return lastRoundCompleted;
}

List<AreaEntity> IModuleRegistryProvider.Entities() {
    throw new NotImplementedException();
}

bool IModuleRegistryProvider.Ignore() {
    return false;
}

/*Getters*/

public int RoundsLeft {
    get {
        return roundsLeft;
    }
}

public bool IsStillExplorable {
    get {
        return isStillExplorable;
    }
}

public AreaEntity ShipEntity {
    get {
        return level.SpaceShip;
    }
}

public EntitySpawner Spawner {
    get {
        return spawner;
    }
}

public bool IsExploring {
    get {

```

```
        return isExploring;
    }
}
public bool IsActioning {
    get {
        return countdown;
    }
}

public ModuleSpeecher ModuleSpeecher
{
    get
    {
        return moduleSpeecher;
    }
}

public int ActionLeftTries {
    get {
        return actionLeftTries;
    }
}

public bool AutoControl {
    get {
        return autoControl;
    }
}
}
```
