

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA INFORMÁTICA EN
TECNOLOGÍAS DE LA INFORMACIÓN



Biblioteca

“DESARROLLO DE UNA APLICACIÓN DE
REALIDAD VIRTUAL PARA LA
SIMULACIÓN DE DALTONISMO”

TRABAJO FIN DE GRADO

Diciembre 2022

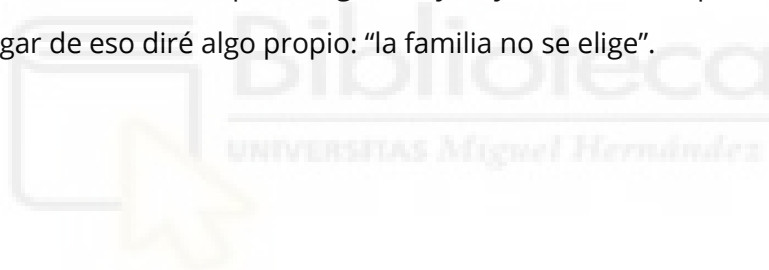
AUTOR: Javier Rodríguez Ruiz
DIRECTOR: Manuel Quesada Martínez

AGRADECIMIENTOS

Agradecer a la Escuela Politécnica Superior de Elche (EPSE) y al Instituto Centro de Investigación Operativa (CIO) de la Universidad Miguel Hernández de Elche por ceder temporalmente el equipamiento necesario para poder desplegar la aplicación de Realidad Virtual aquí desarrollada.

Quiero agradecer a Javier Ángel por fomentar mi curiosidad y mente analítica desde pequeño con esas charlas mientras le ayudaba a recoger el puesto. También quiero agradecer a mis compañeros del Grado Superior sin un orden concreto, Mitogo, Santi, Sergio y Juanico por apoyarme y en especial a Adrián Maciá, sin él nunca habría ido a la Universidad. Claro está también a mis compañeros de la carrera, es especial a Miguel Ángel. Y por último y por ello más importante a Susana, sin ella apoyándome no habría llevado esto a buen puerto.

Estaba buscando citas que me gusten y hay demasiadas que son importante para mí. En lugar de eso diré algo propio: "la familia no se elige".



RESUMEN

Las deficiencias visuales afectan a un gran porcentaje de la población y este problema se ha visto agravado en las últimas décadas. En este proyecto trabajamos sobre el daltonismo, que es una deficiencia visual consistente en una dificultad o impedimento a la hora de distinguir ciertas longitudes de onda o colores concretos debido a un defecto en los conos presentes en la retina.

El objetivo principal de este trabajo es la creación de una Juego Serio de Realidad Virtual (RV) con el que podamos no solo detectar sino también concienciar acerca de los defectos visuales relacionados con el daltonismo y sus distintos tipos. El análisis del estado del arte ha permitido identificar fortalezas y debilidades de aplicaciones relacionadas, que se han utilizado para definir algunos de los requisitos del juego aquí desarrollado. El juego está formado por tres escenarios: (1) el primero permite identificar problemas del jugador por medio del test de Ishihara, (2) el segundo implementa un minijuego que permite al jugador empatizar con la visión de personas daltónicas en un entorno simulado de diferentes tipos de daltonismo, y (3) el tercero ofrece un visor de fotografías 360° en entornos cercanos al usuario ofreciendo información sobre los diferentes tipos de problemas.

El proceso de desarrollo ha requerido un análisis y diseño de software, que se planificado y ejecutado por medio de la metodología ágil GTD. Desde el punto de vista técnico, se ha definido un marco de trabajo basado en Unity y el SDK de Oculus para la implementación de la aplicación de RV. También se han usado otras herramientas de modelado 3D como Blender para editar los modelos 3D generados, algunos de ellos descargados de Sketchfab. Para la simulación de los daltonismos se ha hecho uso de lenguaje C# manipulando las texturas y a través de script que modifica las propiedades de la cámara de Unity. La aplicación de RV ha sido desplegada en dos dispositivos: las Oculus Rift S y las Meta Quest 2. Esto ha permitido realizar una serie de pruebas preliminares con usuarios en las que, a pesar de detectar diferentes aspectos de mejora, los jugadores expresaron su satisfacción con la experiencia de aprendizaje propuesta.

Palabras clave: Realidad Virtual, VR, Oculus, Daltonismo, Protanopia, Deuteranopia, Tritanopia, Acromatopsia, Deficiencia visual, Concienciar.

Índice de Contenido

1	INTRODUCCIÓN	12
1.1	DALTONISMO	13
1.2	LOS VIDEOJUEGOS Y LAS DEFICIENCIAS VISUALES	14
1.2.1	JUEGOS SERIOS Y SU UTILIDAD PARA CONCIENCIAR.....	17
1.2.2	REALIDAD VIRTUAL (RV): DEFINICIÓN Y SU POTENCIAL	19
1.3	OBJETIVOS.....	23
1.3.1	OBJETIVO GENERAL.....	23
1.3.2	OBJETIVOS SECUNDARIOS	23
1.3.3	OBJETIVOS PERSONALES Y FORMATIVOS	24
2	MATERIALES Y MÉTODOS.....	25
2.1	ESTADO DEL ARTE	25
2.1.1	TIPOS DE DALTONISMO Y SOLUCIONES PARA EL USUARIO	25
2.1.2	TEST DE VISIÓN PARA DALTONISMO	27
2.1.2.1	EL TEST DE ISHIHARA	28
2.1.3	ANÁLISIS DE VIDEOJUEGOS PARA LA AGUDEZA VISUAL	30
2.1.4	TIPOS Y CARACTERÍSTICAS DE LOS SISTEMAS DE RV	33
2.1.5	EVOLUCIÓN DE LOS DISPOSITIVOS DE REALIDAD VIRTUAL.....	34
2.1.5.1	OCULUS RIFT S.....	36
2.2	ANÁLISIS DE TECNOLOGÍAS DE DESARROLLO SOFTWARE	37
2.2.1	MOTORES DE VIDEOJUEGOS Y DESARROLLO EN RV	37
2.2.1.1	UNITY vs UNREAL ENGINE.....	37
2.2.1.2	SDKS DE DESARROLLO PARA OCULUS RIFT S.....	40
2.2.2	MODELADO DE OBJETOS 3D.....	41
2.2.3	IMÁGENES Y FOTOGRAFÍAS 360º	42

2.2.4	SIMULAR DALTONISMO: MODIFICAR CÁMARA Y TEXTURAS	44
2.2.5	ENTORNOS DE DESARROLLO	45
2.2.5.1	MICROSOFT VISUAL STUDIO, RIDER Y VSCODE	46
2.2.6	OTRAS HERRAMIENTAS.....	48
2.3	PROPUESTA DE SOLUCIÓN.....	50
3	METODOLOGÍA Y RESULTADOS	51
3.1	METODOLOGÍAS ÁGILES	51
3.1.1	GTD: PRINCIPIOS BÁSICOS	53
3.1.2	GTD: CRONOGRAMA Y LISTA DE TAREAS	55
3.2	ANÁLISIS Y DISEÑO DEL SOFTWARE	57
3.2.1	MAPA DE NAVEGACIÓN Y ESCENARIOS DEL JUEGO	57
3.2.2	CASOS DE USO.....	57
3.2.3	DIAGRAMAS DE FLUJO	59
3.2.4	GESTIÓN Y PERSISTENCIA DE DATOS	60
3.3	ASPECTOS DE IMPLEMENTACION RELAVANTES.....	61
3.3.1	MODELOS 3D E FOTOGRAFÍAS 360º	61
3.3.2	ESCENARIOS QUE COMPONEN EL PROYECTO	63
3.3.3	DINÁMICA DEL MINIJUEGO	65
3.3.4	EDICIÓN DE TEXTURAS Y MANUPULACIÓN DE LA CÁMARA	68
3.3.5	PANELES INTERACTIVOS Y MENÚ SECUNDARIO.....	69
3.3.6	GENERACION DE EJECUTABLE Y DESPLIEGUE.....	70
3.4	PRUEBAS PRELIMINARES CON USUARIOS	70
3.5	CÓDIGO FUENTE	73
3.6	EJEMPLO DE USO DE LA APLICACIÓN	73
3.6.1.1	ESCENARIO 1: LABORATORIO	74
3.6.1.2	ESCENARIO 2: BOSQUE	76

3.6.1.3 FOTOGRAFÍAS 360	79
3.7 VIDEO CON LA DEMO DE LA APLICACIÓN.....	82
4 CONCLUSIONES Y TRABAJO FUTURO	83
4.1 TRABAJO FUTURO	84
5 BIBLIOGRAFÍA	86
ANEXO I: CASOS DE USO.....	99



Índice de Figuras

Figura 1. Listado de las 10 enfermedades de la vista más comunes. a) Daltonismo, b) Miopía, c) Hipermetropía, d) Presbicia, e) Astigmatismo, f) Conjuntivitis, g) Glaucoma, h) Catarata, i) Ojo seco y j) Degeneración macular	12
Figura 2. Bosque con a) visión normal y b) visión tritanope	13
Figura 3. Herencia genética del daltonismo	14
Figura 4. Imagen de alto contraste en The Last Of Us 2.....	15
Figura 5. Araña modificada por el filtro de aracnofobia en el videojuego Grounded.....	16
Figura 6. Opciones para daltónicos en Overwatch.....	16
Figura 7. Ejemplo de uso de filtro de daltonismo en Overwatch.....	17
Figura 8. Capturas de Juegos Serios. a) Go for your life, juego sobre obesidad infantil y b) Foldit, juego para resolver las estructuras de las enzimas	18
Figura 9. Capturas del juego Experience: Colorblindness de iNFINITE Productions. a) Museo con filtro de tritanopia, b) Experiencia interactiva para identificar la lima con un filtro de deuteranopia	19
Figura 10. a) Fotografía de Charles Wheatstone. b) Simulador Link Trainer considerado uno de los primeros productos de RV	20
Figura 11. a) Google Cardboard, unas gafas de RV de cartón. b) Samsung Gear VR, dispositivo de Samsung para VR usando smartphones de su marca compatibles..	20
Figura 12. Imagen donde se muestra el espacio interactivo creado por Melscience para el aprendizaje de física y química	22
Figura 13. Recreación de Notre-Dame por Ubisoft que se puede visitar mediante una visita guiada en RV.....	22
Figura 14. Representación gráfica sobre cómo se visualizan los colores del arcoíris con los distintos tipos de daltonismo	26
Figura 15. Sistema de identificación del color de ColorADD.....	26
Figura 16. a) Anomaloscopio, aparato para medir la sensibilidad a los colores b) Test de Farnsworth, prueba para la detección del daltonismo	28

Figura 17. Láminas de Ishihara donde a) (lámina nº 1) y b) (lámina nº 12) contienen números y c) (lámina nº 19) contiene formas geométricas.....	29
Figura 18. Imagen de la portada de Eye Ok (imagen extraída de (Mora 2022))	30
Figura 19. a) Flash Focus carátula, b) Dos minijuegos.....	31
Figura 20. Imagen del test de Fansworth en el juego serio Experience: Colorblindness	31
Figura 21. Imagen del Youtuber BennyDaBeast jugando al videojuego Beat Saber en un sistema inmersivo de RV en el que se utilizan dos controladores adaptados	34
Figura 22. a) Virtual Boy, la consola fallida de Nintendo con tecnología VR. b) Primer prototipo conocido de las Oculus Rift	35
Figura 23. Fotografía de las Oculus Rift S	36
Figura 24. Interfaz Unity	38
Figura 25. Interfaz de Unreal Engine	38
Figura 26. Submenú de opciones del usuario dentro del menú de opciones de la build. Concretamente configuración del XR Plug-in Management.....	40
Figura 27. Imagen donde se muestra de la interfaz de Blender	41
Figura 28. Resultados de la búsqueda de “fox lowpoly” en Sketchfab.....	42
Figura 29. Resultados de la búsqueda “imágenes 360” en el banco de imágenes Pixexid	43
Figura 30. Ejemplo de uso de Hugin para la realización de una imagen 360º a partir de imágenes propias.....	43
Figura 31. Capturas propias siendo a) captura del escenario real y b) captura del resultado que muestra la cámara tras las modificaciones.....	44
Figura 32. Tabla que muestra cómo se ven representados los distintos colores en función del tipo de visión	45
Figura 33. Interfaz de Visual Studio.....	47
Figura 34. Interfaz de Rider	47
Figura 35. Interfaz de VSCode	48
Figura 36. Interfaz de GitKraken en la que se puede apreciar el control de ramas visual que tiene.....	49
Figura 37. Diagrama con la propuesta de hardware y software de nuestra solución.....	50

Figura 38. Ejemplo de aplicación de GTD mediante un sistema de carpetas físicas conocido como “43 folder system”	53
Figura 39. Cronograma que muestra la distribución de trabajo en el tiempo del proyecto	55
Figura 40. Últimas tareas de la programación del videojuego con Unity	56
Figura 41. Recreación del estado inicial de la tarea “Unity”	56
Figura 42. Mapa de navegación de la aplicación.....	57
Figura 43. Diagrama de casos de uso de la aplicación	58
Figura 44. Diagrama de flujo simplificado de la aplicación	59
Figura 45. Diagrama de flujo detalle del proceso Realizando test.....	59
Figura 46. Diagrama de flujo detalle del proceso Modo Libre	60
Figura 47. Diagrama de flujo detalle del proceso Juego Serio	60
Figura 48. Modelo original de árbol creado en Blender	61
Figura 49. a) Modelo de zorro extraído de Sketchfab. b) Zorro manipulado y en posición base del modelo incluido en nuestro juego.....	62
Figura 50. Imagen 360º de una habitación, utilizada en el proyecto.....	62
Figura 51. Captura del Laboratorio en la que se muestran sus componentes ..	63
Figura 52. Imagen donde se muestran los objetos que componen el escenario de Simulación	64
Figura 53. Escenario Visor 360º donde se muestran sus componentes	64
Figura 54. Capturas comparativas donde se aprecian las diferencias entre el zorro normal y el variocolor siendo a) en un entorno sin defecto y b) en un entorno con protanopia	65
Figura 55. Código de la función encargada de escoger a los zorros	66
Figura 56. Captura de la función Timer, esta se encarga del temporizador del juego	67
Figura 57. Código de la aplicación, concretamente la función MaterialChanger, encargada de cambiar todos los materiales del escenario	67
Figura 58. Código de la función ColorCreator, función necesaria para el correcto funcionamiento de MaterialChanger	68
Figura 59. Cámara con el script de simulación de daltonismo.....	68

Figura 60. Código de la aplicación llamado al presionar los botones en el modo libre del escenario de Simulación.....	69
Figura 61. Captura del editor en el escenario de pruebas en el que se muestra el menú secundario dentro del controlador de la mano izquierda	70
Figura 62. Imagen donde se muestra el repositorio donde está alojado el proyecto en Bitbutcket.....	73
Figura 63. Fotografía del entorno en el que se ha desarrollado parte de la aplicación y la mayoría de las pruebas con usuarios	73
Figura 64. Zorro tutorial en Laboratorio	74
Figura 65. Cubos interactivables del Laboratorio	74
Figura 66. Test Ishihara.....	75
Figura 67. Panel resultados a) no se han detectado problemas, b) se ha detectado anomalía rojo-verde	75
Figura 68. Zorro tutorial en Simulador	76
Figura 69. Menú selección de dificultad.....	76
Figura 70. Capturas del minijuego con visión normal	77
Figura 71. Captura que muestra el indicador del tiempo restante y del defecto actual	77
Figura 72. Capturas de escenario de Simulación dentro del modo libre para simular a) protanopia, b) deuteranomia, c) tritanopia y d) acromatopsia	78
Figura 73. Captura con el panel de resultados del minijuego.....	78
Figura 74. Captura del escenario Visor 360º	79
Figura 75. Capturas del Visor de imágenes con la primera imagen simulando a) protanopia, b) deuteranomia, c) tritanopia y d) acromatopsia	79
Figura 76. Menú secundario en escenario Visor 360º	80
Figura 77. Visor 360º con la segunda imagen simulando protanopia	80
Figura 78. Visor 360º con la segunda imagen simulando deuteranomia	81
Figura 79. Visor 360º con la segunda imagen simulando tritanopia	81
Figura 80. Visor 360º con la segunda imagen simulando acromatopsia	82
Figura 81. Vídeo con la demo del juego serio desarrollado.....	82

Índice de Tablas

Tabla 1. Comparativa entre los juegos serios citados del cumplimiento de las características a destacar	32
Tabla 2. Comparativa de características de una selección de cascos de VR por su popularidad	36
Tabla 3. Comparativa entre Unity y Unreal.....	39



1 INTRODUCCIÓN

Las deficiencias visuales afectan a un gran porcentaje de la población y este problema se ha visto agravado en las últimas décadas. La mayor parte de las enfermedades del ojo, problemas que afectan a la vista, están relacionados con la pérdida parcial o total de visión (Opeluca Team 2022). En la Figura 1 se muestra un resumen visual de las mismas y en este trabajo nos centraremos en el Daltonismo (Figura 1 a)).

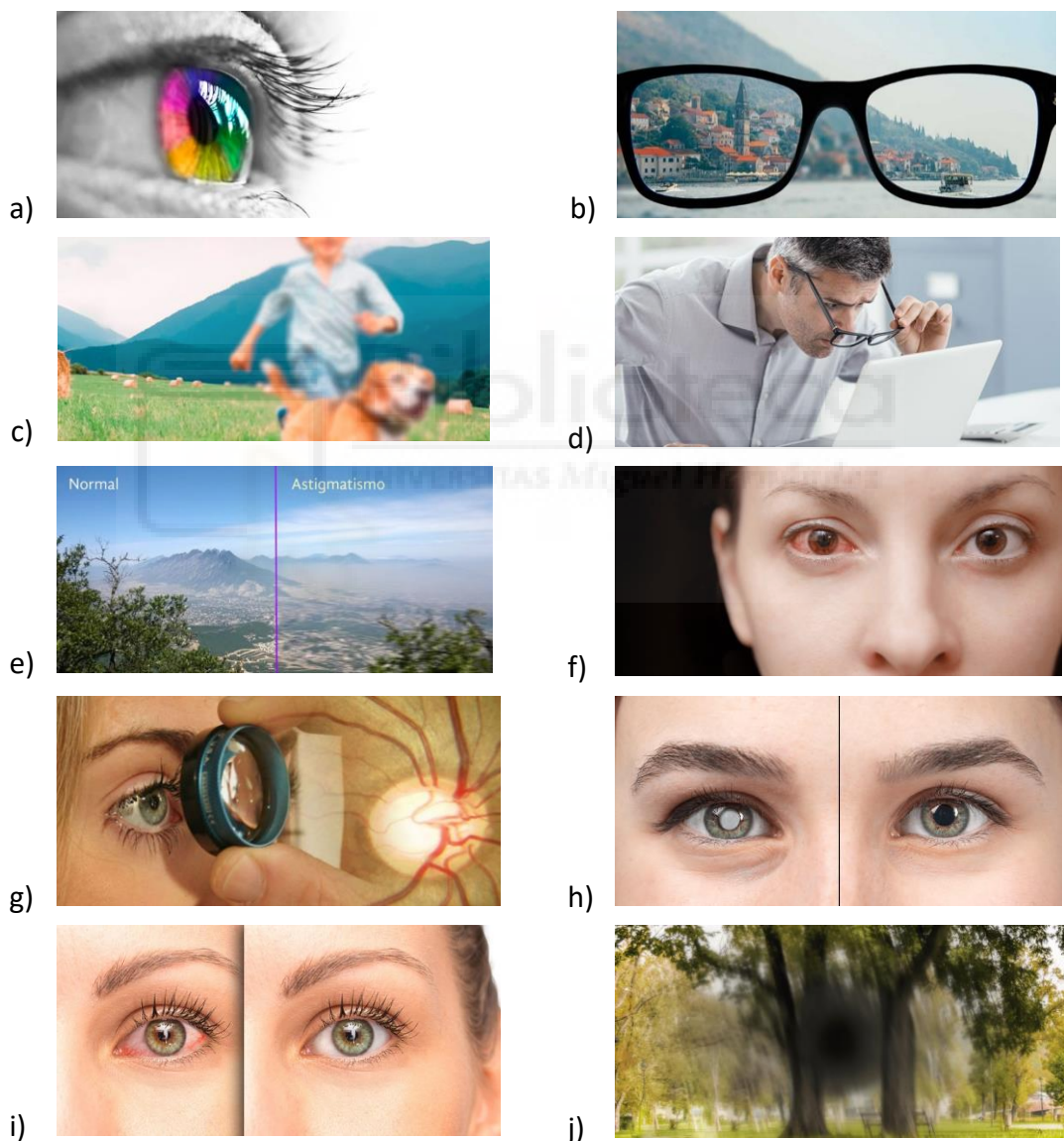


Figura 1. Listado de las 10 enfermedades de la vista más comunes. a) Daltonismo, b) Miopía, c) Hipermetropía, d) Presbicia, e) Astigmatismo, f) Conjuntivitis, g) Glaucoma, h) Catarata, i) Ojo seco y j) Degeneración macular (imágenes extraídas de (Opeluca Team 2022))

1.1 DALTONISMO

El primer trabajo científico que habla de sobre el daltonismo fue escrito por su descubridor, y primer caso registrado, John Dalton (archiconocido químico, físico y matemático británico). Dalton escribió su artículo titulado *“Hechos extraordinarios relativos a la visión de los colores”* en el que postulaba erróneamente que estas deficiencias visuales eran provocadas por unas anomalías en el humor vítreo (líquido gelatinoso y transparente que rellena el espacio comprendido entre la superficie interna de la retina y la cara posterior del cristalino) que cambiaban su color. Pese a que sus teorías fueron descartadas en vida, mandó conservar sus ojos, lo que permitió en 1995 demostrar con el análisis de su ADN que, efectivamente, tenía ceguera al color verde (Órbita Laika 2020; Wikipedia 2011b).

Actualmente hay alrededor de 350 millones de personas con algún tipo de daltonismo en el mundo (López Escorial 2015). El término Daltonismo es el nombre que recibe el conjunto de defectos visuales basados en la incapacidad total o parcial de detectar uno o varios colores. Esta enfermedad visual es debido a que las células encargadas de captar los colores, los conos, tienen un defecto con sus pigmentos y no son capaces de realizar su trabajo correctamente. Estas células se encuentran formando una capa de tejido que es sensible a la luz recubriendo la parte posterior del ojo, la retina (Medicineplus 2021). Como se muestra en la Figura 2, una persona que sufre daltonismo ve la realidad de forma distinta.

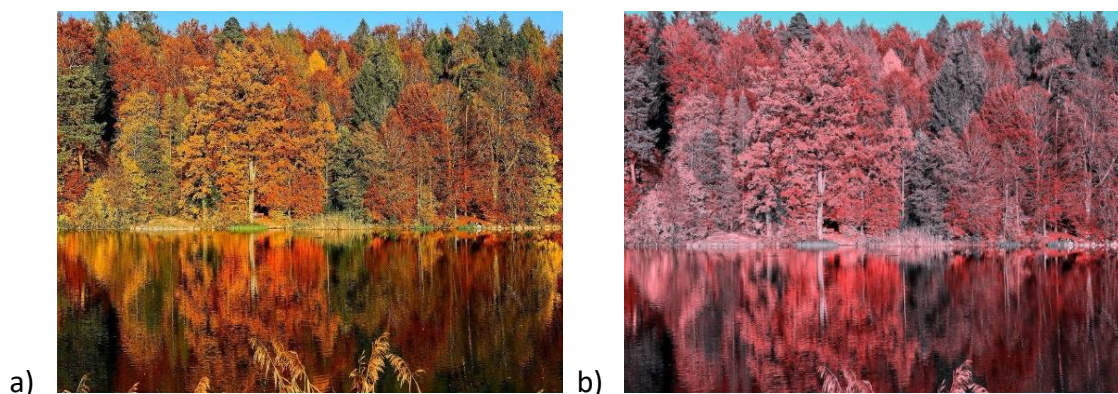


Figura 2. Bosque con a) visión normal y b) visión tritanope (imagen extraída de (Araujo 2021))

Aunque no percibir los mismos colores que el resto del mundo puede ser un problema, en ocasiones su visión del mundo da lugar a imágenes muy bellas. Por ejemplo, en la Figura 2 b) se puede apreciar como las tonalidades típicas del otoño pasan a ser rojos y grises bajo la mirada de una persona que sufre tritanopia (daltonismo que consiste en la carencia de receptores del color azul).

Desde un punto de vista genético y a modo de curiosidad, el daltonismo está relacionado con un defecto genético del cromosoma X. Este hecho provoca que la enfermedad esté más presente en hombres que en mujeres (ver Figura 3). Aproximadamente 1 de cada 12 hombres y 1 de cada 200 mujeres son daltónicos.

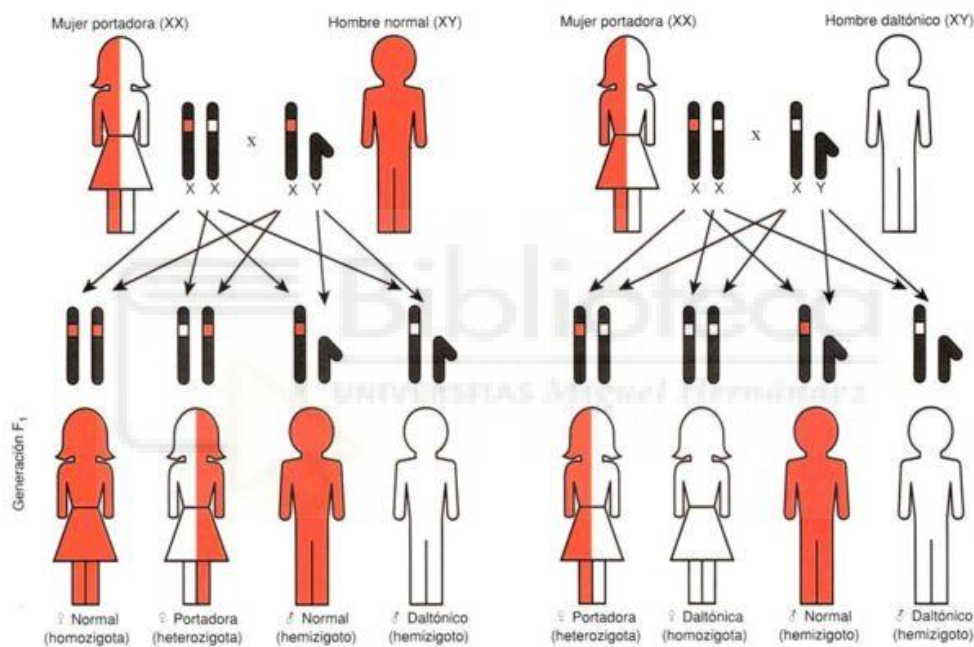


Figura 3. Herencia genética del daltonismo (imagen extraída de (Pau Surribas 2010))

1.2 LOS VIDEOJUEGOS Y LAS DEFICIENCIAS VISUALES

Con este trabajo queremos concienciar acerca de los defectos visuales, y concretamente en aquellos relacionados con el daltonismo. En el ámbito audiovisual, dirigimos nuestro análisis a los videojuegos, ya que despiertan el interés de millones de usuarios en todo el planeta. Estos jugadores tendrán distintos tipos de visión, con o sin problemas cognitivos por lo que resulta interesante analizar como los videojuegos tratan los distintos problemas de este tipo.

Actualmente, los videojuegos abordan muchos de los problemas relacionados con la visión a través de las cada vez más presentes opciones de accesibilidad. Por ejemplo, el modo con narrador para las personas con defectos en la visión del videojuego *The Last Of Us 2* (PlayStation 2020; Wolds 2020) es un referente en cuanto a medidas de accesibilidad. Este videojuego permite configurar opciones como: (1) imágenes de alto contraste, (2) HUD extragrande, (3) apuntado automático, (4) recogida de objetos automática, (5) indicadores de percepción, (6) dirección de procedencia de los subtítulos, (7) mantener el botón para realizar combos cuerpo a cuerpo (8) y controles completamente reconfigurables y un amplio etcétera. En la Figura 4 se muestra un ejemplo del videojuego con el modo de alto contraste activo. Esta solución facilita el uso del juego por parte de personas que tengan problemas identificando elementos en la interfaz o en el propio juego.



Figura 4. Imagen de alto contraste en *The Last Of Us 2* (imagen extraída de (PlayStation 2020))

En este trabajo también estamos interesados por desarrollar una solución que permita a los usuarios vivir una experiencia en la que puedan empatizar con problemas de otros. En este sentido, ponemos como ejemplo el modo aracnofobia de *Grounded* (Dowling 2020) que convierte a los arácnidos en esferas para hacer la vida más fácil a todos aquellos que no soportan las arañas. En la parte a) de la Figura 5 tenemos un ejemplo de la araña sin manipular parte y en b) la araña manipulada.

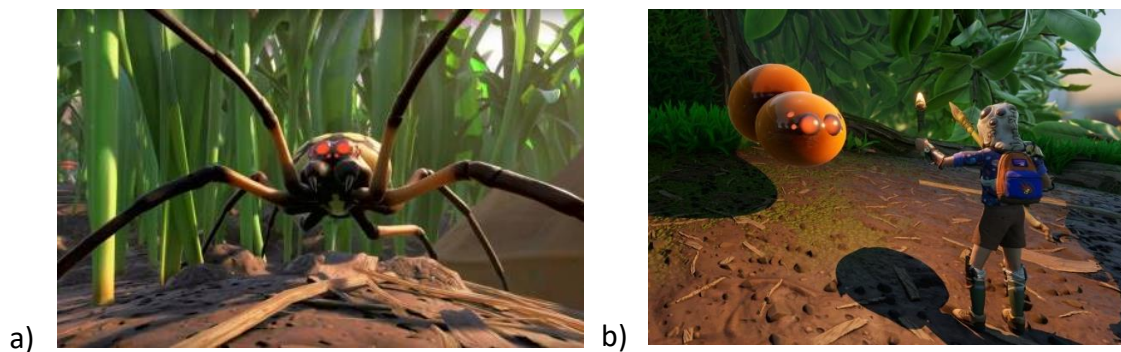


Figura 5. Araña modificada por el filtro de aracnofobia en el videojuego Grounded (imagen extraída de (Dowling 2020))

En relación con el daltonismo, en los videojuegos modernos no es extraño encontrar un modo para daltónicos. Generalmente, la activación de este modo cambia los colores de algunas partes clave de la interfaz del juego para que sean más fáciles de distinguir. Otros juegos van un poco más allá permitiendo escoger el tipo de daltonismo que se sufre y sobre qué se aplican los filtros (interfaz, mundo o ambos).

Por ejemplo, en el juego Overwatch puedes seleccionar la intensidad del daltonismo mediante un *slider* como se muestra en la Figura 6. En la Figura 7 podemos observar el antes (parte izquierda) y el después (parte derecha) de aplicar el filtro.



Figura 6. Opciones para daltónicos en Overwatch (imagen extraída de (Frankie MB 2018))

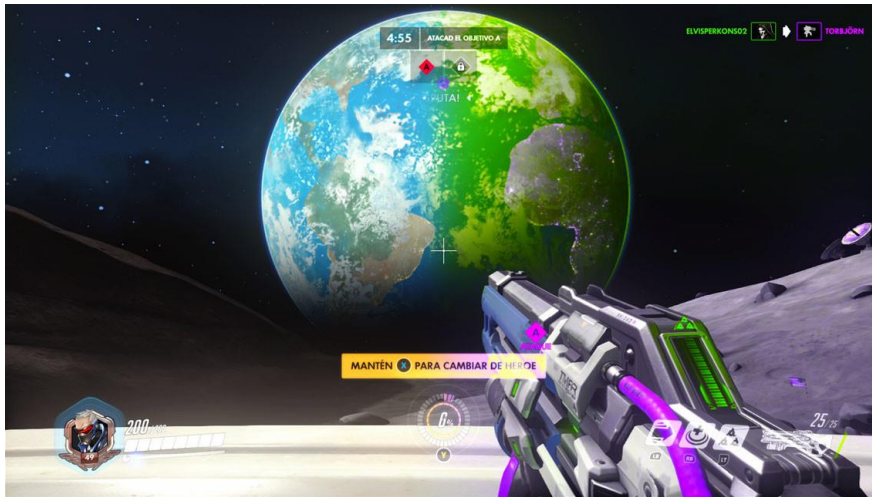


Figura 7. Ejemplo de uso de filtro de daltonismo en Overwatch (imagen extraída de (Frankie MB 2018))

Ejemplos de este tipo de medidas ponen de manifiesto la presencia de soluciones a distintos tipos de problemas cognitivos en un mercado en auge como el de los videojuegos. Creemos que el formato videojuego hace atractivo su uso al todo tipo de públicos; por tanto, queremos contribuir a la concienciación sobre daltonismo con una solución en esta línea. Es decir, que aborde el problema desde una perspectiva gamificada e interactiva haciendo uso de un Juegos Serio.

1.2.1 JUEGOS SERIOS Y SU UTILIDAD PARA CONCIENCIAR

El término “Juego Serio” es utilizado para referirse a juegos que no tienen la diversión como objetivo principal. Normalmente su objetivo es potenciar la formación, desarrollo de habilidades específicas, comprensión de procesos complejos (Díaz 2015; Díaz, Queiruga, y Fava 2015). Desde el punto de su evolución, Clark Abt acuñó este término en 1970 en su libro *“Serious Games”* donde hablaba principalmente de juegos de mesa y de cartas. En 2005, Mike Zyda actualizó el termino en la revista *“Computer”* de la IEEE Computer Society definiéndolo como *“una prueba mental, de acuerdo con unas reglas específicas, que usa la diversión como un modo de formación gubernamental o corporativo, con objetivos en el ámbito de la educación, sanidad, política pública y comunicación estratégica”* (Marcano 2008).

Las conclusiones de (Calleros et al. 2019; González Calleros, Guerrero García, y Navarro Rangel 2019) sobre el uso de Juegos Serios en procesos de aprendizaje dejan patente que proponen una forma de adquirir conocimientos mucho más interactiva y amena que las metodologías tradicionales. Los Juegos Serios presentan diversas ventajas frente aprender de forma tradicional como: (1) abordan los problemas de forma lúdica, (2) brindan oportunidades de aprender sin riesgo a ser juzgados por los fallos, (3) permiten cambiar los roles, (4) promueven la experimentación y (6) propician una mejora en las habilidades cognitivas, así como en la creación de estrategias de resolución de problemas.

Algunos ejemplos representativos de juegos serios en distintos ámbitos son: (1) Go for your life (Guitarte 2011b) que propone una experiencia para móviles sobre la obesidad infantil con el objetivo de prevenir sobre los problemas que pueden derivar en obesidad e inculcar una dieta sana y ejercicio físico en el usuario a modo de prevención (ver Figura 8 a)), (2) Foldit (FoldIt 2022; Guitarte 2011a) usado para resolver la estructura de una encima presente en el VIH en tres semanas, alcanzando un logro que la ciencia no había conseguido en más de una década (ver Figura 8 b)) , o (3) Un viaje a través de las Matemáticas (Calleros et al. 2019) enfocado en la enseñanza de las matemáticas a niños con TDAH.

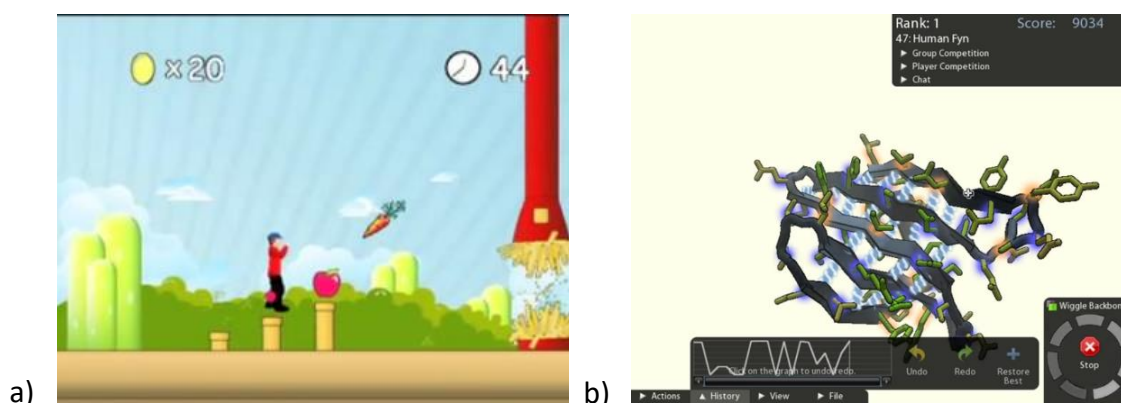


Figura 8. Capturas de Juegos Serios. a) Go for your life, juego sobre obesidad infantil (imagen extraída de (Guitarte 2011b)) y b) Foldit, juego para resolver las estructuras de las enzimas (imagen extraída de (Guitarte 2011a))

Al dirigir nuestra búsqueda a juegos serios relacionados con daltonismo, el número de resultados se vio reducido considerablemente. Sin embargo, hemos encontrado el simulador Experience: Colorblindness (iNFINITE Production 2022) que

destaca por su similitud con el producto que aquí queremos desarrollar. Este simulador permite al usuario ponerse en la piel de un daltónico en situaciones cotidianas como: descansar en el jardín, hacer la compra, visitar un museo, etc. Además, destacamos que este simulador debe ser usado en un entorno de Realidad Virtual donde la experiencia del usuario en un entorno inmersivo permite captar mejor su atención y lograr una mayor concienciación. Por ejemplo, en la Figura 9 se pueden apreciar dos de estas situaciones. El problema de este juego es la falta de una parte jugable que sea realmente entretenida, ya que su enfoque es principalmente de simulador. En nuestra propuesta perseguiremos desarrollar una experiencia gamificada que sirva al usuario no solo como concienciación a través de simulaciones sino también de entretenimiento.

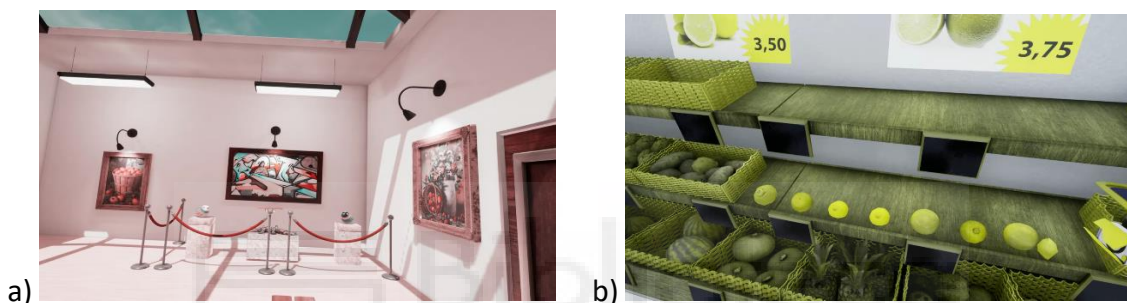


Figura 9. Capturas del juego Experience: Colorblindness de iNFINITE Productions. a) Museo con filtro de tritanopia, b) Experiencia interactiva para identificar la lima con un filtro de deuteranopia (imágenes extraídas de (AltVR 2020))

1.2.2 REALIDAD VIRTUAL (RV): DEFINICIÓN Y SU POTENCIAL

El término “Realidad Virtual” (RV) ha tenido diversas definiciones a lo largo de los años, fruto de los grandes cambios que ha sufrido esta tecnología desde su origen. Hoy en día, la RAE define la RV como *“representación de escenas o imágenes de objetos producidas por un sistema informático, que da la sensación de su existencia real”*.

Pese a ser un concepto novedoso, se encuentran ejemplos de entornos de RV creados por el ser humano desde hace algunos siglos (Xperimenta Cultura 2016). Por ejemplo, en la China de finales del siglo X el artista Zhang Zeduan creó un artefacto de RV a través de una imagen envolvente con sus pinturas panorámicas. Posteriormente, en 1838, el científico e inventor británico Charles Wheatstone (ver Figura 10 a)), logró desarrollar el primer estereoscopio con el que simular una imagen tridimensional. Más tarde, en 1929, el inventor y pionero de la aviación Edwin Albert Link comercializó el

sistema *Link Trainer* (ver Figura 10 b)) con la capacidad de recrear de forma inmersiva un simulador de avión para el entrenamiento de soldados de la marina estadounidense (van Hoek y Link 2003; National Academy of Sciences (U.S.) 1993).

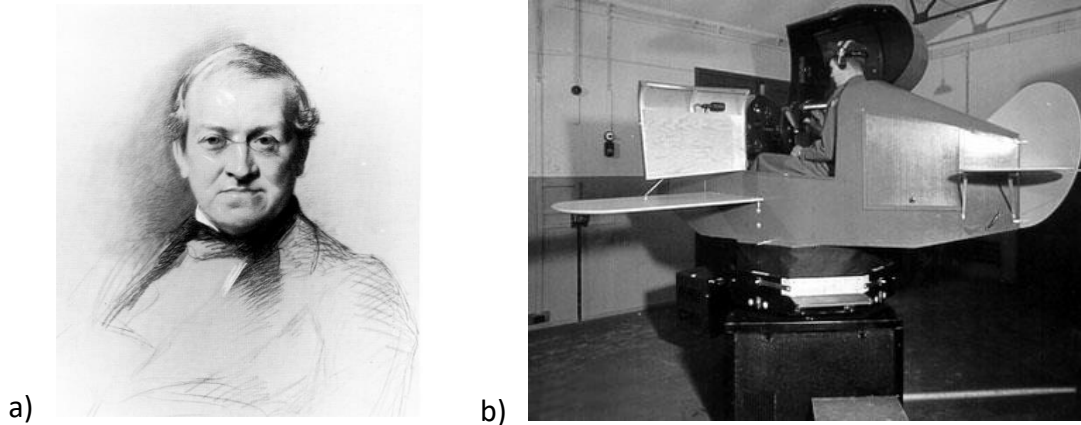


Figura 10. a) Fotografía de Charles Wheatstone (imagen extraída de (Gifreu 2014)). b) Simulador Link Trainer considerado uno de los primeros productos de RV (imagen extraída de (Flightdemy 2022))

En la última década empresas como Google o Samsung han liderado algunas iniciativas para democratizar el uso de esta tecnología con proyectos como Google Cardboard (Google 2022) o Samsung Gear VR ((Samsung 2021)). En estos proyectos se propuso el uso de un dispositivo móvil como pantalla de RV, teniendo el usuario que adquirir únicamente un accesorio donde colocarlo. Por ejemplo, Google comercializó visores de cartón que denominó Cardboard (ver Figura 11 a)) y Samsung optó por un visor más complejo (ver Figura 11 b)) compatible con un conjunto de sus dispositivos.

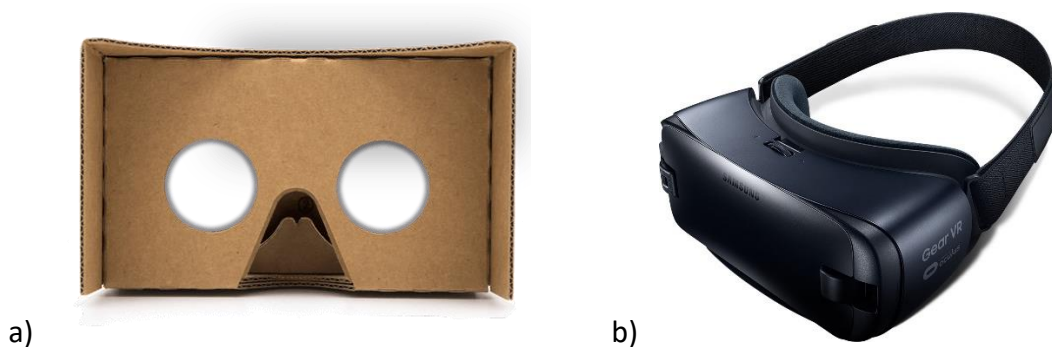


Figura 11. a) Google Cardboard, unas gafas de RV de cartón (imagen extraída de (Google 2022)). b) Samsung Gear VR, dispositivo de Samsung para VR usando smartphones de su marca compatibles (imagen extraída de (Samsung 2021))

Pese a que estas propuestas no tuvieron el éxito esperado, en los últimos años la popularidad de la RV ha crecido significativamente, motivada por su incursión en el mundo de los videojuegos. La llegada en 2012 de las Oculus Rift supuso un punto de inflexión que consolidó el uso de esta tecnología en el ámbito del *gaming*. Oculus Rift es el primer producto de la empresa estadounidense Oculus, especialista tanto de productos software como hardware relacionados con la VR. Oculus Rift se inició como una de las campañas Kickstarter más fructíferas, llegando a recaudar casi 2 millones y medio de dólares (dacangi 2013). Dos años después de sacar las Rift al mercado, Oculus fue adquirida por su actual propietario Mark Zuckerberg (Pozzi 2014), lo que demuestra que Facebook vio el gran futuro de esta tecnología, lo que le llevó a comprar la empresa. La primera remesa de dispositivos fue lo suficientemente potente como para que otras grandes empresas empezaran a ver el potencial que ya había visto Facebook. Esto provocó que años más tarde muchas compañías como PlayStation tuvieran sus propios cascos de VR como PlayStation VR. Pese a su popularidad, actualmente la RV queda reservada a un espectro del público que debe hacer un esfuerzo económico medio-alto para poder adquirir alguno de estos dispositivos.

En 2021 se produce un nuevo punto de inflexión alrededor de la RV. En este año, Facebook decide reorientar su estrategia enfocando su desarrollo a un mundo de RV (Meta 2021), lo que la lleva incluso a cambiar su nombre a Meta. Aparece con fuerza el denominado Metaverso, una realidad digital a la que nos conectaremos usando dispositivos de Realidad Virtual o Realidad Aumentada. Según Meta, esto nos permitirá llevar una vida paralela casi por completo, reuniones de trabajo, tiendas virtuales y estando incluso todo integrado en su propia economía (Fernández 2022; Meta 2022^a).

Desde el punto de vista de la experiencia de usuario, la RV ha demostrado ser una herramienta muy útil para la enseñanza ofreciendo un aprendizaje basado en la experiencia y el entendimiento mejorando significativamente la capacidad de retención (Meneses 2021). Por ejemplo, en el campo de la Medicina o la Ingeniería, es interesante disponer de modelos virtuales que manipular para así poder comprender en mayor medida el funcionamiento de las estructuras (Medellín et al. 2014). Como podemos apreciar en la **Figura 12**, la empresa Melscience ha desarrollado una aplicación de RV para el aprendizaje de física y química de forma interactiva y sin riesgo (Melscience

2022). Otro ejemplo de aprendizaje mediante el uso de VR es la visita guiada que hizo Ubisoft a Notre-Dame basada en sus estudios para la localización de uno de sus juegos (de Rochefort 2021) (ver **Figura 13**).

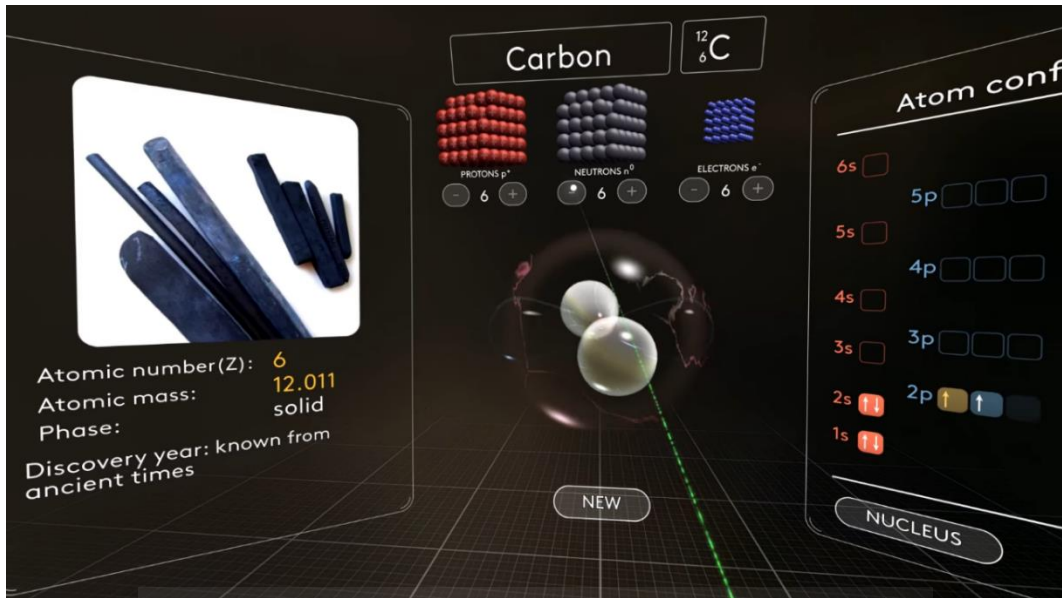


Figura 12. Imagen donde se muestra el espacio interactivo creado por Melscience para el aprendizaje de física y química (imagen extraída de (Melscience 2022))



Figura 13. Recreación de Notre-Dame por Ubisoft que se puede visitar mediante una visita guiada en RV (imagen extraída de (Ubisoft 2020))

Por todos estos motivos, creemos que desarrollar un entorno de simulación de daltonismo usando un sistema inmersivo de RV complementará el trabajo ya existente ofreciendo una experiencia enriquecedora para el usuario.

1.3 OBJETIVOS

En este proyecto se creará un Juego Serio para concienciar sobre daltonismo. Se hará uso de la Realidad Virtual para crear una mayor inmersión en el entorno simulado. Además, disponemos de unas Oculus Rift S, prestadas por la Escuela Politécnica Superior de Elche en las que deberá quedar desplegado el videojuego para poder probar su funcionamiento.

1.3.1 OBJETIVO GENERAL

La creación de un Juego Serio de Realidad Virtual haciendo uso de un motor de desarrollo de videojuegos, que sea capaz de detectar y simular distintos tipos de daltonismo.

1.3.2 OBJETIVOS SECUNDARIOS

Para conseguir el objetivo anterior debemos alcanzar los siguientes objetivos secundarios:

- **Obj 1.** Analizar los fundamentos físicos y multimedia que hay detrás de los daltonismos para poder simularlos. Estudio de pruebas para su detección.
- **Obj 2.** Analizar los trabajos y herramientas relacionadas que nos permitan identificar funcionalidades interesantes de incluir en nuestra propuesta.
- **Obj 3.** Definir un marco de trabajo apropiado para desarrollar un Juego Serio de RV con entornos: (1) de diagnóstico donde realizar la prueba de daltonismo y obtener sus resultados, y (2) de simulación mediante un minijuego interactivo en un mundo virtual.
- **Obj 4.** Diseño e implementación de un prototipo funcional de aplicación de RV que pueda ser desplegado en un dispositivo de RV.

1.3.3 OBJETIVOS PERSONALES Y FORMATIVOS

La principal motivación para escoger este Trabajo Fin de Grado (TFG) fue una inquietud que he tenido desde pequeño: siempre he querido que la gente que no necesita gafas sepa como veo sin ellas. Mi primer intento lo realicé cuando era pequeño y prestaba mis gafas a los compañeros de clase quienes se mareaban al instante. Sin embargo, con ese primer experimento sabía que no estaba haciéndoles sentir lo yo sentía sin ellas. Aunque en mi caso el problema es la miopía, esta enfermedad ya se había trabajado previamente en (Martínez Botella 2022) por lo que descarté su uso. En su lugar, apareció la posibilidad de dedicar este TFG a una deficiencia visual completamente distinta, el daltonismo, y lo escogí sin dudar.

Desde un punto de vista académico, este TFG ha sido un reto ya que durante la carrera no se estudian este tipo de tecnologías. En mi caso, a pesar de tener algo de experiencia en el campo del desarrollo de videojuegos, nunca había desarrollado un proyecto funcional desde cero. A esto hay que añadir que no había hecho uso de unas gafas de RV por lo que desconocía las peculiaridades del desarrollo de soluciones que este tipo de dispositivos. Programar un Juego Serio, integrarlo y desplegarlo en un sistema de RV también me llamó la atención y me animó a escoger este TFG.

2 MATERIALES Y MÉTODOS

En este capítulo continuaremos el análisis iniciado en la introducción dirigiendo nuestro enfoque a aspectos relacionados con el daltonismo y las tecnologías necesarias para desarrollar nuestro Juego Serio de RV.

2.1 ESTADO DEL ARTE

En este apartado analizaremos el estado del arte desde el estudio de fundamentos físicos que hay detrás del daltonismo o sus tipos, hasta el análisis de aplicaciones relacionadas y tecnologías actuales que utilizar para desarrollar el juego.

2.1.1 TIPOS DE DALTONISMO Y SOLUCIONES PARA EL USUARIO

Como comentamos en la introducción el daltonismo es la incapacidad total o parcial de detectar uno o varios colores. Dependiendo del grado y del tipo de color no percibido, en (Méndez 2016) se clasifican los daltonismos en tres grandes familias: (1) Tricromía anómala, (2) Dicromacia y (3) Monocromía. A su vez, en cada familia se identifican diferentes tipos de problemas relacionados con la dificultad de percibir según qué colores básicos.

- **Tricromía anómala:** dificultad para percibir ciertos colores.
 - Protanomalía: dificultad para percibir el color rojo.
 - Deuteranomalía: dificultad para percibir el color verde.
 - Tritanomalía: dificultad para percibir el color azul.
- **Dicromacia:** ausencia para percibir ciertos colores.
 - Protanopia: no se percibe el color rojo.
 - Deuteranopia: no se percibe el color verde
 - Tritanopia: no se percibe el color azul.
- **Monocromía:** no permite la visión de ningún color, se denomina Acromatopsia.

En la Figura 14 se muestran visualmente las diferencias entre la visión normal y los diferentes tipos de dicromacias.



Figura 14. Representación gráfica sobre cómo se visualizan los colores del arcoíris con los distintos tipos de daltonismo (imagen extraída de (Alomar 2017))

Para facilitar la vida a las personas que sufren este tipo de problemas de visión, existen soluciones como la del diseñador gráfico portugués Miguel Neiva, quien propone el sistema ColorADD (López Escorial 2015; Nieva 2010). En este sistema se añade a cada color básico un símbolo representativo del mismo como se muestra en la Figura 15.

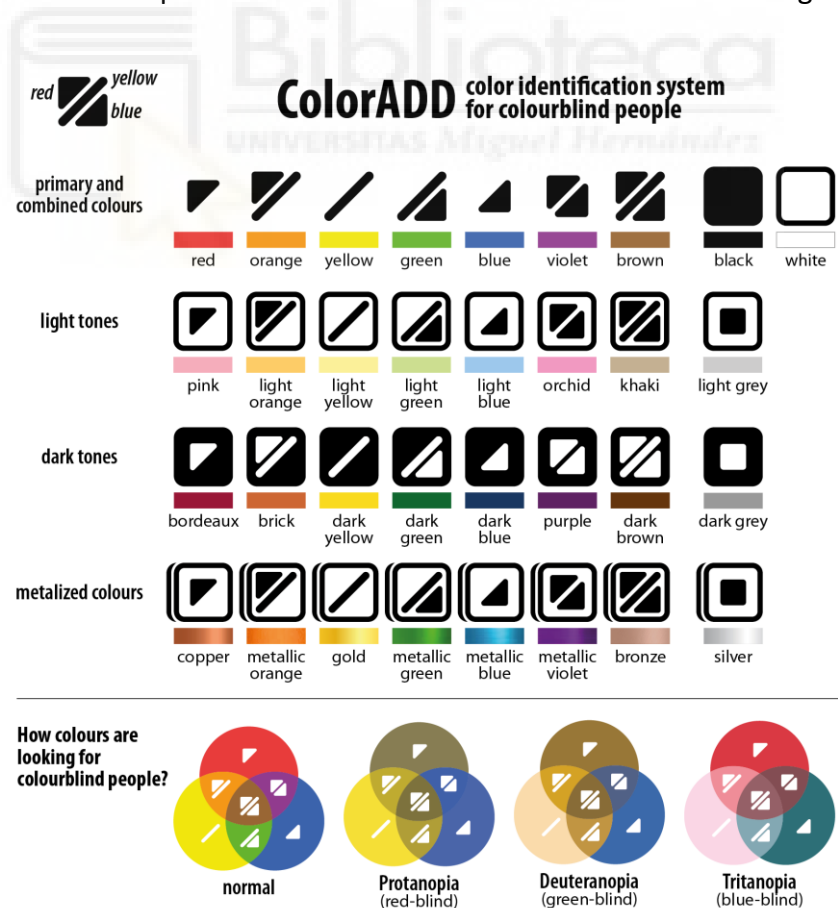


Figura 15. Sistema de identificación del color de ColorADD (imagen extraída de (Wikipedia 2011a))

Merece la pena destacar que los símbolos propuestos por ColorADD están lejos de ser elegidos de forma azarosa ya que se basan en el sistema sustractivo del color del mundo real. En el sistema sustractivo tradicional (RYB) es el que encontramos en los pigmentos, en este el naranja es resultado de la mezcla de rojo y amarillo, así como el verde sería amarillo y azul (Quirell 2017). Por tanto, como se puede ver en la parte inferior izquierda de la Figura 15, el símbolo del naranja, que es la suma de los colores primeros rojo y el amarillo se obtiene uniendo los símbolos de estos dos colores.

Por un lado, soluciones como ColorADD son útiles para daltónicos porque, se ha de tener en cuenta que, la persona daltónica lo es para todo: desde ir a comprar ropa hasta coger un tren, pasando por tomarse una pastilla. Estas acciones cotidianas tienen una alta carga visual ya que, sin darnos cuenta, dotamos de carga semántica a elementos según sus colores. Por otro lado, el problema de este tipo de soluciones es que se necesitan incluir los símbolos y asociarlos con todos los colores que percibe una persona con daltonismo. Esto dificulta su uso como ayuda para mejorar procesos cotidianos que no han sido integrados con soluciones de este tipo.

Como alternativa a soluciones como la anterior, últimamente se han popularizado las llamadas gafas para daltónicos. Lejos de dotar a la gente con una visión cromática perfecta, estas gafas filtran ciertas longitudes de onda para que el cerebro pueda diferenciar mejor los colores que de otro modo le parecen iguales. Pese a no ser una solución perfecta, es una mejora que puede ayudarles en su día a día (Grefer 2022).

2.1.2 TEST DE VISIÓN PARA DALTONISMO

Hoy en día existen diversas pruebas para detectar las deficiencias en la percepción del color. Existen soluciones de diferentes niveles de complejidad y requerimientos técnicos. Por ejemplo, en la **Figura 16 a)** se muestra un anomaloscopio que es un sistema muy preciso, pero que requiere de instrumental especializado. Este aparato pone al paciente frente al reto de igualar dos colores, a y b. El usuario hace esto moviendo una barra cuya posición indica la proporción de dos de los colores que componen b (de Fez Saiz et al. 2022). Por otro lado, en la **Figura 16 b)** se muestra la prueba de Farnsworth (J. Luque, D. de Fez, y A. Díez 2001) en la que el paciente debe ordenar unos discos de colores de forma gradual. Mientras que la prueba de Farnsworth

se puede identificar el tipo de grado de daltonismo del paciente sin la necesidad de aparatos complejos, el uso del anomaloscopio va más allá y es el único método a día de hoy, capaz de distinguir entre un dicrómata (persona que solo tiene receptores de dos colores) y un tricrómata anómalo (persona que pese a tener receptores de tres colores algunos no funcionan correctamente).

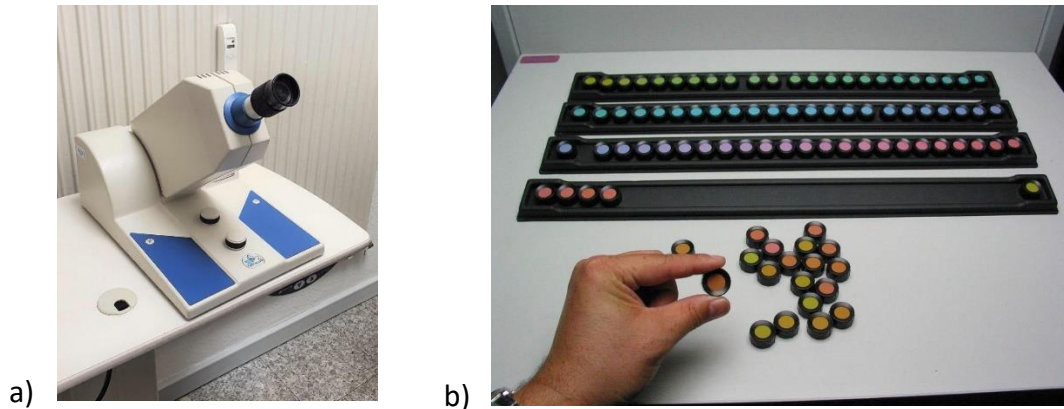


Figura 16. a) Anomaloscopio, aparato para medir la sensibilidad a los colores (imagen extraída de (Sky Optic 2022)) b) Test de Farnsworth, prueba para la detección del daltonismo (imagen extraída de (Área Oftalmológica Avanzada 2020))

2.1.2.1 EL TEST DE ISHIHARA

En este trabajo hemos elegido el Test de Ishihara. Pese a tener esta prueba más de 100 años, sigue siendo a día de hoy uno de los métodos más utilizados para la detección de distintos tipos de daltonismo pese a sus limitaciones para detectar con precisión algunos tipos como la tritanopia.

El japonés Ishihara Shinobu se graduó en 1905 con una beca militar en Medicina. Tras estar unos años en el ejército como cirujano volvió a la Universidad e ingresó en un curso de postgrado en oftalmología. A su regreso a Japón, tras unos años de estudio en Alemania, fue nombrado profesor y presidente, sustituyendo a quien fue su mentor Komoto Jujiro al cargo del Departamento Oftalmológico de la Universidad Imperial de Tokio (hmgong 2022). Durante su periodo en la Escuela de Medicina Militar se le pidió que encontrara una manera de detectar anomalías en la visión de los colores en los soldados. Para esto desarrollo unas primeras placas con puntos de colores en las cuales se podía distinguir un carácter hiragana (método de escritura asiático) representado con unos colores distintos al resto. Así, una persona sin deficiencias de este tipo podría leer

el hiragana sin problema, mientras que alguien con daltonismo, como su asistente, no sería capaz de distinguir el carácter.

La prueba de Ishihara consiste en mostrar al paciente una serie de imágenes compuestas por puntos en las cuales por diferencia de colores se pueden apreciar distintas formas o figuras las cuales no pueden ser vistas por ojos con determinados problemas a la hora de captar el color. Esta prueba originalmente constaba de 38 placas las cuales en grupos de 4 o 5 detectaban el mismo tipo de anomalía de la misma forma, puedes encontrar algunos ejemplos en la Figura 17. Por ejemplo, hay 4 placas con números que una persona protánope no puede ver.

Podemos consultar la prueba completa y los tipos de daltonismo que detecta cada placa en (Colblindor 2021). Además, merece la pena destacar que encontramos páginas web en las que se implementa esta prueba, pero no de forma gamificada ni inmersiva con RV. Algunas de estas webs serían: (1) Colorite, en esta puedes realizar una gran variedad de pruebas y puedes incluso realizar un juego de conciliación de colores básicos (Colorite 2022), (2) Clínica Villoria, permite la realización del test de Ishihara (Villoria 2022) y, (3) en la web de EnChroma también disponemos de un test propio basado en la premisa del test de Ishihara (EnChroma 2022).

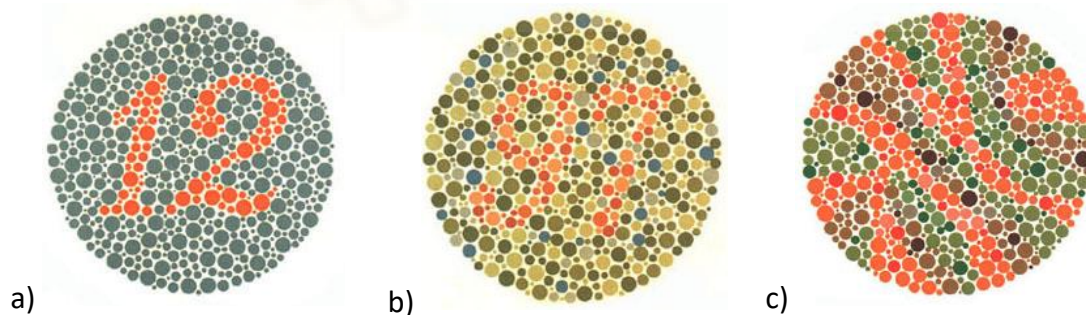


Figura 17. Láminas de Ishihara donde a) (lámina nº 1) y b) (lámina nº 12) contienen números y c) (lámina nº 19) contiene formas geométricas (imágenes extraídas de (Colblindor 2021))

Desde el punto de vista de las condiciones bajo las que se debe realizar el test hay que tener en cuenta lo siguiente. Las placas que componen el test de Ishihara fueron diseñadas para ser mostradas bajo unas situaciones específicas de luz y temperatura para ser 100% efectivas. En (Cwierz et al. 2021) se presenta una discusión sobre la validez de las aplicaciones de RV para el diagnóstico de la visión en color. En el trabajo

de Cwierz se concluye que combinar RV con este tipo de test basados en color no supone un problema, por lo que usarlo como parte de nuestro juego sería acertado.

2.1.3 ANÁLISIS DE VIDEOJUEGOS PARA LA AGUDEZA VISUAL

La concienciación sobre deficiencias visuales ha permeado en nuestra sociedad hasta el punto en que hay videojuegos para prevenir o detectar algunas de ellas. A continuación, mostramos algunos ejemplos interesantes por estar orientados a niños directamente u ofrecen un sistema de tutoría para padres y profesores:

- **Eye Ok (ver Figura 18):** desarrollado por el Instituto Tecnológico de Óptica, Color e Imagen ofrece una herramienta para padres y profesores para identificar miopía, daltonismo, hipermetropía en niños. Enseña buenas prácticas para evitar la fatiga visual (Elena 2015; Longitud 2013; Mora 2022).



Figura 18. Imagen de la portada de Eye Ok (imagen extraída de (Mora 2022))

- **Flash Focus, Vision Training in Minutes a Day (ver Figura 19):** siguiendo la estela de la saga Brain Training, este juego del 2007 consta de ejercicios de agudeza visual y relajación ocular para toda la familia. Su jugabilidad se basa en la resolución de puzles visuales y minijuegos acompañados de ejercicios para realizar con los ojos (Harris 2007; Riera Muñoz 2007; Wikipedia 2019).

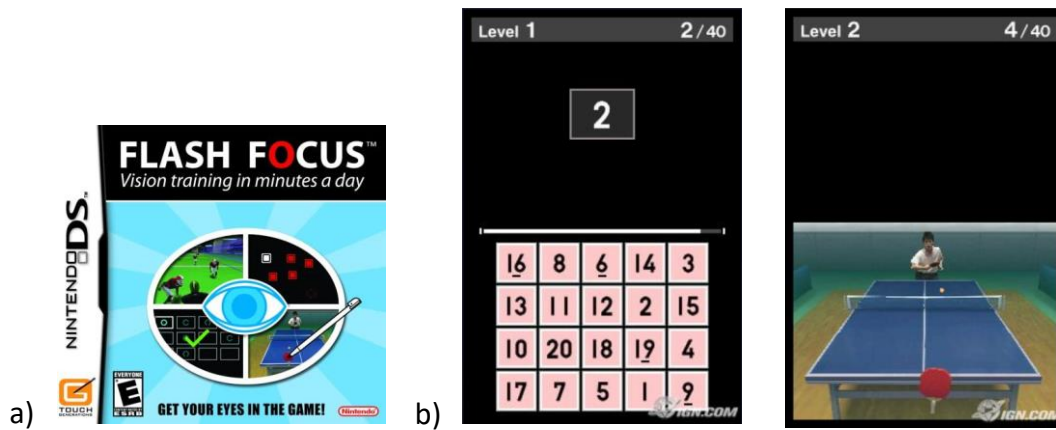


Figura 19. a) Flash Focus carátula, b) Dos minijuegos (imágenes extraídas de (Harris 2007))

Como ya se ha mencionado en la introducción, podemos encontrar en Steam el Juego Serio Experience: Colorblindness. Además de los escenarios interactivos en los que se pueden simular en tiempo real los distintos tipos de daltonismo (ver **Figura 9**), este juego permite la realización del test de Ishihara y el test de Fansworth. En la **Figura 20** se muestra minijuego relacionado con el test de Fansworth.



Figura 20. Imagen del test de Fansworth en el juego serio Experience: Colorblindness (imagen extraída de (iNFINITE Production 2018))

A excepción de Experience: Colorblindness, no hemos encontrado otros videojuegos orientados en la detección o concienciación sobre daltonismos. Como conclusión del análisis de herramientas realizados en este apartado (y algunos Juegos Serios genéricos descritos en la introducción) identificamos una serie de características comunes que deberíamos incluir en nuestra solución:

1. **Detección precoz de defectos visuales:** mediante la realización de algún tipo de prueba detectar defectos visuales.
2. **Empatía y simulación:** es importante hacer sentir al jugador lo mismo que siente una persona con una deficiencia concreta.
3. **Aprendizaje y concienciación:** aprender acerca de los problemas de visión en el proceso de simulación y juego.
4. **Ejercicios y buenas prácticas:** realización, a modo de juego, de algún ejercicio que forme parte de una buena práctica durante la experiencia.
5. **Atractivo para el jugador:** motivación para entrar en la aplicación sucesivas veces y mejorar los aprendizajes.
6. **Dinámicas de juego:** disponer de unas mecánicas de juego interesantes que fomenten que el jugador mantenga su interés.
7. **Interactividad:** dotar al usuario de posibilidades de interacción y manipulación de las deficiencias trabajadas durante el juego.

En la Tabla 1 se relaciona cada aplicación analizada con estas características:

	Eye OK	Flash Focus	Experience: Colorblindness
Detección	Alta	Baja	Baja
Empatía y simulación	No	No	Si
Aprendizaje	Si	No	Si
Ejercicios	No	Si	No
Atractivo	Baja	Media	Muy baja
Dinámicas jugables	Medio	Alto	Bajo
Interactividad	No	No	Si

Tabla 1. Comparativa entre los juegos serios citados del cumplimiento de las características a destacar

En nuestro proyecto queremos cubrir la mayoría de estas características centrándonos en el problema de daltonismo. Además, ninguna de las soluciones analizadas cubre el problema de daltonismo de forma realmente divertida para el usuario. Mientras que Eye OK y Flash Focus sí que trabajan la diversión de los usuarios, ninguno de ellos se centra en el daltonismo. Por el contrario, Experience: Colorblindness

sí se centra en este problema, pero pese a ser una herramienta muy conseguida, detectamos en él una falta de diversión para el usuario ya que carece de minijuegos. En nuestro caso, queremos que nuestro proyecto sea un juego muy interactivo y “adictivo” con un *loop* jugable entretenido, de manera que el jugador lo sienta como un juego y no como una experiencia a la que solo accederías para probarla.

2.1.4 TIPOS Y CARACTERÍSTICAS DE LOS SISTEMAS DE RV

En este apartado algunos aspectos técnicos para tener en cuenta al desarrollar una aplicación de RV. Indirectamente, el término RV agrupa una serie de técnicas con diferentes objetivos que reúnen una serie de elementos y rasgos comunes. Esto permite categorizar los sistemas de RV en tres tipos:

- **Sistemas de sobremesa:** tienen un bajo nivel de inmersión física ya que usan la pantalla del ordenador para presentar el entorno. El usuario tiene libertad de movimientos y la capacidad para interactuar con el escenario y puede lograr una gran sensación de inmersión psicológica como se ha demostrado en videojuegos como *The Last Of Us* (Márquez 2013).
- **Sistemas proyectivos:** pretenden generar la inmersión mediante la proyección de imágenes del espacio virtual sobre paredes reales del entorno. La mayoría de los simuladores de vuelo o conducción utilizan este sistema, en la cual el usuario está sentado en una réplica del vehículo a simular sobre el que se muestran las imágenes del mundo virtual. CAVE, proyecto creado en el Laboratorio de Visualización Electrónica de la Universidad de Illinois en Estados Unidos es un ejemplo representativo de este tipo de sistemas (CIS Tecnología e Diseño 2021).
- **Sistemas inmersivos:** tienen el objetivo de “engañar” por completo al usuario haciéndole pensar que está en un entorno completamente virtual. Para esto es imprescindible un casco de RV mediante el que proporcionarle las imágenes del mundo virtual y a su vez aislarle del mundo real. Una variante de este tipo de sistemas es la Realidad Aumentada, en este caso el casco aísla parcialmente y muestra objetos virtuales en el mundo físico. También pueden utilizarse objetos adaptados a la experiencia para una mayor inmersión (ver Figura 21).

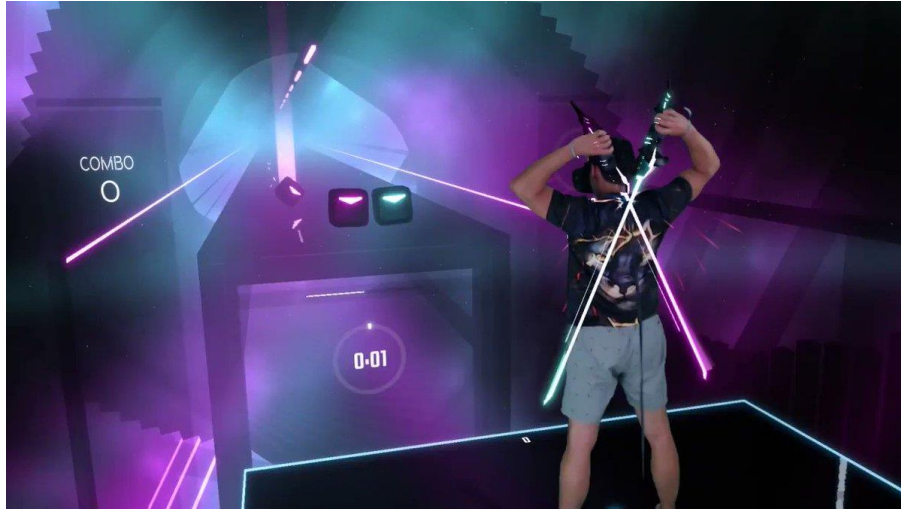


Figura 21. Imagen del Youtuber BennyDaBeast jugando al videojuego Beat Saber que es un sistema inmersivo de RV en el que se utilizan dos controladores adaptados (imagen extraída de (Moss 2019))

En lo que se refiere a interactividad en un mundo virtual podemos distinguir tres estadios según Diego Levis (Diego Levis 2006):

- **Pasivo:** son entornos inmersivos, pero no interactivos. En ellos tienes la capacidad de ver y oír lo que sucede, el entorno puede tener movimiento, pero este será forzado no permitiendo al usuario tener control sobre este.
- **Exploratorio:** tenemos características del estadio anterior, pero dotamos al usuario de control sobre su propio cuerpo. Este estadio es el usado en paseos arquitectónicos y otras de arte virtuales.
- **Interactivo:** permite no solo ver, oír y moverte por el escenario, sino que te permite interactuar con él llegando a provocar cambios sobre el entorno. La mayoría de los videojuegos de RV entrarían en este estadio.

Queremos que nuestro juego sea un sistema inmersivo e interactivo, ya que permitir interacciones entre el usuario y su entorno logrará una mayor concienciación sobre este tipo de problemas.

2.1.5 EVOLUCIÓN DE LOS DISPOSITIVOS DE REALIDAD VIRTUAL

La pregunta que nos hacemos en este punto es, ¿podemos con la tecnología actual llegar a un Juego Serio de RV inmersivo e interactivo? La respuesta a esta

pregunta nos lleva varias décadas atrás, donde encontramos una definición más específica de RV que la ofrecida por la RAE y descrita en la introducción. Nos referimos a la definición que dieron los autores Aukstakalnis y Blatner en 1993 quienes definen la RV como “una forma humana de visualizar, manipular e interactuar con ordenadores y datos complejos” (Aukstakalnis y Blatner 1993). En línea con esta definición, analizamos la evolución de soluciones de RV inmersiva comenzando nuestro análisis en un producto de la conocida compañía Nintendo.

De forma coetánea a la salida del libro de Aukstakalnis y Blatner, Nintendo desarrolló en Japón la consola Virtual Boy (ver Figura 22 a)) como parte del Proyecto VR32 (Harris 2014; Wikipedia 2022c). Esta consola pretendía conseguir una experiencia inmersiva en RV mediante el uso de un casco con una pantalla LED estereoscópica. La consola fue un desastre de ventas debido a su alto precio y su pantalla monocromática, siendo este un producto adelantado a su época. Tal fue el fiasco de la compañía nipona con este proyecto, que su interés en la RV ha provocado su recelo hacia esta tecnología hasta la aparición en 2012 de las Oculus Rift (ver Figura 22 b)).



Figura 22. a) Virtual Boy, la consola fallida de Nintendo con tecnología VR (imagen extraída de (Fantasymundo 2020)). b) Primer prototipo conocido de las Oculus Rift (imagen extraída de (Kickstarter 2016))

Como se mencionó en la introducción, el éxito actual de la RV está sobre todo relacionado con el ámbito de los videojuegos. Destacamos la popularidad de dispositivos como (1) PlayStation VR, (2) HTC Vive, (3) Valve Index, (4) StarVR o (5) FOVE VR entre otros. Hoy en día, y a pesar de tener casi cuatro años de vida, las Index de Valve siguen siendo de los dispositivos más potentes (Meirsschaut 2022; Versus 2022). Otro modelo que destacar sería las Meta Quest 2, descendientes de las Oculus Rift originales, que en su versión actual han evolucionado para dotar al usuario de un funcionamiento *stand-*

alone sin necesidad de estar conectado con cables a un dispositivo principal. Las futuras PlayStation VR2 también prometen grandes mejoras respecto su antecesor como nuevas funcionalidades como *eye-tracking* (Meta 2022b; PlayStation 2022).

A modo de resumen, se incluyen en la Tabla 2 las diferentes características de los cascos de RV más actuales:

	PS VR	HTC Vive	Index	StarVR	Quest 2
Standalone	No	No	No	Si	Si
Peso (g)	610	550	810	380	503
Campo de visión (º)	100	110	130	210	100
Tasa de refresco (Hz)	120	90	144	90	90
Tamaño de pantalla (")	5,7	3,8	7	10,1	5,46
Resolución (px)	1920 x 1080	2160 x 1200	2880 x 1600	5120 x 2880	3664 x 1920
Seguimiento	Si	Si	Si	No	Si

Tabla 2. Comparativa de características de una selección de cascos de VR por su popularidad

2.1.5.1 OCULUS RIFT S

Como ya se ha comentado, para el desarrollo de este proyecto disponemos de unas Oculus Rift S como las presentes en la Figura 23. Este modelo vio la luz en 2019 y fue un remplazo de las Oculus Rift originales. Tienen una tasa de refresco de 80 Hz, una resolución de 1280 x 1440 en cada ojo y otorga un campo de visión (fov) de 88º.



Figura 23. Fotografía de las Oculus Rift S (imagen extraída de (Amazon 2022))

2.2 ANÁLISIS DE TECNOLOGÍAS DE DESARROLLO SOFTWARE

2.2.1 MOTORES DE VIDEOJUEGOS Y DESARROLLO EN RV

Un motor de videojuegos se define como un conjunto de herramientas software para agilizar su desarrollo aportando al programador los siguientes elementos: un entorno de desarrollo integrado (IDE), así como un generador de imágenes (ya sea 2D o 3D), un motor físico (gravedad, masa, fricción, etc) y otras funciones que facilitan el desarrollo.

Actualmente, existen dos motores que gozan de la mayor popularidad y que analizamos a continuación: Unity y Unreal Engine. Ambos motores permiten desarrollar soluciones multiplataforma, es decir, videojuegos que pueden ser ejecutados en diferentes tipos de dispositivos. Será el propio motor quien se encarga de resolver las peculiaridades de cada dispositivo, abstrayendo al desarrollador de aspectos o cambios de bajo nivel hardware.

2.2.1.1 UNITY vs UNREAL ENGINE

Unity (Unity 2022a) es un motor de videojuegos multiplataforma lanzado en junio de 2005, siendo la versión 2022.1.13 la más reciente. Este motor es una aplicación gratuita escrita en C++, que hace uso de C# como lenguaje de scripting y de la que es propietaria la compañía Unity Technologies.

Es capaz de generar juegos tanto en 2D como en 3D, para distintas plataformas como PC, Switch, Xbox Series, Xbox One, PS5, PS4, Android, iOS, IOT, Oculus, etc. Sin duda es uno de los motores de videojuegos más versátiles hoy en día (Wikipedia 2022a). Además, posee una interfaz modular como se muestra en la **Figura 24** que lo hace muy cómodo de utilizar y personalizar.

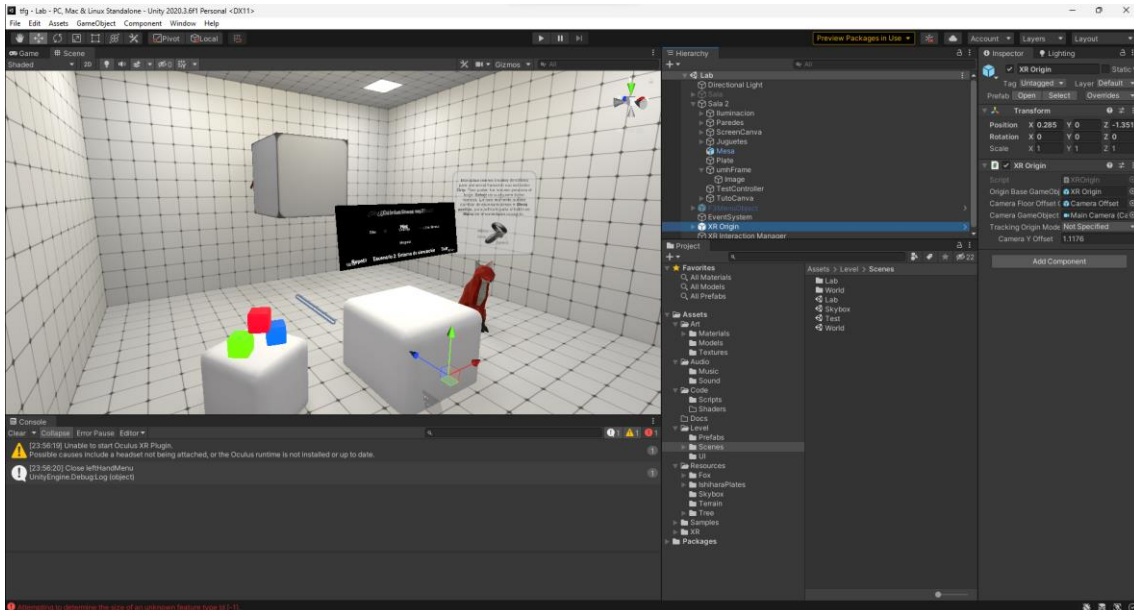


Figura 24. Interfaz Unity (imagen propia)

Por otro lado, Unreal Engine (Epic Games 2022) hizo su aparición en 1998 de la mano de Epic Games y su última versión estable es la 4.27 (publicada a finales de agosto de 2021). Ya se ha anunciado de forma oficial la llegada de Unreal Engine 5 en 2024. Al igual que Unity genera aplicaciones multiplataforma, es gratuito y es también ampliamente personalizable como se puede ver en la Figura 25.



Figura 25. Interfaz de Unreal Engine (imagen extraída de (Epic Games 2022))

Al igual que Unity, Unreal también está escrito en C++ y como lenguaje de scripting ofrece al usuario C++ y un sistema de Blueprints nativo, que es un sistema de programación por bloques. Actualmente puede generar aplicaciones para Windows, PS4 y PS5, Android, Oculus, iOS, etc. Unreal es actualmente uno de los motores con mayor capacidad gráfica, y esta será todavía superior con su versión 5 (Wikipedia 2022b). Desde el punto de vista de su instalación, Unity es más sencillo de instalar en los diferentes sistemas operativos que Unreal. Por ejemplo, la instalación de Unreal en sistemas Linux requiere descargar su código fuente desde el repositorio de GitHub y compilarlo directamente.

En la Tabla 3 se muestra una tabla comparativa que recoge las principales diferencias entre estos dos motores de desarrollo de videojuegos:

	Unity	Unreal Engine
Público objetivo	Enfocado a desarrolladores indie	Enfocado a juegos AAA
Aprendizaje	Una curva muy poco pronunciada la cual se ameniza con la gran cantidad de documentación y de tutoriales disponibles.	Tiene una curva muy inclinada la cual han suavizado en los últimos años con la creación de documentación y tutoriales
Soporte 2D	Sin problemas	De forma limitada
Soporte 3D	Sin problemas	Sin problemas
Lenguaje predilecto	C#	C++ y Blueprints
Apartado gráfico	Decente, con un buen trabajo de iluminación obtienes buenos resultados	Completamente fotorrealista en algunas ocasiones, muy buenos resultados con pocos conocimientos
Multijugador	Nativo en desarrollo, plugins <i>third-party</i> disponibles	Nativo
Rendimiento	Decente	Excelente

Tabla 3. Comparativa entre Unity y Unreal (Tristem 2022)

Para realizar este proyecto me he decantado por Unity principalmente porque a pesar de tener experiencia con ambos, cuando utilice Unreal aún no habían puesto a disposición gran parte de la documentación por lo que mis conocimientos sobre este son mucho más limitados que los de Unreal.

2.2.1.2 SDKS DE DESARROLLO PARA OCULUS RIFT S

Merece la pena destacar que Unity y Unreal requieren de librerías específicas que ofrezcan funciones para controlar dispositivos hardware avanzados como las Oculus Rift S. Por ejemplo, las Oculus Rift S disponen de dos mandos con seis botones y un *joystick*, que deberemos poder manipularlos desde el código que programemos a través de eventos y funciones específicas de este desarrollador. Para ello, el motor deberá ser compatible y tener instalados los SDKs necesarios para integrarse con este hardware, compilar su código y generar los ejecutables compatibles.

En el caso concreto de la programación para Oculus debemos instalar los plugin de XR Interaction Toolkit y XR Plugin Management. Una vez hecho esto, desde el menú de *File > Build Settings > Player Settings > XR Plug-in Management* deberemos marcar la opción Oculus, como se muestra en la Figura 26. Para realizar una *build* para PC solo hay que ir a *File > Build Settings* y presionar en el botón *Build*. Como vemos, gracias al SDK el uso de Oculus desde Unity queda bien integrado en el motor y es sencillo de utilizar.

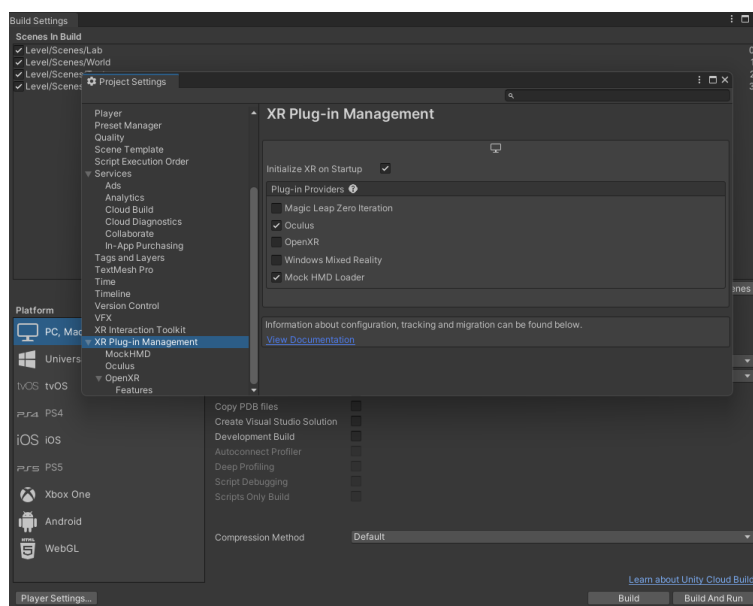


Figura 26. Submenú de opciones del usuario dentro del menú de opciones de la *build*. Concretamente configuración del XR Plug-in Management (imagen propia)

2.2.2 MODELADO DE OBJETOS 3D

El desarrollo de un videojuego requiere de desarrollo 3D para implementar los escenarios y los objetos con los que el usuario deberá interactuar. Para la creación del entorno tridimensional se ha hecho uso de modelos generados en Blender 3D.

- **Blender 3D** (Blender 2022a): es un potente software que permite realizar diversas acciones como el modelaje tridimensional, animación, simulación de fluidos, *rigging*, etc. Con esta herramienta de software libre desarrollada en C, C++ y Python se han realizado numerosas películas de animación (Darkcritz 2019). Blender fue creada en 2002 por desarrollador de software de procedencia holandesa Ton Roosendaal, quien ganó previamente diversos premios con el estudio de animación NeoGeo (precursor de Blender) (Blender 2022b). Esta herramienta dispone de una interfaz cargada de menús que puede resultar abrumadora para usuarios nóveles (ver Figura 27).

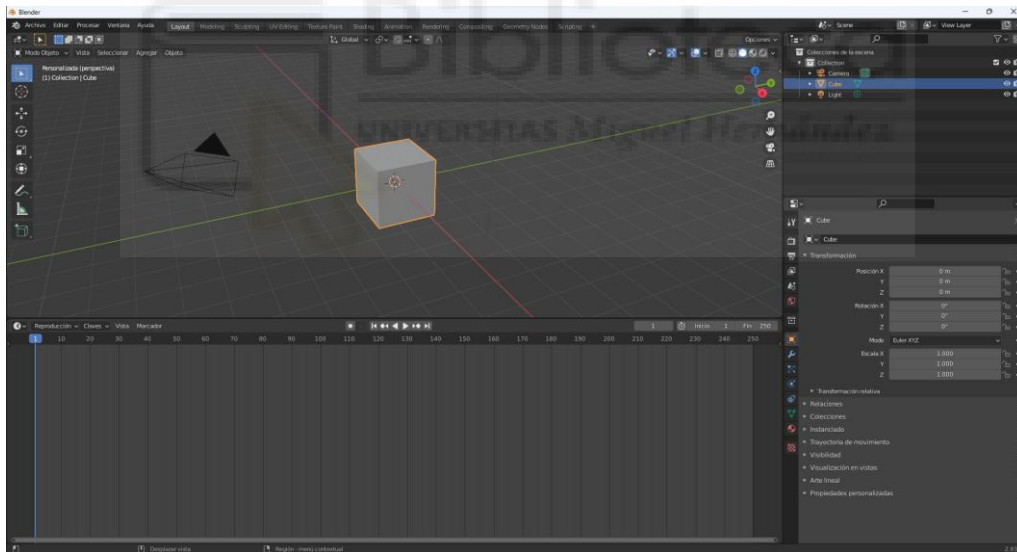


Figura 27. Imagen donde se muestra de la interfaz de Blender (imagen propia)

- **Sketchfab** (Sketchfab 2022): es un sitio web donde se pueden compartir modelos 3D. Proporciona un visualizador de modelos basado en tecnología WebGL permitiendo su visualización tanto en escritorio como en dispositivos móviles. El visualizador que utiliza puede ser embebido en sitio web externos, Facebook entre ellos (Sketchfab 2015). Desde finales de 2014, Sketchfab permite a sus usuarios dejar sus modelos disponibles para la descarga bajo

licencias de Creative Commons (Sketchfab 2014). Esta web dispone de múltiples opciones de gran calidad como se puede apreciar en la Figura 28.

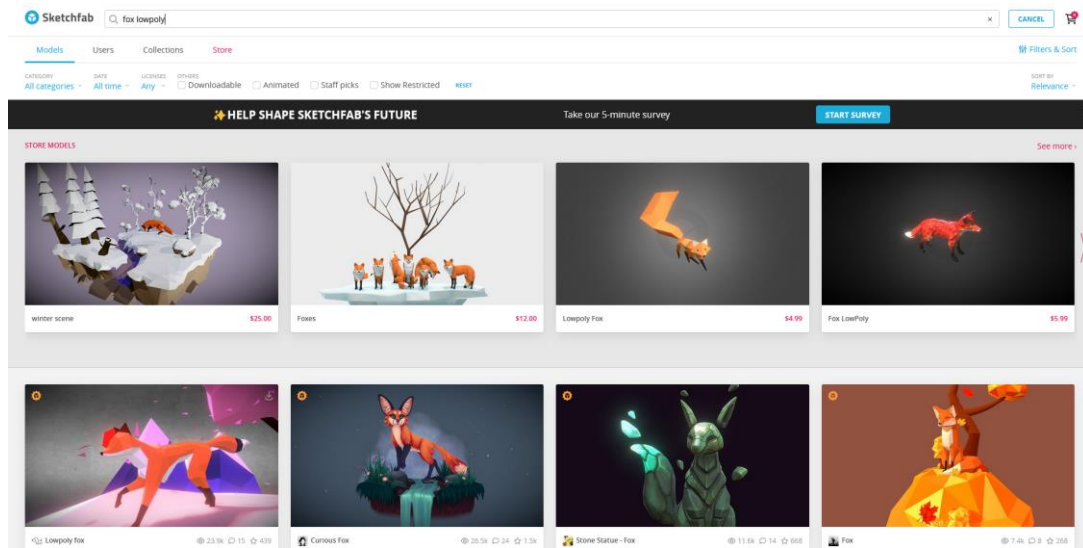


Figura 28. Resultados de la búsqueda de “fox lowpoly” en Sketchfab (imagen extraída de (Sketchfab 2022))

Para nuestro proyecto vamos a reutilizar modelos 3D de esta web ya que hay una gran disponibilidad y tengo experiencia previa en descargarlos y editarlos para crear versiones de estos personalizadas. Del mismo modo, se ha utilizado Blender porque tiene total compatibilidad con Unity, permite editar los objetos 3D descargados de Sketchfab y he trabajado previamente con él.

2.2.3 IMÁGENES Y FOTOGRAFÍAS 360º

Los objetos 3D anteriores permiten recrear mundos 100% virtuales con los que el usuario pueda interactuar. Sin embargo, en este punto nos preguntamos si sería posible crear una inmersión del usuario en un mundo virtual que represente la realidad. Para ello encontramos la opción de utilizar fotografías 360º. Tenemos varias opciones para obtener este tipo de recursos:

- **Pixexid** (Pixexid 2022): es un banco de imágenes gratuito y que distribuye recursos bajo una licencia Creative Commons Public Domain (Creative Commons 2022). Por ejemplo, en la Figura 29 se muestran los resultados ofrecidos por Pixexide al buscar “imágenes 360º”.

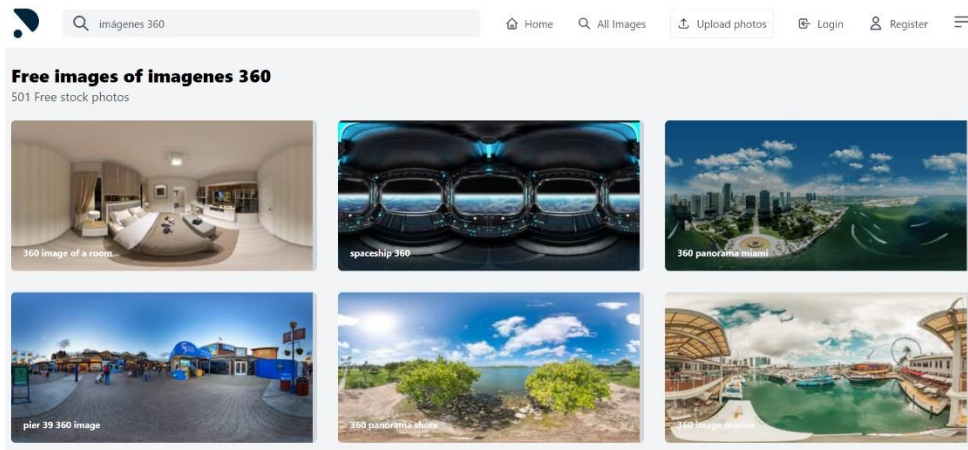


Figura 29. Resultados de buscar “imágenes 360” en el banco de imágenes Pixeid (imagen extraída de (Pixeid 2022))

- **Hugin** (Hugin 2021): permite la combinación de imágenes superpuestas para formar panorámicas, corrección de panorámicas, etc. Se trata de un software libre y gratuito que está disponible en Windows bajo la licencia GPL (GNU 2022) en su página web. En la Figura 30 se muestra el uso de Hugin para generar una imagen 360º de una de las plazas del campus de Elche de la Universidad Miguel Hernández (UMH).

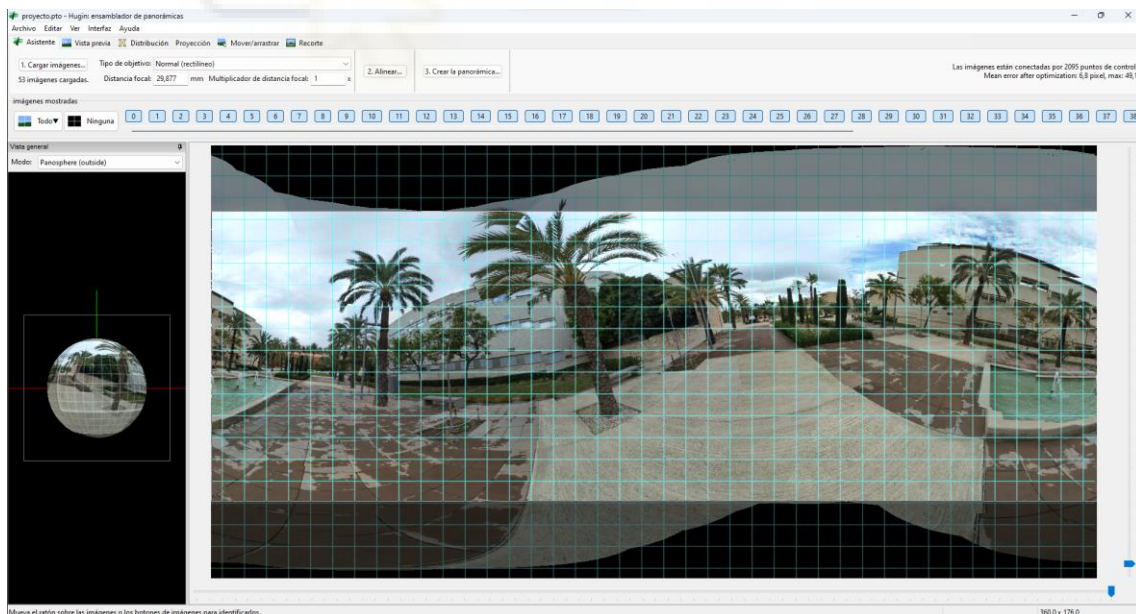


Figura 30. Ejemplo de uso de Hugin para la realización de una imagen 360º a partir de imágenes propias (imagen propia)

Hemos escogido estas dos herramientas en nuestro proyecto debido a sus buenos resultados, su modo de distribución y su pequeña curva de aprendizaje. En el caso de Pixeid nos ha permitido incluir en nuestro juego una fotografía 360º de alta calidad, similar a la que podríamos realizar con una cámara 360º que no hemos podido adquirir por falta de presupuesto. Para suplir esta carencia, hemos usado Hugin para generar una imagen 360º relacionada con la UMH.

Por último, hemos analizado si es posible incluir imágenes 360º en Unity y hemos encontrado que lo permite perfectamente de forma nativa (Unity 2022c).

2.2.4 SIMULAR DALTONISMO: MODIFICAR CÁMARA Y TEXTURAS

En Unity uno de los componentes que representa los ojos de los jugadores son las cámaras (Unity 2022b). Como se puede apreciar en la Figura 31 la cámara muestra al jugador una imagen completamente configurable y que puede no corresponderse con la que se ve durante el desarrollo de la escena.

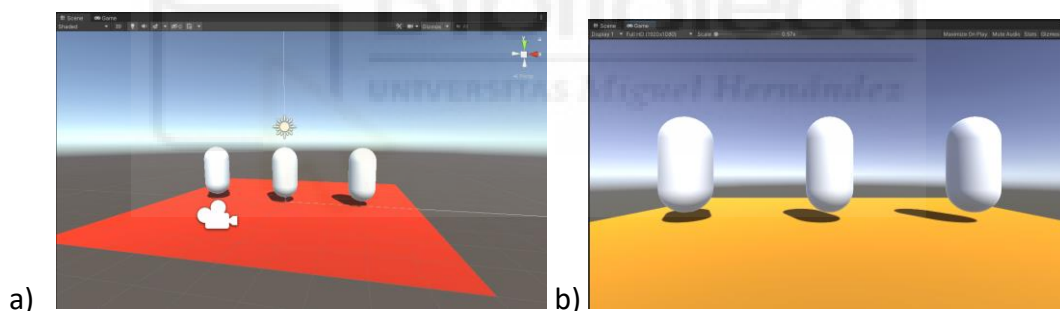


Figura 31. Capturas propias siendo a) captura del escenario real y b) captura del resultado que muestra la cámara tras las modificaciones (imágenes propias)

Dado que todo lo que ve el jugador pasa por los ojos de la cámara, se consideró que una opción razonable y adecuada era modificarla de manera que sea esta la que simule directamente los daltonismos. Pronto nos encontramos con un nombre conocido, Alan Zucconi, creador del juego The Witness en 2016 (The Witness 2016). Zucconi, en su web, publicó un artículo de buenas prácticas para tener en cuenta en el desarrollo de videojuegos que sean accesibles a personas con distintos tipos de daltonismos (Zucconi 2015). En este post nos incita, por ejemplo, a utilizar algún identificador visual para los colores como ColorADD si van a tener importancia en el

juego. Además de esto presenta una tabla (ver Figura 32) donde muestra cómo se pueden alterar los colores primarios según los distintos tipos de daltonismos. Por si esto fuera poco, pone a disposición del lector un script para realizar las modificaciones de cámara y simular los daltonismos en Unity.

Type	%	Red Channel			Green Channel			Blue Channel		
		R	G	B	R	G	B	R	G	B
Normal	92%	100%	-	-	-	100%	-	-	-	100%
Protanopia	0.59%	56.667%	43.333%	-	55.833%	44.167%	-	-	24.167%	75.833%
Protanomaly	0.66%	81.667%	18.333%	-	33.333%	66.667%	-	-	12.5%	87.5%
Deuteranopia	0.56%	62.5%	37.5%	-	70%	30%	-	-	30%	70%
Deuteranomaly	2.7%	80%	20%	-	-	25.833%	74.167%	-	14.167%	85.833%
Tritanopia	0.016%	95%	5%	-	-	43.333%	56.667%	-	47.5%	52.5%
Tritanomaly	0.01%	96.667%	3.333%	-	-	73.333%	26.667%	-	18.333%	81.667%
Achromatopsia	<0.001%	29.9%	58.7%	11.4%	29.9%	58.7%	11.4%	29.9%	58.7%	11.4%
Achromatomaly	<0.001%	61.8%	32%	6.2%	16.3%	77.5%	6.2%	16.3%	32.0%	51.6%

Figura 32. Tabla que muestra cómo se ven representados los distintos colores en función del tipo de visión (tabla extraída de (Zucconi 2015))

Además, una alternativa a la modificación de la cámara sería modificar las texturas de los materiales del mundo 3D que estemos utilizando. A partir de la información de la figura anterior, se pueden usar los valores en porcentaje de la representación que de los distintos colores que ven los daltónicos para crear un sistema que dado un material modifique su color para que se adecúe al daltonismo elegido.

2.2.5 ENTORNOS DE DESARROLLO

Un entorno de desarrollo integrado o IDE es un software para el diseño de aplicaciones que combina herramientas de uso común para el desarrollo en una misma interfaz (Red Hat 2019). Por norma general un IDE cuenta con:

- **Editor de código:** suele tener ayudas para el desarrollador como autocompletado, resaltado de sintaxis, detección de errores en tiempo real, refactorizado, etc.

- **Herramientas de automatización:** para automatizar tareas repetitivas como la compilación y creación de un fichero ejecutable, así como su posterior ejecución con una serie de parámetros determinados.
- **Depurador:** herramienta que nos ayuda a detectar los errores. Ofrecen una ejecución por bloques delimitados por *breakpoints* y permite consultar el estado de las variables en tiempo de ejecución.

Los editores de código suelen formar parte de sus hermanos mayores, los IDE, aunque son más ligeros que ellos ya que solo disponen de lo necesario para ser productivo. A menudo, requieren la instalación de extensiones o *plugins* que doten a nuestro editor de funcionalidad específica (Pingüino Digital 2020). A continuación, analizamos los entornos de desarrollo y editor que he utilizado con anterioridad y sé que disponen de integración con Unity para facilitar el desarrollo.

2.2.5.1 MICROSOFT VISUAL STUDIO, RIDER Y VSCODE

Microsoft Visual Studio (Microsoft 2022b) está disponible únicamente en Windows, y soporta de forma nativa 36 lenguajes de programación. Este software lleva con nosotros desde 1997 habiendo pasado por muchos cambios a lo largo de su historia; a fecha de la redacción de este documento va por su versión 17.3.6 de 2022, mostrada en la Figura 33. Actualmente su modelo de negocio es *freemium* por lo que algunas de sus características, no esenciales, aunque sí atractivas, requieren del pago de una licencia. Este IDE hace uso del conjunto de herramientas de IntelliCode (Microsoft 2022d) para la finalización automática de código. Tiene numerosas funcionalidades online como son las sesiones de colaboración en tiempo real de Live Share (Microsoft 2022e) que permiten la edición a distintos usuarios del mismo fichero de forma simultánea. Además de esto, tiene integrado un sistema de control de versiones basado en Git (JetBrains 2022d; Microsoft 2022c; Quiroa 2019).

Una opción similar a la anterior es la de Rider (JetBrains 2022c) propiedad de JetBrains. JetBrains es una empresa de origen checo con sede en Praga que tiene una amplia gama de entornos de desarrollo especializados en distintos lenguajes de programación. Rider destaca por su uso la interfaz, así como múltiples funcionalidades básicas de IntelliJ IDEA (JetBrains 2022a), entorno de desarrollo para Java de los mismos

autores. Gracias a esto una vez estas familiarizado con los editores marca JetBrains cambiar de uno a otro no es ningún problema ya que todos tienen una interfaz similar y disponen de comandos idénticos, mostrada en la Figura 34. Además de esto dispone de la herramienta ReSharper que la dota de más funciones como refactorización, inspección de código, navegación mejorada, etc. A diferencia Visual Studio este entorno de desarrollo sí es multiplataforma (JetBrains 2022b, 2022d).

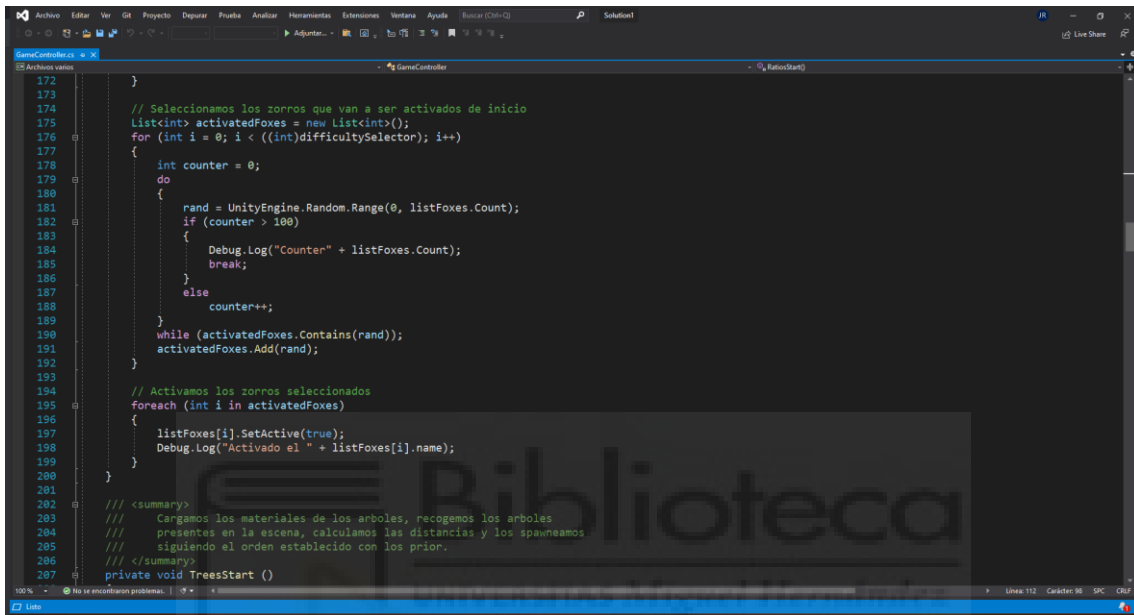


Figura 33. Interfaz de Visual Studio (imagen extraída de (Microsoft 2022b))

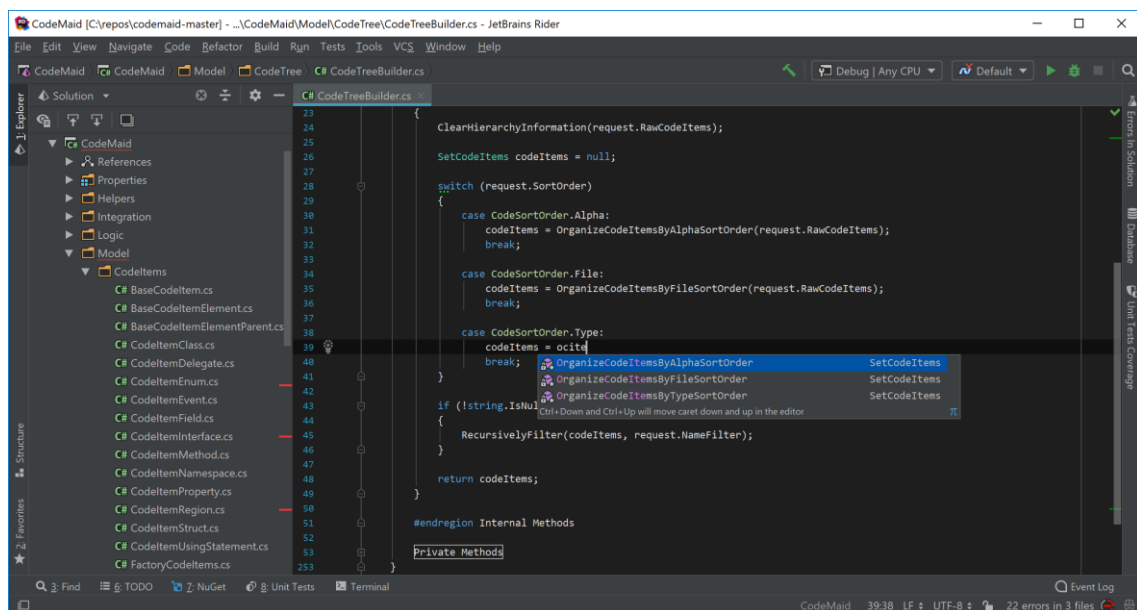


Figura 34. Interfaz de Rider (imagen extraída de (JetBrains 2022c))

Por último, Visual Studio Code (VSCode) (Microsoft 2022a) es un editor de código ligero de la mano de Microsoft. En este caso la herramienta es completamente gratuita, *Open Source* y multiplataforma. Al igual que su hermano mayor, permite programar en gran cantidad de ofreciendo autocompletado y marcado de errores sintácticos gracias al uso de IntelloSente. A pesar de no ser un IDE, tiene incorporadas herramientas de *debug* y Git. Además, tiene una gran comunidad detrás que lo provee de extensiones tanto de funcionalidad como de apariencia y personalización (ver Figura 35).

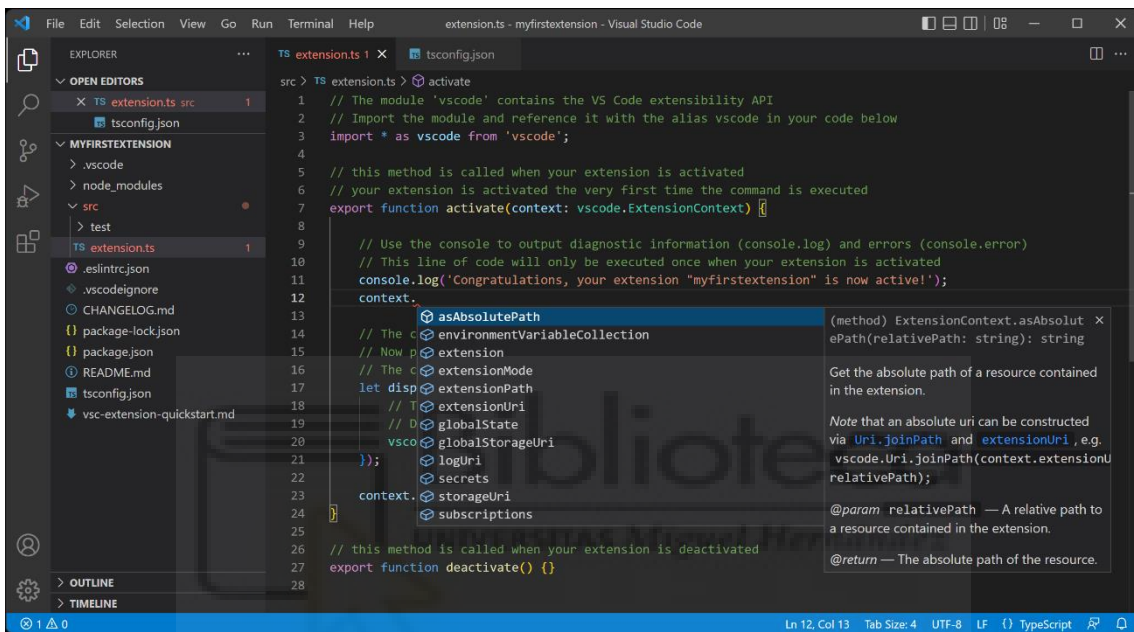


Figura 35. Interfaz de VSCode (imagen extraída de (Microsoft 2022a))

En este caso, me he decantado por Visual Studio porque sabía de antemano que la integración con Unity era más completa que la que ofrece VSCode. Rider se estuvo utilizando en el pasado y no generó problemas, pero tras formatear y reinstalar Unity instaló automáticamente Visual Studio por lo que entendemos que es el recomendado.

2.2.6 OTRAS HERRAMIENTAS

Además de las herramientas analizadas hasta este punto, se han utilizado otros programas y plataformas que han ayudado en la realización de este proyecto:

- **GitKraken** (GitKraken 2022): utilizado para llevar a cabo el control de versiones de código fuente de la aplicación desarrollada. GitKraket es un

cliente de Git multiplataforma que permite llevar de forma visual un gran control sobre las ramas del proyecto mediante líneas de distintos colores como los mostrados en la Figura 36.

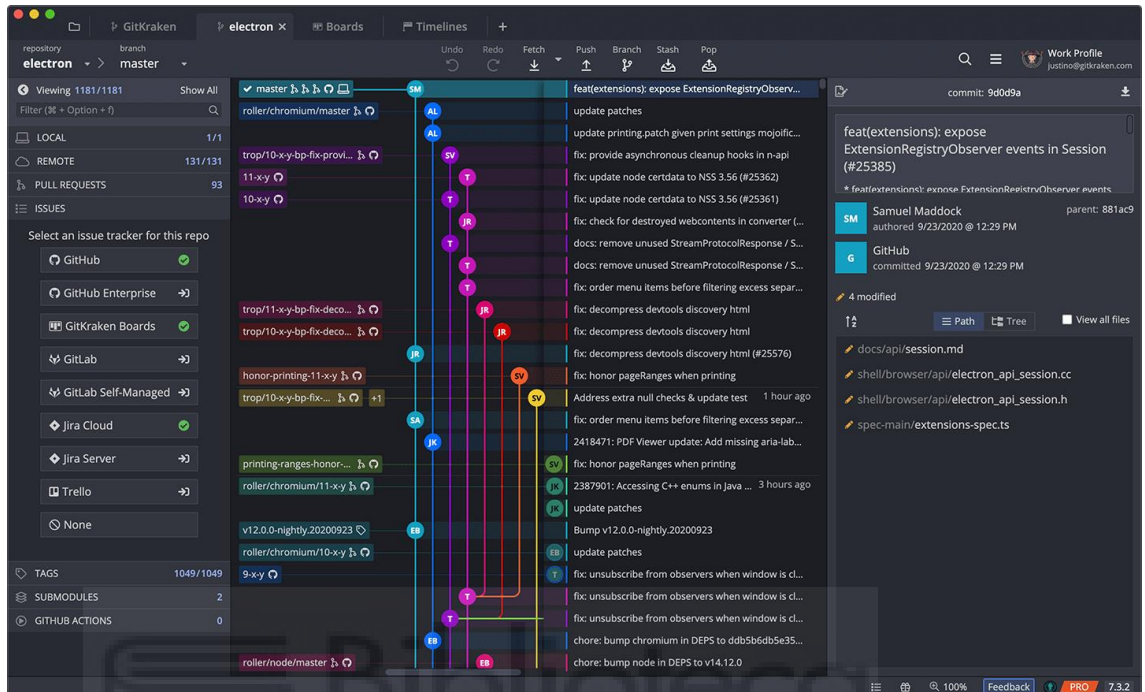


Figura 36. Interfaz de GitKraken en la que se puede apreciar el control de ramas visual que tiene (imagen extraída de (Isaac 2022))

- **Bitbucket** (Bitbucket 2022): utilizado para alojar el repositorio GIT del control de versiones de nuestro proyecto. Es una alternativa a GitHub, pero nos decantamos por Bitbucket porque permite un tamaño máximo de archivo mayor que GitHub que en proyectos como el nuestro es un factor importante.
- **TickTick** (TickTick 2022): utilizada para la gestión propia del tiempo y las tareas en las que se ha organizado el proyecto. Se trata de una aplicación para la gestión de tareas. Tiene un planteamiento sencillo en el que dejas las tareas en una lista de entrada, *inbox*, y posteriormente escoges una fecha de realización de todas ellas. Esto permite una implementación más sencilla del sistema GTD utilizado para la realización de este proyecto, lo cual se explicará con detalle en el apartado 3.1.

2.3 PROPUESTA DE SOLUCIÓN

Para el desarrollo de la aplicación ha sido necesario el uso de varios dispositivos de hardware con los que trabajar:

- **Ordenador personal:** ordenador de sobremesa en el que se ha desarrollado la aplicación.
- **Oculus Rift S:** dispositivo cedido por la Escuela Politécnica Superior de Elche (EPSE) de la Universidad Miguel Hernández (UMH) donde se han realizado las pruebas reales y la calibración de la aplicación VR.
- **Ordenador CIO:** se ha utilizado un ordenador cedido por el Centro de Investigación Operativa (CIO) para realizar todas las pruebas y desplegar la aplicación en las Oculus.

Tras exponer las tecnologías a usar y los dispositivos disponibles, se muestra en la Figura 37 un esquema de la relación entre los dispositivos y tecnologías usadas.

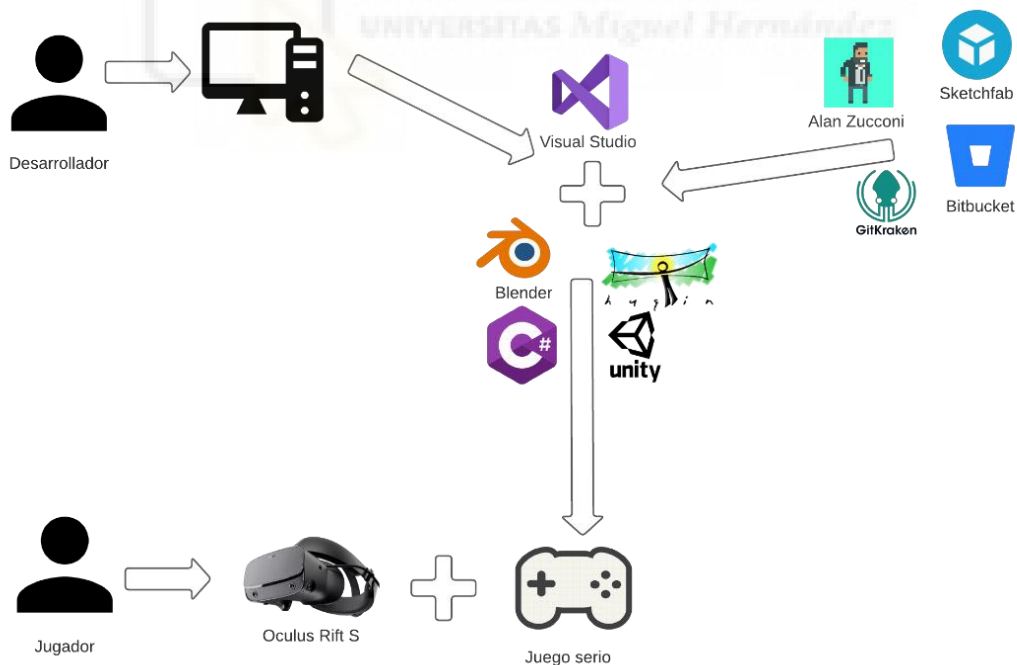


Figura 37. Diagrama con la propuesta de hardware y software de nuestra solución (imagen propia)

3 METODOLOGÍA Y RESULTADOS

En este capítulo, se van a tratar aspectos técnicos relacionados con la implementación del proyecto, desde su planificación y metodología hasta algunos aspectos relevantes de ingeniería de software y el juego implementado.

3.1 METODOLOGÍAS ÁGILES

Por definición, las metodologías ágiles son aquellas que permiten una rápida adaptación de los métodos de trabajo a las condiciones cambiantes del proyecto y su desarrollo. A diferencia de los métodos tradicionales, las metodologías ágiles evitan planificar el proyecto con pocas iteraciones largas que den lugar a una primera y casi última versión “finalizada”. Para ello este tipo de metodologías proponen pequeñas iteraciones en las que cada una de ellas incluye un ciclo de vida completo con planificación, análisis de requisitos, diseño, codificación, pruebas y documentación (Garrido Sotomayor 2022).

A continuación, destacamos algunas metodologías ágiles que han alcanzado altos niveles de popularidad en los últimos años:

- **Kanban:** consiste en la elaboración de un diagrama con tres columnas en las que se clasifican las tareas según están pendientes, en proceso o terminadas. Esto permite una mayor organización para los trabajadores ya que en todo momento pueden ver de forma gráfica el estado del proyecto en su totalidad. Asimismo, esto permite que distintos trabajadores no seleccionen la misma tarea por error (Lasa Gómez, Álvarez García, y Heras del Dedo 2017; Shingo y Dillon 1989).
- **Extreme Programming (XP):** potencia la relación entre los trabajadores y los clientes permitiendo así una mayor adaptabilidad en detrimento de la previsibilidad. Este aumento de la interacción trabajador-cliente produce cambios de requisitos sobre la marcha lo cual es uno de sus puntos fuertes según sus defensores (Beck 2000).

- **SCRUM:** es un método pensado para grupos de trabajadores, aunque el flujo de trabajo general es aplicable a un solo trabajador. En líneas generales se toman los requisitos del cliente, se traducen a la funcionalidad real desde el punto de vista del desarrollo de software que está pidiendo el cliente, se realiza un sprint (ciclo de trabajo) tras el cual se realizan una serie de reuniones para verificar los requisitos cumplidos en el sprint actual, así como se incorporan nuevos requisitos de cara al siguiente sprint. Normalmente, existe una persona que toma el rol de Scrum Master que gestiona los cambios a realizar, otra con el rol de *Product Owner*, en representación de los interesados por el producto, y el *Team*, que sería el equipo de trabajo que se encarga del desarrollo (Lasa Gómez et al. 2017).
- **GTD:** de sus siglas en inglés *Getting Things Done*, GTD es un libro publicado en 2001 y escrito por David Allen que trata sobre un método para la gestión del tiempo de manera que no se deba tener en la cabeza una lista de tareas pendientes permitiendo una mayor concentración en la tarea actual.

A modo de resumen, GTD consiste en apuntar todas las tareas en una lista, las tareas realizables de forma rápida hacerlas directamente, una vez terminada la lista hay que organizarla en cuanto a fechas de realización, dependencias, etc. Cualquier idea nueva es vaciada en la lista de entrada y a final de la jornada procesada y clasificada. Cada día se deben realizar aquellas tareas planificadas para ese día. Tenemos, por tanto: (1) una "lista" de tareas pendientes, (2) una lista con las tareas que están realizándose (coincide con la planificación de hoy) y (3) opcionalmente, una lista de tareas realizadas (David Allen Company 2022).

GTD se podría entender como un Kanban (tenemos las mismas listas) pero aplicado a una persona, modificado y con una metodología de actuación más compleja. En este trabajo hemos decidido utilizar GTD porque es una metodología aplicable a proyectos de este tipo y que tengo implementada en mi día a día. A continuación, profundizamos más en esta metodología.

3.1.1 GTD: PRINCIPIOS BÁSICOS

Originalmente el sistema GTD se diseñó con el uso de carpetas en mente. Se propone el uso de 12 carpetas que representan los meses y 31 carpetas adicionales para los días como se muestra en la Figura 38. Al organizar las tareas se debe poner cada una en el día en el debería ser completada. Al inicio de cada día se debe mirar la carpeta actual, realizar sus tareas y organizar el resto si procede.



Figura 38. Ejemplo de aplicación de GTD mediante un sistema de carpetas físicas conocido como “43 folder system” (imagen extraída de (Middleton 2022))

GTD utiliza cinco principios fundamentales que son: recopilar, procesar, organizar, revisar y hacer. A continuación, se describen cada uno de ellos:

- **Recopilar:** para empezar, debemos eliminar de nuestra cabeza todas aquellas ideas y tareas que tenemos que realizar. Esto se consigue apuntándolas en listas que vaciaremos una vez al día.
- **Procesar:** una vez recopilada la lista debemos organizarla con las tareas pendientes siguiendo un orden concreto. (1) siempre se debe empezar por la primera tarea, (2) nunca se debe procesar de forma simultánea más de un elemento, (3) no se puede traer de vuelta ninguna tarea, (4) si un elemento requiere de una acción y es una acción corta se debe realizar en el momento siguiendo la regla de los dos minutos (“si la tarea tarda menos de dos minutos debes realizarla inmediatamente”, el número de minutos es orientativo), (5) si

es una acción que no tienes que realizar tú debes delegar, en otro caso la pospones, (6) si el elemento no requiere una acción debes archivarlo, desecharlo si no es procedente, ponerlo en cuarentena si no se puede realizar ahora o informar si no puedes delegarlo correctamente.

- **Organizar:** Allen describe algunas pautas para organizar las tareas:
 - **Acciones próximas:** para los elementos que deban ser tenidos en cuenta se decide cual es la siguiente acción para realizar y preferentemente el lugar donde se llevará a cabo.
 - **Proyectos:** se consideran así todas aquellas tareas que requieran de múltiples acciones. Los proyectos deben tener una tarea próxima realizable asociada y han de ser revisadas periódicamente para asegurar esto último.
 - **En espera:** aquí se mantienen las tareas que están esperando a que suceda algo de forma externa o bien porque depende de una tarea delegada o bien porque no se puede realizar aún.
 - **Algún día:** aquí se apuntan cosas que queremos realizar en un futuro, pero sin poner fechas, ya sea porque no se pueden realizar en este momento o porque no queremos.
 - También se debe tener un calendario en el que apuntar las cosas que se deben realizar de forma obligatoria en fechas específicas. Allen recomienda tener un sistema de recordatorios único, simple y ordenado de forma que permita buscar de forma rápida y fácil.
- **Revisar:** deberemos revisar la lista de cosas en espera y la que contiene las tareas recopiladas al menos una vez al día. Todo el trabajo es inútil si esta parte no se realiza correctamente. Y para esto se deben realizar las tareas en el orden que han sido ordenadas, así se evita realizar solo aquellas tareas que nos resulten más atractivas o veamos más fáciles de completar.
- **Hacer.** Realizamos las tareas en el orden correspondiente.

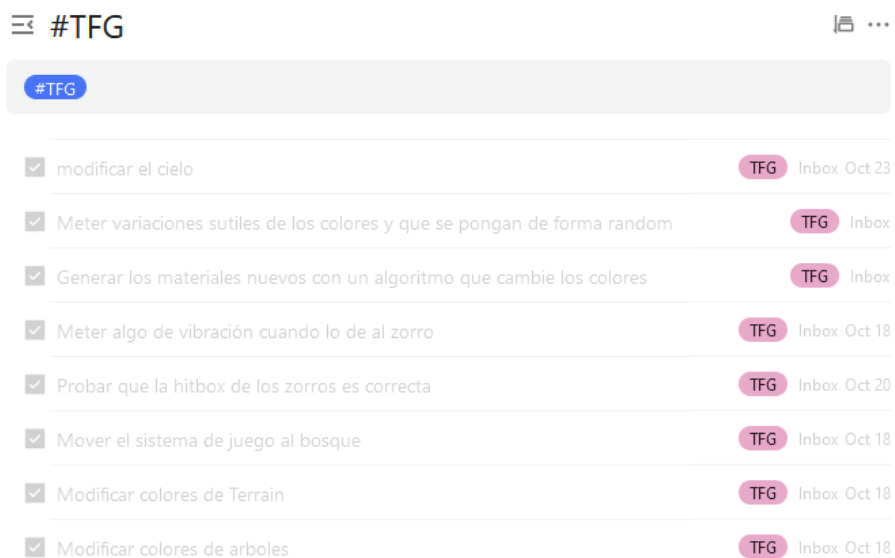
3.1.2 GTD: CRONOGRAMA Y LISTA DE TAREAS

El proyecto ha tenido una duración aproximada de 8 meses. En la Figura 39 se muestra un cronograma del trabajo realizado dividido en bloques de 14 días, momento en que se realizaba una revisión rutinaria del estado del proyecto.

	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15	B16	B17	
Bloque de trabajo investigación																		
Daltonismo	■	■																
Trabajos relacionados		■											■					
Unity						■	■											
Bloque de trabajo Unity																		
Diseño de niveles		■	■															
Desarrollo de Test				■														
Creación de modelos					■				■									
Sistema de cambio de materiales						■			■	■								
Sistema de modificación de cámara											■							
Loop Jugable							■	■	■									
Sistema de imágenes 360												■						
Bloque de trabajo memoria																		
Introducción	■																	
Materialiales y métodos													■	■				
Metodología y resultados															■	■		
Conclusiones y trabajo futuro																		■

Figura 39. Cronograma que muestra la distribución de trabajo en el tiempo del proyecto (imagen propia)

En la Figura 40 se muestra, a modo de ejemplo, una captura de mi aplicación TickTick con las tareas del Bloque de trabajo Unity con octubre como fecha de entrega.



<input checked="" type="checkbox"/>	Manus para las pruebas	TFG	Inbox Oct 18
<input checked="" type="checkbox"/>	Generar informe simple	TFG	Inbox Oct 18
<input checked="" type="checkbox"/>	Arreglar arboles	TFG	Inbox Oct 18
<input checked="" type="checkbox"/>	Arreglar terrain	TFG	Inbox Oct 18
<input checked="" type="checkbox"/>	Sacar las imagenes del test	TFG	Inbox Oct 18
<input checked="" type="checkbox"/>	Meter el zorro en el lab	TFG	Inbox Oct 18
<input checked="" type="checkbox"/>	Objetos no interactivables en lab	TFG	Inbox Oct 18
<input checked="" type="checkbox"/>	Iluminación en el lab	TFG	Inbox Oct 18
<input checked="" type="checkbox"/>	Cambiar de color a los zorros	TFG	Inbox Oct 18
<input checked="" type="checkbox"/>	Juguetes con emisive en colores primarios tonos pastel suave	TFG	Inbox Oct 18
<input checked="" type="checkbox"/>	Arreglar lo del editor	TFG	Inbox
<input checked="" type="checkbox"/>	Arreglar la hitbox de los zorros	TFG	Inbox Oct 18

Figura 40. Últimas tareas referentes a la programación del videojuego con Unity (imagen propia a partir de la aplicación TickTick)

En la Figura 41 se muestra la recreación de la tarea principal del desarrollo en Unity incluyendo todas sus tareas.

The screenshot shows the TickTick application interface. On the left, there is a sidebar with filters: 'Completed', 'Won't Do', and 'Trash'. The main area displays a task list for '#TFG'. The 'Unity' task is expanded, showing a hierarchical list of sub-tasks. Each task has a checkbox and a 'TFG' tag. The sub-tasks under 'Unity' are: Laboratorio (with sub-tasks: Buscar placas, Implementar test, Creación de escenario), Aplicación daltonismos (with sub-tasks: Funcionamiento materiales y texturas), Desarrollo sistema cambio materiales (with sub-tasks: Cambio a sistema sin random, Montar escenario), Simulación (with sub-tasks: Modelos, Sistema de juego), Visor 360 (with sub-tasks: Imágenes (with sub-tasks: Búsqueda de imagenes de prueba, Creación de imagenes), and Creación de TutoPanel), and Sistema loop.

Figura 41. Recreación del estado inicial de la tarea “Unity” (imagen propia a partir de la aplicación TickTick)

3.2 ANALISIS Y DISEÑO DEL SOFTWARE

3.2.1 MAPA DE NAVEGACIÓN Y ESCENARIOS DEL JUEGO

En la Figura 42 se muestra el mapa de navegación del juego serio para el que se han definido tres escenarios que trabajan diferentes conceptos:

- **Escenario 1:** laboratorio para el diagnóstico a través del test de Ishihara.
- **Escenario 2:** minijuego interactivo de simulación.
- **Escenario 3:** concienciación con imágenes 360°.

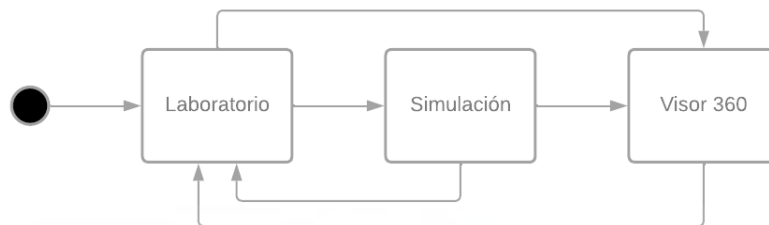


Figura 42. Mapa de navegación de la aplicación (imagen propia)

Se propondrá al usuario una experiencia guiada a través de estos tres escenarios. Por defecto, el jugador deberá realizarlos en orden, aunque se le ofrecerá de saltar entre ellos sí lo considera necesario. La idea es que el jugador realice en el primer escenario un test que le permita valorar su visión. En caso de que su visión sea correcta, se le ofrecerá continuar con el escenario 2 en el que a modo de minijuego se la hará sentir como una persona con distintos tipos de daltonismos a través de la manipulación de un entorno 100% virtual. Por último, una vez el jugador ha empatizado con este tipo de problemas, se le llevará al escenario 3 en el que se verá inmerso en el entorno de fotografías 360° que le permitirán repetir las simulaciones del escenario dos, pero en este caso con imágenes que representan su realidad de una forma más cercana.

3.2.2 CASOS DE USO

Un caso de uso es una lista de acciones que indican como debería actual el sistema para desarrollar una tarea concreta cuando es utilizado por un usuario o con otro subsistema. Los principales elementos de los casos de uso son: (1) actor, quién o

qué está usando el sistema, (2) descripción, explicación corta de las operaciones que el actor debe ser capaz de realizar en el caso de uso, (3) precondition, una condición que debe cumplirse antes de darse el caso de uso, (4) postcondición, condición que se cumple tras realizar el caso de uso, (5) secuencia normal, sucesión de pasos para llevar a cabo un caso de uso, y (6) excepción, cambio respecto a la secuencia normal.

Además, existe un Lenguaje Unificado de Modelado (UML) (UML 2022) con el que especificar, visualizar, construir y documentar el diseño de un sistema. Existen varios tipos de diagramas UML: (1) diagramas de casos de uso para mostrar la relación entre los actores y los casos de uso dando una visión general del sistema, y (2) diagramas de secuencia para mostrar el flujo de la lógica que ocurre dentro del sistema a través de la información que los objetos se envían entre sí.

En la Figura 43 se muestra el diagrama de casos de uso de la aplicación, que está desarrollada pensando en un único jugador. El propósito de esta es permitir al jugador experimentar sobre deficiencias visuales a través de la simulación de sus efectos. Los casos de uso se encuentran descritos y extendidos en el ANEXO I: CASOS DE USO.

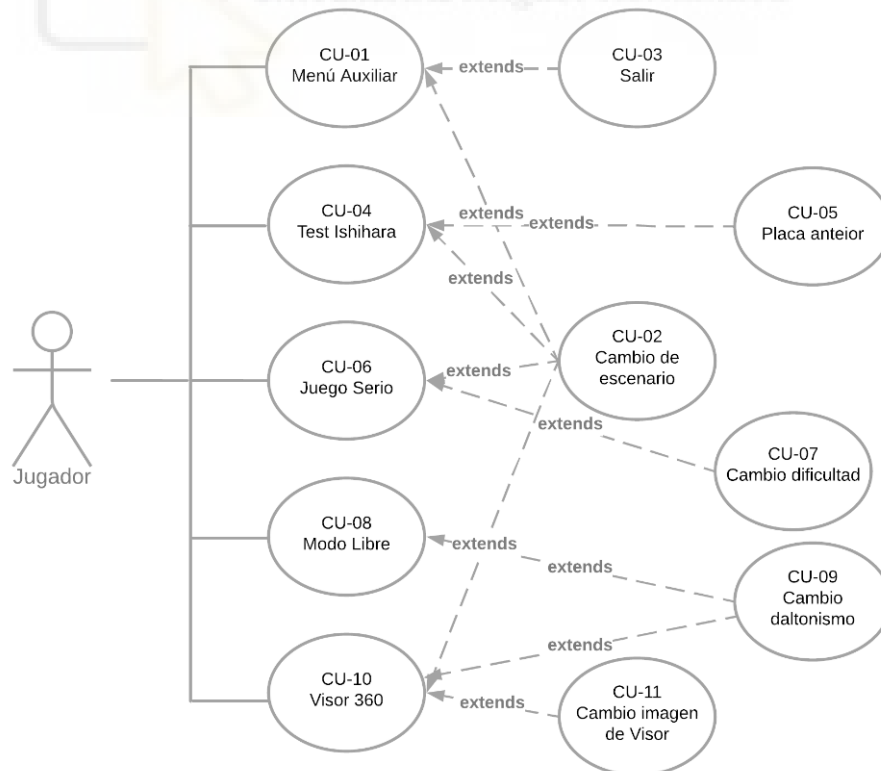


Figura 43. Diagrama de casos de uso de la aplicación (imagen propia)

3.2.3 DIAGRAMAS DE FLUJO

A continuación, se muestra un diagrama de flujo simplificado del funcionamiento general de la aplicación, así como el diagrama en detalle de procesos relevantes como: Realizando Test, Modo Libre y Juego Serio.

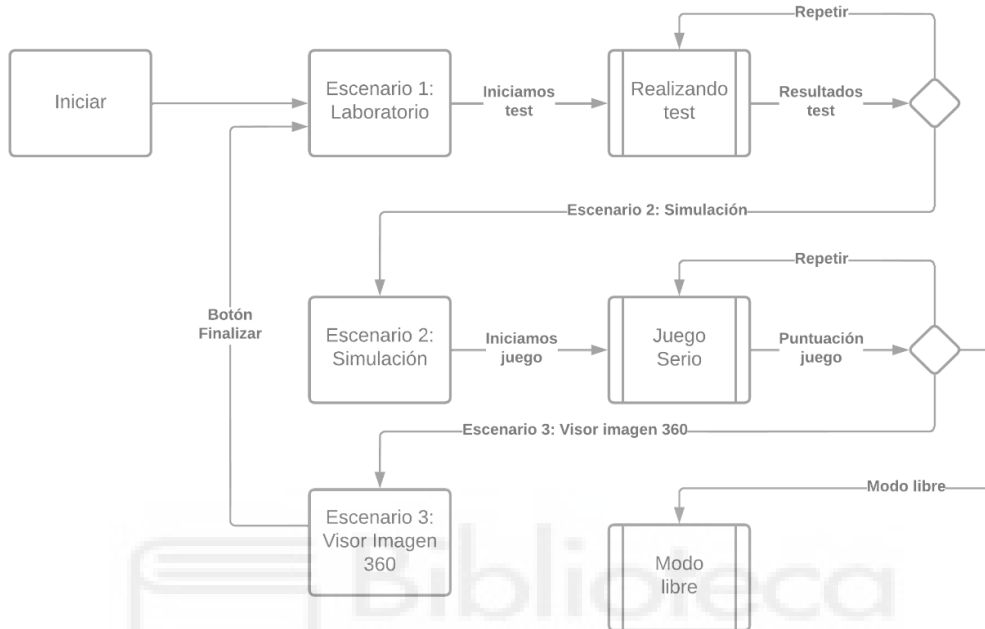


Figura 44. Diagrama de flujo simplificado de la aplicación (imagen propia)

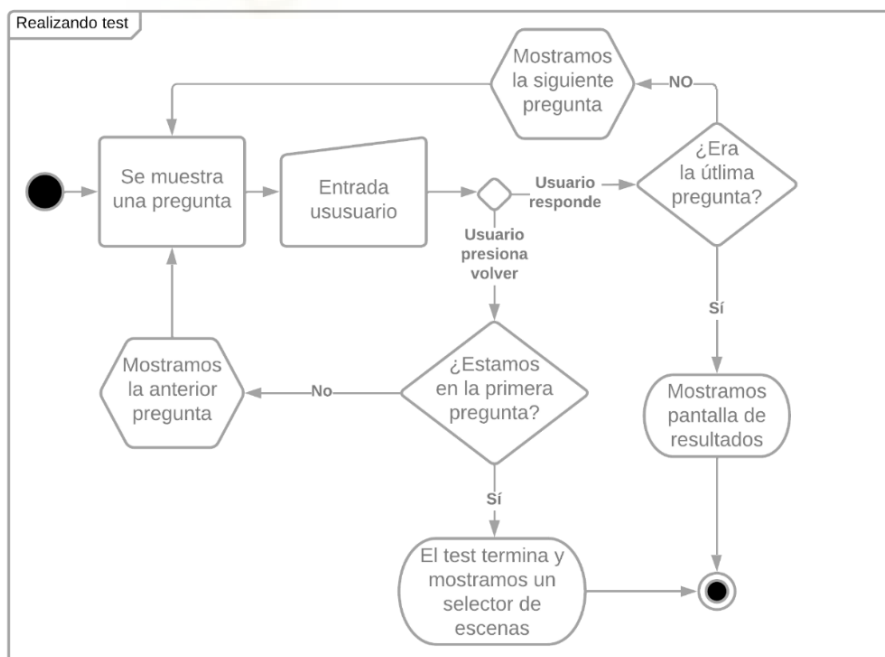


Figura 45. Diagrama de flujo detalle del proceso Realizando test (imagen propia)

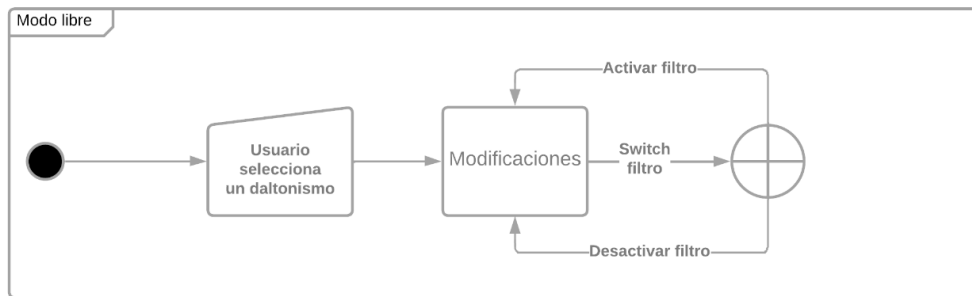


Figura 46. Diagrama de flujo detalle del proceso Modo Libre (imagen propia)

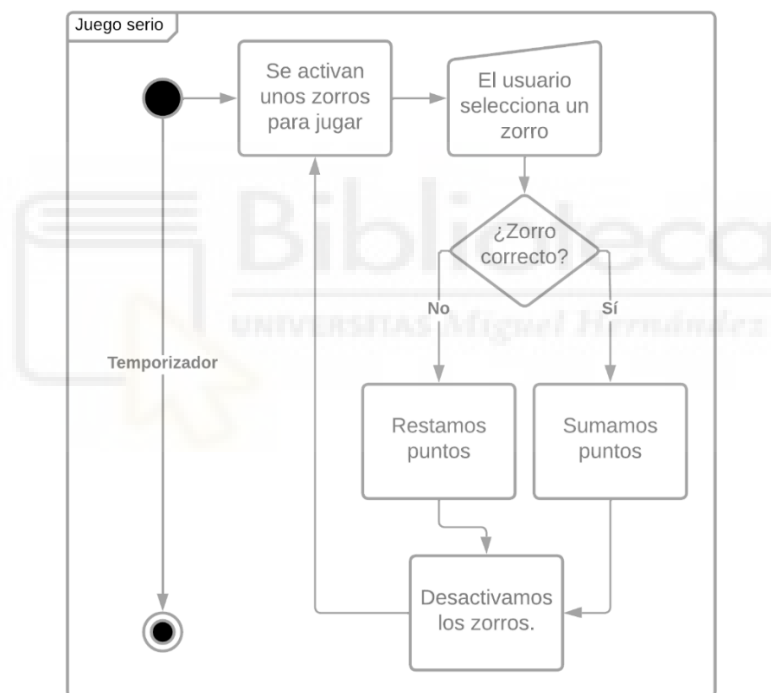


Figura 47. Diagrama de flujo detalle del proceso Juego Serio (imagen propia)

3.2.4 GESTIÓN Y PERSISTENCIA DE DATOS

Debido a la naturaleza de la aplicación no es necesario el uso de una base de datos como tal. Los únicos datos que tienen que ser guardados son las puntuaciones de los usuarios las cuales se guardan de forma nativa haciendo uso de las *PlayerPrefs* (Unity 2022e), esta clase es comúnmente utilizada para guardar configuraciones del usuario y en ocasiones pequeñas ristas de datos como las estadísticas básicas del personaje.

En la parte del test de Ishihara debemos gestionar datos para establecer si una persona es o no daltónica. A medida que el usuario realiza el test se van acumulando puntos de los distintos tipos de daltonismo, mediante las repuestas a las preguntas del test. Esta información solo estará disponible en el momento de hacer el test y se utilizará para generar una recomendación al usuario. Por ejemplo, si tienes siete puntos de protanopia es muy probable que tengas dicha deficiencia, mientras que si tienes solo tres se considera que no tienes ninguna deficiencia y esos puntos se deben a errores durante la realización de la prueba.

En la parte del juego serio permitimos al usuario compararse con los resultados de otras ejecuciones. Para ello guardamos la puntuación obtenida en la clase *PlayerPrefs* y le mostramos la puntuación más alta registrada en su instalación en el panel de resultados.

3.3 ASPECTOS DE IMPLEMENTACION RELAVANTES

3.3.1 MODELOS 3D Y FOTOGRAFÍAS 360º

Haciendo uso de Blender se han creado los tres modelos de árbol que han sido utilizados en el proyecto. Debido que se quería conseguir una estética *lowpoly* se optó por un modelo de árbol muy sencillo que se muestra en la Figura 48.

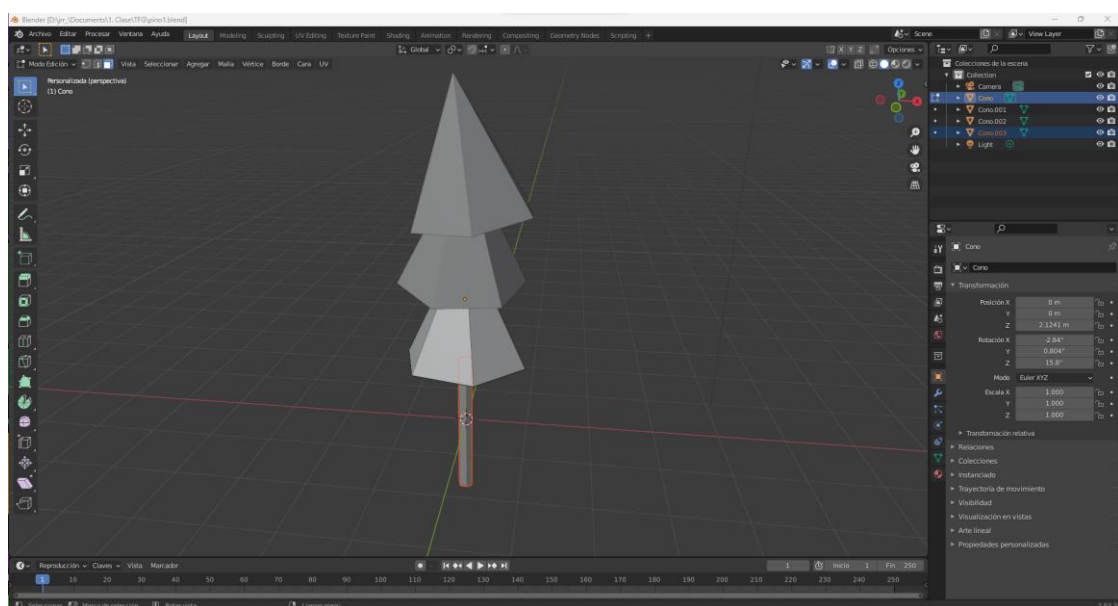


Figura 48. Modelo original de árbol creado en Blender (imagen propia)

El modelo tiene tres partes de hojas y una de tronco. Esto nos permite jugar con distintas tonalidades de verde en un mismo árbol. De este modelo original se han creado 3 modelos finales modificándolo ligeramente.

De Sketchfab se ha sacado el modelo de zorro presente en la Figura 49 a). El modelo 3D descargado de la web se ha manipulado con Blender cambiando a su postura a la que se observa en el juego y en la Figura 49 b).

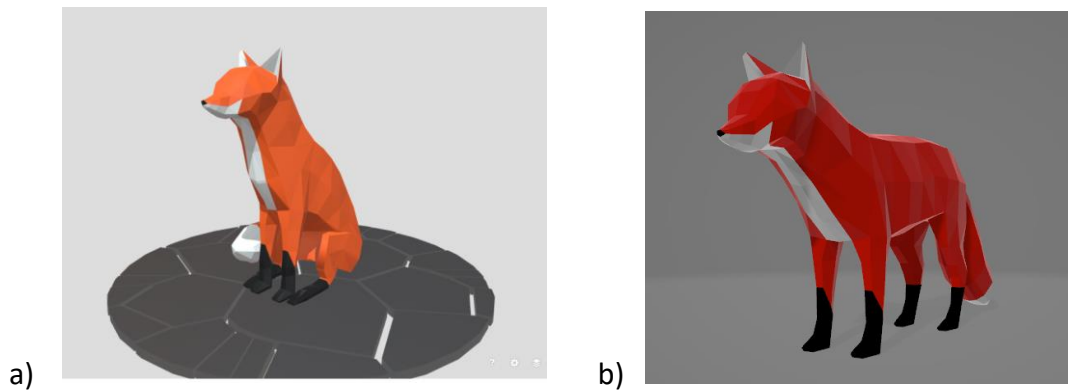


Figura 49. a) Modelo de zorro extraído de Sketchfab (imagen propia). b) Zorro manipulado y en posición base del modelo incluido en nuestro juego (imagen propia)

En cuanto a las fotografías 360º, hemos optado por incluir dos. Una imagen de alta calidad de una habitación descargada de Pixexid para simular un entorno realista y poner el usuario ante una situación cotidiana (Figura 50). Además, hemos generado una imagen con Hugin de la UMH que simule un entorno cercano al usuario (ver Figura 30).



Figura 50. Imagen 360º de una habitación, utilizada en el proyecto (imagen extraída de (Pixexid 2022))

3.3.2 ESCENARIOS QUE COMPONEN EL PROYECTO

Como se ha comentado anteriormente, se han implementado tres escenarios en el juego. Aquí describimos resumidamente los tres escenarios con el fin de mostrar los distintos objetos 3D y otros elementos que las componen.

El escenario 1, es el de inicio del proyecto. En él se dispone de tres cubos interactuables y de lo necesario para la realización de la prueba de daltonismo como se puede apreciar en la Figura 51. En ella se pueden identificar los paneles para la realización de la prueba, los cubos, un zorro que actúa de interlocutor con el usuario y un objeto flotante donde se mostrarán las placas de la prueba.

El escenario 2 es el más complejo ya que es el que implementa el minijuego. En este escenario podemos encontrar árboles y zorros para la realización del juego. Además, hay diversos paneles para el tutorial y para mostrar información sobre los distintos daltonismos, como se muestra en la Figura 52. Las texturas de los árboles se gestionan dinámicamente según el tipo de daltonismo que se esté simulando. Durante el tiempo de juego también se gestionan dinámicamente los colores de los zorros y su visibilidad.

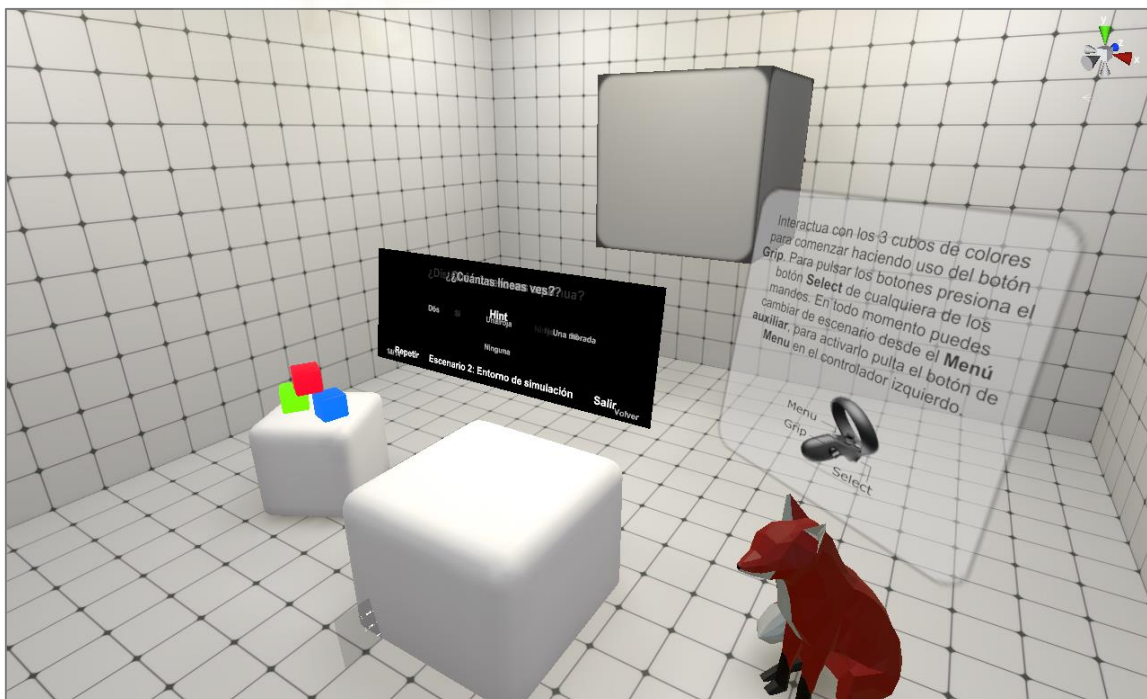


Figura 51. Captura del Laboratorio en la que se muestran sus componentes (imagen propia)



Figura 52. Imagen donde se muestran los objetos que componen el escenario de Simulación (imagen propia)

Por último, el escenario 3 se centra en la experiencia basada en fotografías 360°. Este escenario, mostrado en la Figura 53, solo dispone de los paneles informativos para formación y una esfera que es la que recrea la imagen 360°. Como se puede observar en esta figura, la esfera contiene la fotografía 360° de una de las plazas de la UMH.



Figura 53. Captura del escenario Visor 360° donde se muestran sus componentes (imagen propia)

3.3.3 DINÁMICA DEL MINIJUEGO

En este apartado vamos a describir brevemente la dinámica implementada para el minijuego de simulación del escenario 2. Hemos optado por una versión arcade que simplifique la jugabilidad y limite a un tiempo concreto la partida.

Cuando el jugador inicia el juego empieza a sonar una música de fondo que dota al juego de mayor dinamismo. Además de esto, el controlador del juego escoge un número de zorros a mostrar dependiente de la dificultad de juego seleccionada. Entre los zorros mostrados al usuario, se escoge de forma aleatoria uno de ellos (el zorro *variocolor*) y se le cambia su color original.

El color configurado para el zorro *variocolor* se ha diseñado para cada tipo de daltonismo de manera que, mientras un usuario sin daltonismo lo reconocería fácilmente, en el entorno simulando (en el que el jugador verá el bosque como si padeciese daltonismo) las diferencias entre los zorros normales y el *variocolor* serán mínimas.

En la Figura 54 mostramos un ejemplo de esta situación. Por ejemplo, en la Figura 54 a) se puede observar el minijuego con visión normal donde se diferencian perfectamente los dos tipos de zorros. En la Figura 54 b) se ven las dificultades para apreciar las diferencias en un entorno simulado con protanopia.

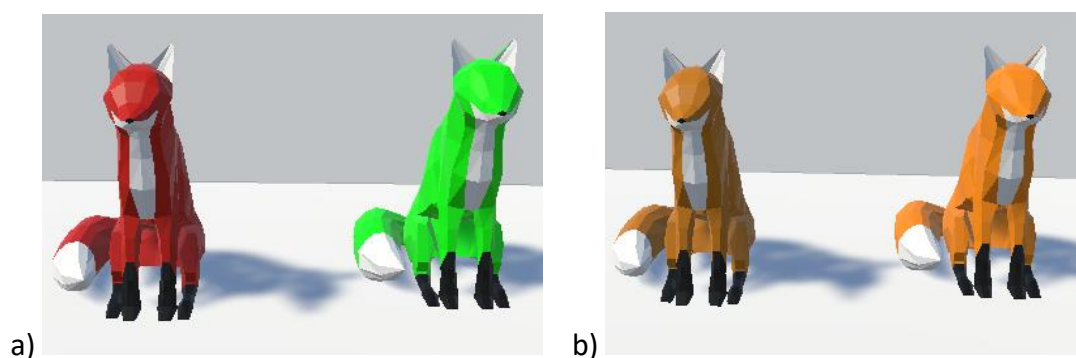


Figura 54. Capturas comparativas donde se aprecian las diferencias entre el zorro normal y el variocolor siendo a) en un entorno sin defecto y b) en un entorno con protanopia (imágenes propias)

Gamificar la búsqueda del zorro *variocolor* es la dinámica que proponemos para que el jugador empatice con el problema. En la Figura 55 se muestra el código usado para dinamizar la visualización de los zorros y el cambio de texturas. El jugador dispondrá de 10 segundos en cada entono simulado (en total cinco, uno normal y el resto para los distintos tipos de daltonismo). El primer entorno simulado será con visión normal. Una vez finaliza el tiempo de 10 segundos, se manipularán los materiales del escenario para simular el siguiente tipo de daltonismo; de esto se encarga la función asíncrona *Timer* (Unity 2022d) (ver Figura 56).

```

1 private static void FoxStart2 ()
2 {
3     int rand;
4     // Desactivamos los zorros
5     foreach (var a in listFoxes)
6     {
7         a.SetActive(false);
8         a.GetComponent<FoxInteractor>().SetTarget(false);
9     }
10    // Seleccionamos los zorros que van a ser activados de inicio
11    List<int> activatedFoxes = new List<int>();
12    for (int i = 0; i < (int)staticDifficulty; i++)
13    {
14        int counter = 0;
15        do
16        {
17            rand = UnityEngine.Random.Range(0, listFoxes.Count);
18            if (counter > 100)
19            {
20                Debug.Log("Counter" + listFoxes.Count);
21                break;
22            }
23            else
24                counter++;
25        }
26        while (activatedFoxes.Contains(rand));
27        activatedFoxes.Add(rand);
28    }
29
30    // Activamos los zorros seleccionados pero les quitamos el fox interactor
31    foreach (int i in activatedFoxes)
32    {
33        listFoxes[i].SetActive(true);
34        listFoxes[i].GetComponent<Renderer>().materials[0].color = ColorCreator(listFoxMaterials[0][0].color, contador);
35        //Destroy(listFoxes[i].GetComponent<XRSimpleInteractable>());
36    }
37
38    // Seleccionamos el zorro especial
39    rand = UnityEngine.Random.Range(0, (int)staticDifficulty);
40    listFoxes[activatedFoxes[rand]].AddComponent<XRSimpleInteractable>();
41    listFoxes[activatedFoxes[rand]].GetComponent<Renderer>().materials[0].color = ColorCreator(shiny[contador], contador);
42    listFoxes[activatedFoxes[rand]].GetComponent<FoxInteractor>().SetTarget(true);
43 }

```

Figura 55. Código de la función encargada de escoger a los zorros (imagen propia)

Durante los 10 segundos en cada entorno simulado, el jugador deberá encontrar y seleccionar al zorro *variocolor* tantas veces como le sea posible. Si lo hace tendrá un input sonoro y háptico, además de esto sumará una cantidad de puntos en función de la dificultad jugada. Cada vez el zorro *variocolor* sea encontrado, se ocultarán todos los

zorros y aparecerán nuevos en nuevas posiciones para seguir jugando. Si el jugador selecciona un zorro erróneo recibirá una señal acústica distinta y se le restarán puntos.

```
1 private async void Timer (int time)
2 {
3     string[] strings = { "Normal", "Protan", "Deuteran", "Tritan", "Acroma" };
4     for (int i = time; i >= 0; i--)
5     {
6         for(int j = 99; j >= 0; j--) {
7             //Debug.Log
8             timerDisplay.text = i + ":" + j + "\n" + strings[contador];
9             await Task.Delay(TimeSpan.FromSeconds(0.01));
10        }
11        Debug.Log("Tiempo restante: " + i);
12        //await Task.Delay(TimeSpan.FromSeconds(1));
13    }
14    if (contador < 4)
15    {
16        contador++;
17        StartGame2();
18    }
19    else
20        NewEndGame();
21 }
```

Figura 56. Captura de la función *Timer*, esta se encarga del temporizador del juego (imagen propia)

Cuando se han simulado todos los daltonismos la música termina, el escenario vuelve a tener los materiales de una visión normal, los zorros son desactivados y se le muestra al jugador un panel con sus resultados, la mejor puntuación registrada.

```
1 private void MaterialChanger (BlindnessType blindness)
2 {
3     int rand;
4
5     tutoFox.GetComponent<Renderer>().materials[0].color = ColorCreator(listFoxMaterials[0][0].color, (int)blindness);
6     tutoFox.GetComponent<Renderer>().materials[1].color = ColorCreator(listFoxMaterials[1][0].color, (int)blindness);
7     tutoFox.GetComponent<Renderer>().materials[2].color = ColorCreator(listFoxMaterials[2][0].color, (int)blindness);
8
9     foreach (var fox in listFoxes)
10    {
11        rand = UnityEngine.Random.Range(0, listFoxMaterials[0].Count);
12
13        fox.GetComponent<Renderer>().materials = new Material[] { listFoxMaterials[0][rand], listFoxMaterials[1][rand], listFoxMaterials[2][rand] };
14        fox.GetComponent<Renderer>().materials[0].color = ColorCreator(listFoxMaterials[0][rand].color, (int)blindness);
15        fox.GetComponent<Renderer>().materials[1].color = ColorCreator(listFoxMaterials[1][rand].color, (int)blindness);
16        fox.GetComponent<Renderer>().materials[2].color = ColorCreator(listFoxMaterials[2][rand].color, (int)blindness);
17    }
18
19    foreach (var tree in listTrees)
20    {
21        rand = UnityEngine.Random.Range(0, listTreeMaterials[0].Count);
22
23        tree.transform.GetChild(0).GetComponent<Renderer>().material = listTreeMaterials[0][rand];
24        tree.transform.GetChild(0).GetComponent<Renderer>().material.color = ColorCreator(listTreeMaterials[0][rand].color, (int)blindness);
25
26        tree.transform.GetChild(1).GetComponent<Renderer>().material = listTreeMaterials[1][rand];
27        tree.transform.GetChild(1).GetComponent<Renderer>().material.color = ColorCreator(listTreeMaterials[1][rand].color, (int)blindness);
28
29        tree.transform.GetChild(2).GetComponent<Renderer>().material = listTreeMaterials[2][rand];
30        tree.transform.GetChild(2).GetComponent<Renderer>().material.color = ColorCreator(listTreeMaterials[2][rand].color, (int)blindness);
31
32        tree.transform.GetChild(3).GetComponent<Renderer>().material = listTreeMaterials[3][rand];
33        tree.transform.GetChild(3).GetComponent<Renderer>().material.color = ColorCreator(listTreeMaterials[3][rand].color, (int)blindness);
34    }
35
36    terrain.materialTemplate = listTerrainMaterials[(int)blindness];
37
38    skybox.SetColor("_SkyTint", ColorCreator(skybox.GetColor("_SkyTint"), (int)blindness));
39
40    HandColorchanger(blindness);
41 }
42 }
```

Figura 57. Código de la aplicación, concretamente la función *MaterialChanger*, encargada de cambiar todos los materiales del escenario (imagen propia)

3.3.4 EDICIÓN DE TEXTURAS Y MANUPULACIÓN DE LA CÁMARA

En el escenario 2, se ha simulado el daltonismo mediante la modificación de los materiales. Para ello, se ha desarrollado el código mostrado en la Figura 58 que unifica el cambio de todos los materiales del escenario en una sola función. Para realizar esta tarea se requiere el uso de otra función que, dado un color y un daltonismo, devuelva el color nuevo. Esta función se muestra en la Figura 58.

```
1 /// <summary>
2 ///     Usa los ratios preestablecidos para los distintos tipos de daltonismo y
3 ///     usa los elementos r, g y b del color dado y les aplica los ratios para
4 ///     crear un nuevo color.
5 /// </summary>
6 /// <param name="rgb">Color que se debe filtrar para simular el defecto.</param>
7 /// <param name="d">Tipo de daltonismo en formato BlindnessType casteado a Int.</param>
8 /// <returns>Color modificado a partir del color dado y el defecto deseado.</returns>
9 private static Color ColorCreator (Color rgb, int d)
10 {
11     return new Color(
12         rgb.r * ratios[d][0] + rgb.g * ratios[d][1] + rgb.b * ratios[d][2],
13         rgb.r * ratios[d][3] + rgb.g * ratios[d][4] + rgb.b * ratios[d][5],
14         rgb.r * ratios[d][6] + rgb.g * ratios[d][7] + rgb.b * ratios[d][8]
15     );
16 }
```

Figura 58. Código de la función *ColorCreator*, función necesaria para el correcto funcionamiento de *MaterialChanger* (imagen propia)

En el escenario 3, dado que las imágenes 360º carecen de objetos independientes a los que modificar sus texturas, se ha hecho uso de la manipulación directa de la cámara con el script de Allan Zucconi mencionado en el capítulo 2. Para esto debemos incluir el script a la cámara, como se muestra en la Figura 59 y modificar el valor de una variable ya sea mediante un *dropdown* en el editor o mediante código.

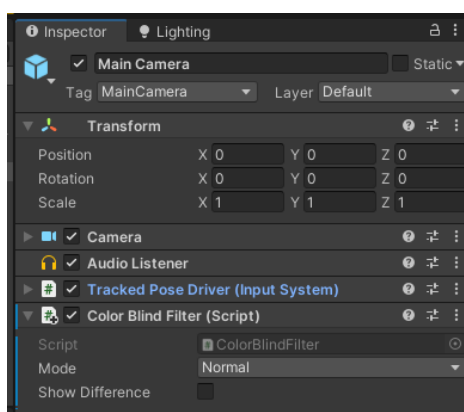


Figura 59. Cámara con el script de simulación de daltonismo (imagen propia)

3.3.5 PANELES INTERACTIVOS Y MENÚ SECUNDARIO

A medida que se usa la aplicación el usuario utiliza distintos menús. Para interactuar con todos ellos se propone el uso de uno de los botones del controlador: el gatillo. Por norma general la lógica detrás de esto consiste en activación y desactivación de paneles. Un panel que tiene un poco más de complejidad es el que se encarga de modificar el daltonismo simulado en entorno de Simulación, el código se encuentra en la **Figura 60**. En este caso, todos los botones llaman a la misma función y le pasan un entero como parámetro con el que identificar el daltonismo seleccionado, una vez sabemos esto modificamos los textos de los paneles activos y activamos la función encargada de cambiar los materiales del escenario.

```
1 public void BlindnessChanger(int type)
2 {
3     switchButton.gameObject.SetActive(true);
4     MaterialChanger((BlindnessType)type);
5     switch (type)
6     {
7         case 1:
8             titleText.text = "PROTANOPIA";
9             uiText.text = "Se caracteriza por deficiencia al captar colores rojos. Junto con la deficiencia al verde son las más comunes.";
10            blindnessSelector = BlindnessType.Protanopia;
11            break;
12            case 2:
13                titleText.text = "DEUTORANOPIA";
14                uiText.text = "Se caracteriza por la deficiencia hasta el color verde. Junto con la deficiencia al rojo con las más comunes.";
15                blindnessSelector = BlindnessType.Deuteranomia;
16                break;
17            case 3:
18                titleText.text = "TRITANOPIA";
19                uiText.text = "Se caracteriza por la falta de percepción a los azules. Esta tiende a afectar más en el día a día a los que la sufren.";
20                blindnessSelector = BlindnessType.Tritanopia;
21                break;
22            case 4:
23                titleText.text = "ACROMATOPSIA";
24                uiText.text = "Los que la sufren pierden por completo la capacidad de captar los colores.";
25                blindnessSelector = BlindnessType.Acromatopsia;
26                break;
27            }
28            FreeModeFoxChanger(blindnessSelector);
29            isActiveFilter = true;
30 }
```

Figura 60. Código de la aplicación llamado al presionar los botones en el modo libre del escenario de Simulación (imagen propia)

Algunos de los paneles con los que interactúa el usuario se encuentran en el menú secundario situado en el brazo izquierdo. Esta localización no ayuda de cara a la usabilidad, pero resulta muy llamativo para usuarios noveles. Para el correcto posicionamiento de estos paneles se han posicionado dentro del objeto que necesita el sistema de Realidad Virtual para detectar el controlador izquierdo. Una vez insertado se ha modificado su posición hasta que nos hemos encontrado satisfechos con su posicionamiento, esto se muestra en la Figura 61.

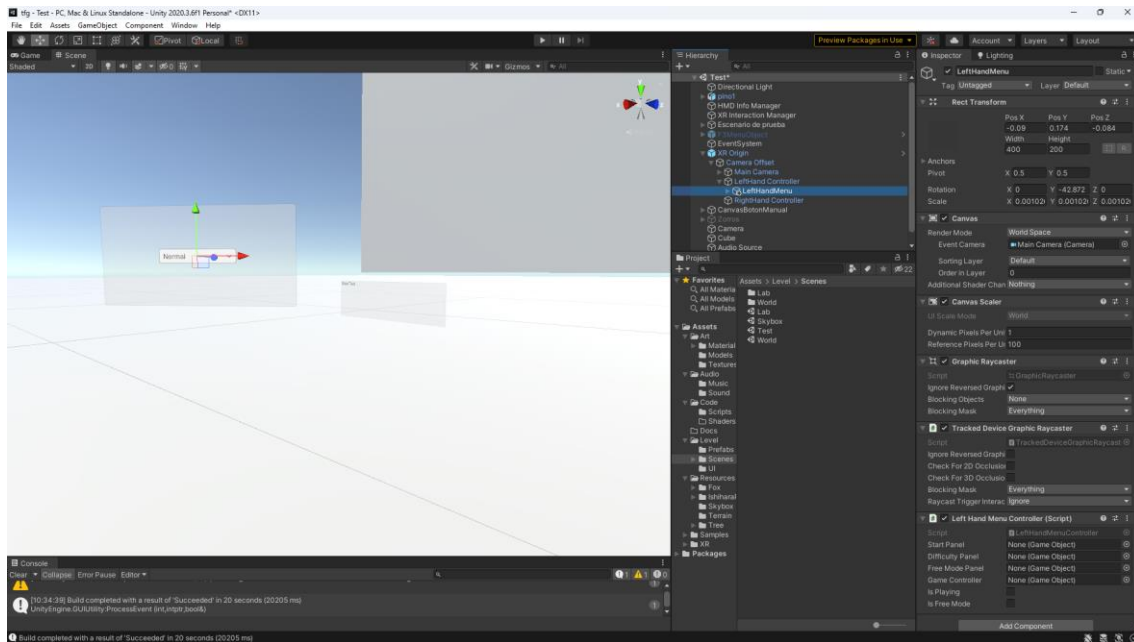


Figura 61. Captura del editor en el escenario de pruebas en el que se muestra el menú secundario dentro del controlador de la mano izquierda (imagen propia)

3.3.6 GENERACION DE EJECUTABLE Y DESPLIEGUE

Gracias a Unity, la generación del ejecutable para PC es sencilla. Primero debemos configurar los datos de la aplicación, para hacer eso debemos dirigirnos a *Edit > Player Settings*, una vez en este menú iremos a la pestaña de Player podremos configurar múltiples opciones. Tras ello, se iniciará el proceso automático de creación del ejecutable pinchando en *File > Build Settings* y presionando *Build*. Una vez comenzado el proceso pedirá una carpeta de destino para la *build*. Este es un proceso lento, aunque si hay pocos cambios respecto a la versión anterior se hace en pocos segundos, al menos en un proyecto de dimensiones reducidas como este. Gracias a la creación de este ejecutable el software ha sido desplegado y probado en hardware con el que no había sido testeado inicialmente.

3.4 PRUEBAS PRELIMINARES CON USUARIOS

Durante el desarrollo del proyecto, la aplicación se ha probado con dos usuarios de que han contribuido al proceso de mejora con propuestas incluidas como requisitos que se han incluido en los casos de uso y forman parte del prototipo aquí desarrollado.

Una vez finalizado el proyecto, se han realizado pruebas con tres usuarios que han podido probar el prototipo funcional completamente. La prueba ha consistido en lo siguiente:

1. Se equipa al usuario con las gafas de RV y los dos controles.
2. Se le describe el objetivo de la aplicación que va a probar y se le indica que solo debe requerir de nuestra ayuda si se encuentra perdido.
3. Se inicia la aplicación en el escenario 1 y se deja al usuario totalmente libre para realizar la experiencia completa. Si el usuario lo requiere le daremos la ayuda que considere necesaria, en otro caso esperaremos a que termine la experiencia.

Esta prueba preliminar nos ha permitido detectar puntos conflictivos respecto al diseño o implementación de la aplicación que se recogen como trabajo futuro y se enumeran a continuación:

- Detectamos comportamientos diferentes en función de la experiencia previa de los usuarios con soluciones de RV. Los usuarios que prueban por primera vez este tipo de tecnología requieren nuestra ayuda en varias partes de la prueba. Sería recomendable incluir una experiencia de RV previa a la prueba de este juego.
- Algunos usuarios han tenido problemas con la manipulación de los controladores de Oculus debido a su falta de experiencia. A pesar de la explicación ofrecida por el asistente dentro del programa, el usuario necesita más formación previa para afianzar los botones y relacionarlos con la infografía utilizada actualmente.
- En cuanto a la navegabilidad entre los tres escenarios, hemos detectado que el usuario, en ocasiones, parece no tener claro cuál es el siguiente paso que debe realizar. Sería conveniente incluir un botón de ayuda que esté disponible en cualquier parte del juego proporcionando indicaciones o incluir una mayor narrativa que ayude a enlazar mejor los distintos escenarios.

- Mientas que la narrativa del escenario 1 (Test de Ishihara) y el escenario 3 (fotografías 360º) ha sido efectiva y los usuarios la han podido realizar sin ayuda, en el escenario 2 (minijuego) los usuarios han experimentado algunos problemas. Se ha detectado una falta de comprensión de las tareas a realizar. La explicación se hace mediante unos párrafos que, en el contexto de este juego, creemos que tiene demasiada información. Sería mejor incluir animaciones para explicarlo de forma amena e interactiva.
- Además, en el escenario 2 (minijuego) el usuario no es consciente que, aunque no tenga ningún tipo de daltonismo se le están simulando una situación en la que lo está viendo todo como si lo tuviese. Algunos usuarios creen que las dificultades para superar el juego son fruto de sufrir algún tipo de problema, y no se dan cuenta hasta que finaliza el juego y pasan al modo libre. Además, el usuario de forma natural no accede al modo libre del escenario 2. Podría ser interesante incluir el modo libre de este escenario a modo de tutorial y obligar al usuario a hacerlo antes de comenzar el minijuego.
- El usuario de forma natural no accede al menú auxiliar. Podría ser recomendable incluir este menú como parte de la ayuda comentada en el punto anterior, que debería estar accesible en cualquier momento de la experiencia.

Merece la pena destacar que una de estas tres pruebas se ha podido realizar en un dispositivo distinto a las Rift S prestadas por la EPSE, concretamente se han utilizado unas Meta Quest 2. En esta prueba, por lo general la aplicación ha funcionado correctamente, pero se han detectado algunos errores que resumimos a continuación. La simulación de daltonismos a través de la modificación de la cámara no funciona, podría deberse a un conflicto con el nuevo hardware al ejecutar el juego directamente desde la *build*. Además de esto el usuario ha buscado intencionalmente errores en el funcionamiento del juego. En general la experiencia ha sido satisfactoria, aunque se han detectado algunos pequeños problemas técnicos relacionados con las físicas de los objetos interactivo del escenario 1.

3.5 CÓDIGO FUENTE

El código fuente de esta aplicación se encuentra disponible en un repositorio privado de Bitbucket al que se puede acceder a través del siguiente enlace: (Rodríguez Ruiz 2022b). Si se desea contribuir al mismo se deberá solicitar acceso. En la Figura 62 se muestra una captura de la carpeta principal del repositorio donde se pueden apreciar algunas de las carpetas que componen el proyecto Unity.

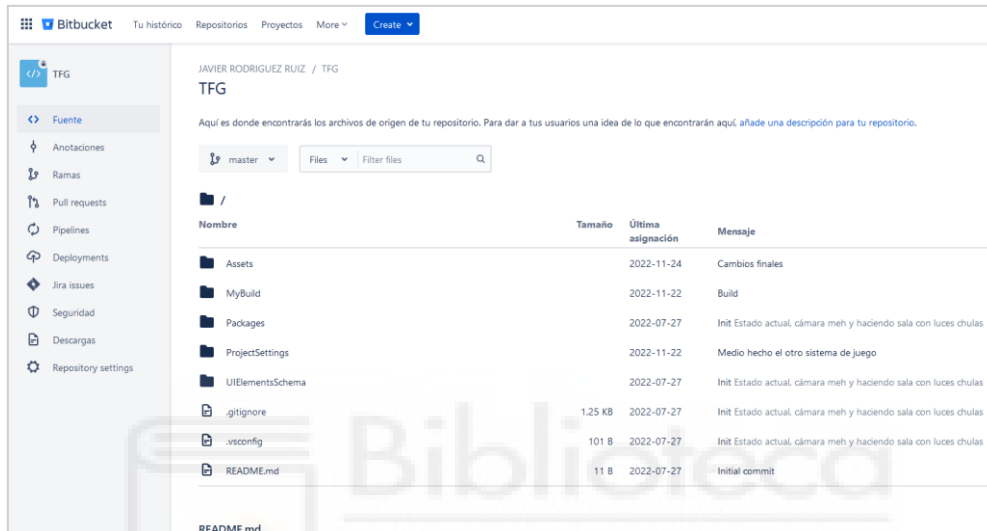


Figura 62. Imagen donde se muestra el repositorio donde está alojado el proyecto en Bitbutcket (imagen propia)

3.6 EJEMPLO DE USO DE LA APLICACIÓN

El tratarse de una aplicación de RV el usuario deberá disponer de unas gafas de RV y ejecutar la aplicación en ellas. En la Figura 63 se muestra el entorno de pruebas.



Figura 63. Fotografía del entorno en el que se ha desarrollado parte de la aplicación y la mayoría de las pruebas con usuarios (imagen propia)

3.6.1.1 ESCENARIO 1: LABORATORIO

El usuario al abrir la aplicación se encuentra en el Laboratorio, lugar donde se realiza la prueba de daltonismo.

Para guiar al usuario en su experiencia se ha introducido un zorro que mostrará mensajes al usuario a modo de tutorial, sirviendo por tanto de guía durante toda la experiencia (ver Figura 64).



Figura 64. Zorro tutorial en Laboratorio (imagen propia)

Para comenzar la prueba se debe interactuar con unos cubos, mostrados en la Figura 65, que sirven tanto para decorar como para que el usuario se familiarice con los controles de RV.

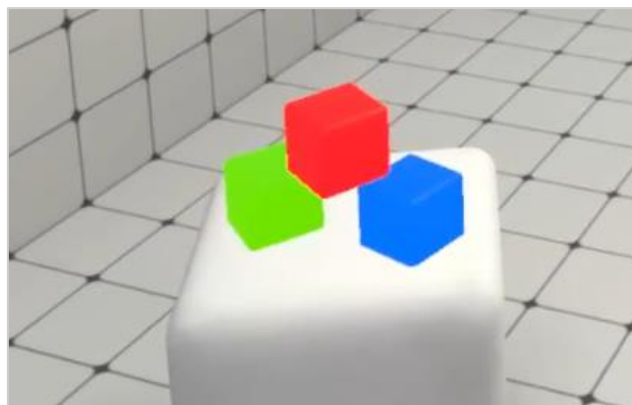


Figura 65. Cubos interactivables del Laboratorio (imagen propia)

Cuando el usuario haya interactuado con los tres cubos, empezará la prueba de Daltonismo. En ella, se irán intercalando las distintas placas de Ishihara con preguntas para determinar los problemas del usuario (ver Figura 66).

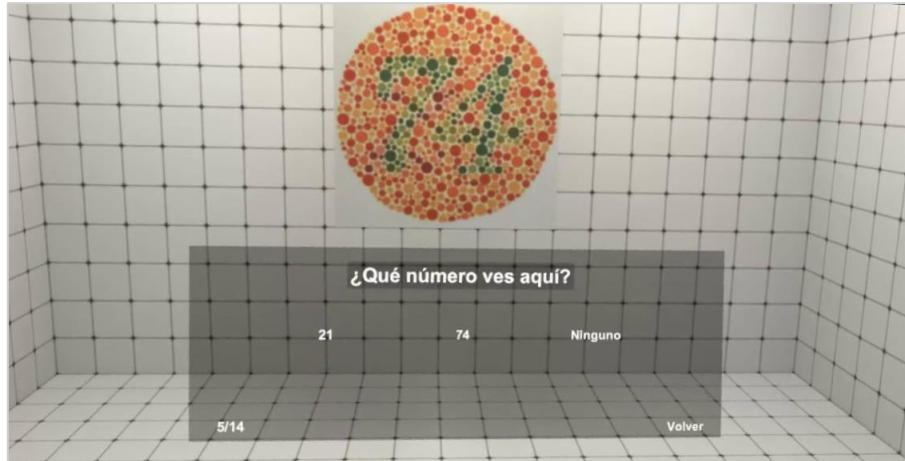
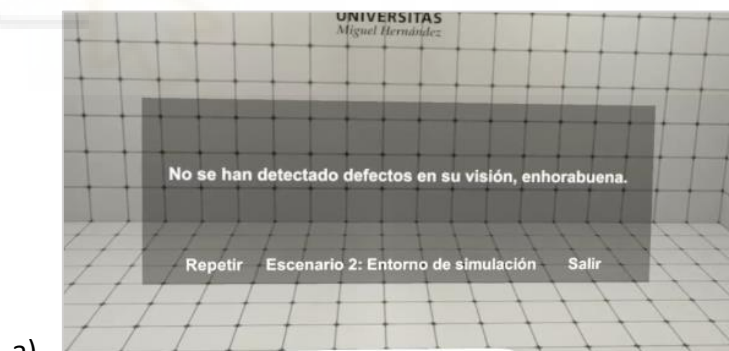
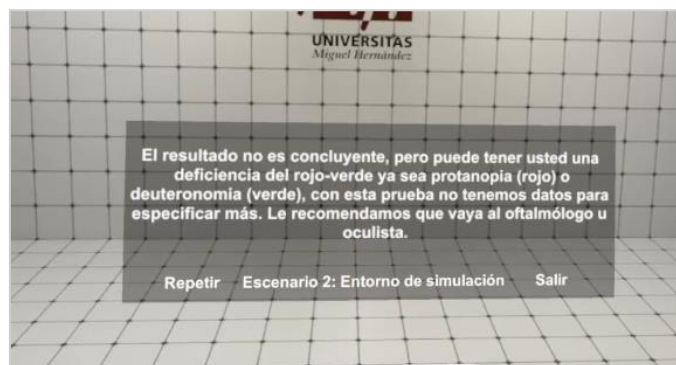


Figura 66. Test Ishihara (imagen propia)

Una vez terminado el test aparecerá uno de los paneles de la Figura 67 a modo de pantalla de resultados. En este panel se ofrecerá al usuario la posibilidad de repetir el test, salir o continuar con la siguiente escena.



a)



b)

Figura 67. Panel resultados a) no se han detectado problemas, b) se ha detectado anomalía rojo-verde (imagen propia)

3.6.1.2 ESCENARIO 2: BOSQUE

En este escenario tiene lugar el minijuego de búsqueda de zorros. Como podemos apreciar en la Figura 68, se mantiene a modo de tutorial el mismo zorro que teníamos en el laboratorio, quien nos espera ahora para explicar la dinámica del juego.

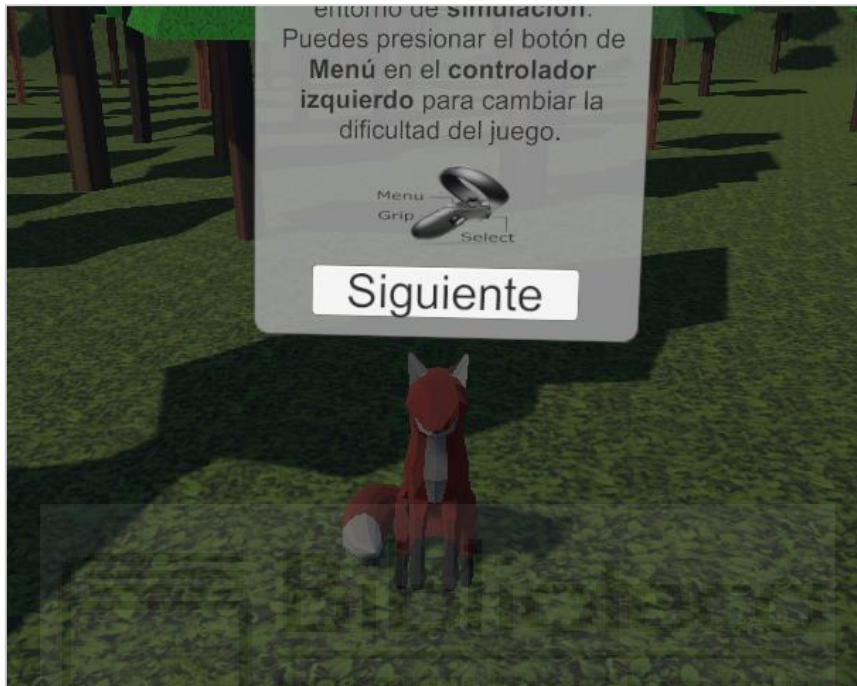


Figura 68. Zorro tutorial en Simulador (imagen propia)

Además, de forma previa al inicio del juego podremos acceder al menú secundario usando el botón de menú del controlador de la mano izquierda. Mostrará una interfaz que permite seleccionar la dificultad del juego (ver Figura 69).



Figura 69. Menú selección de dificultad (imagen propia)

Una vez finalizada la explicación por parte de la mascota dará comienzo el minijuego. En la Figura 70 se aprecia una captura del minijuego en modo normal. Como se puede observar, para una persona sin daltonismo, resulta relativamente sencillo identificar el zorro *variocolor*.



Figura 70. Capturas del minijuego con visión normal (imagen propia)

Durante el juego, y como se observa en la Figura 71, en la parte superior de la pantalla el jugador verá un contador que le indica el tiempo restante de simulación en la que se encuentra. Por defecto, tendrá 10 segundos para encontrar tantos zorros *variocolor* como le sea posible. Al acertar el jugador conseguirán puntos y al fallar se restarán.

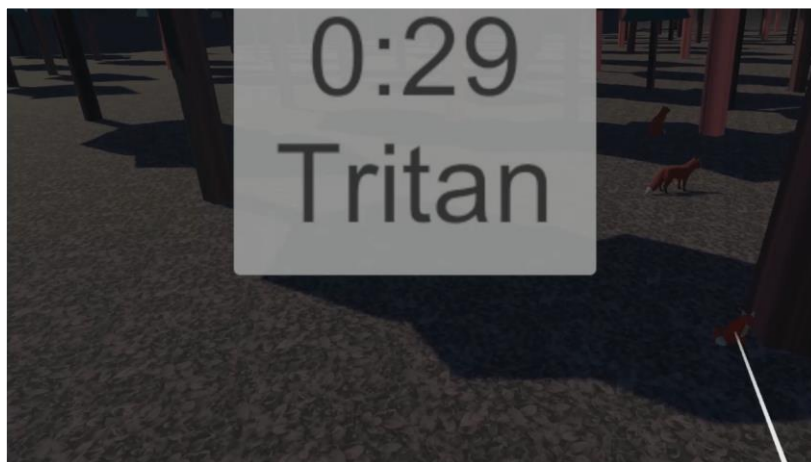


Figura 71. Captura que muestra el indicador del tiempo restante y del defecto actual (imagen propia)

En la Figura 72 se muestran capturas de los cuatro entornos simulando distintos tipos de daltonismo. En estos casos identificar al zorro *variocolor* es más complicado que el modo normal, lo que permite al jugador empatizar con el problema.

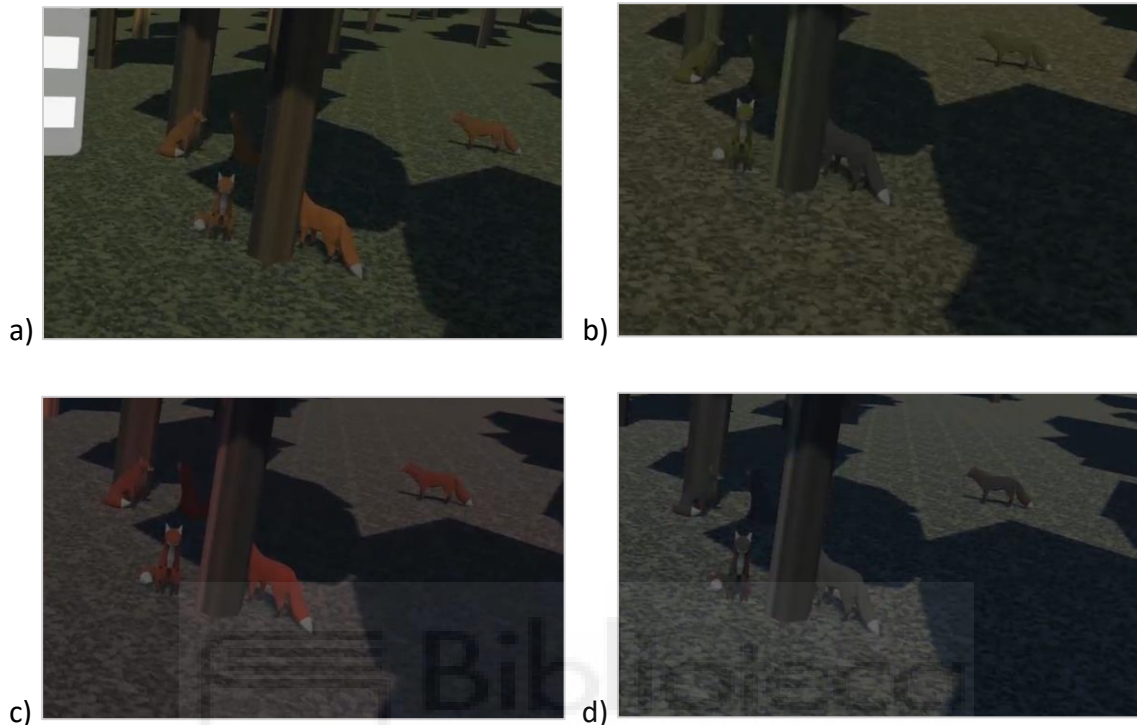


Figura 72. Capturas de escenario de Simulación dentro del modo libre para simular a) protanopia, b) deuteranopia, c) tritanopia y d) acromatopsia (imagen propia)

Una vez terminado el juego la interfaz te invita a volver a jugar cambiando la dificultad, pasar al modo libre o continuar con el escenario 3 (ver Figura 73). Aquí se mostrará una pantalla con la puntuación obtenida por el usuario, y la puntuación máxima obtenida por otros jugadores con esa misma dificultad.

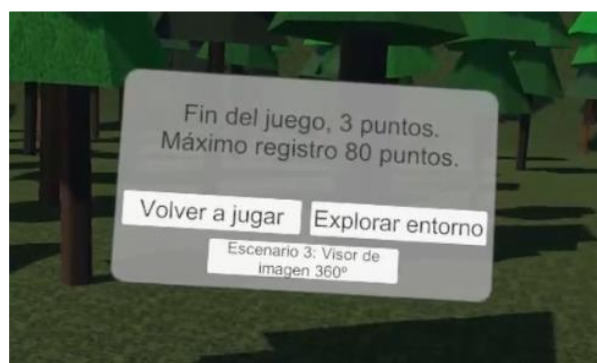


Figura 73. Captura con el panel de resultados del minijuego (imagen propia)

En el modo libre, se le ofrece al usuario la posibilidad de explorar los distintos entornos simulados. Esta vez sin límite de tiempo y con la opción de comparar con la visión normal

3.6.1.3 FOTOGRAFÍAS 360

Por último, el jugador pasará al escenario 3 que, como su propio nombre indica y se puede apreciar en la Figura 74, es un visor de RV de imágenes 360°. La interactividad del usuario se limita a cambiar el daltonismo simulado (ver resultado en Figura 75).



Figura 74. Captura del escenario Visor 360° (imagen propia)

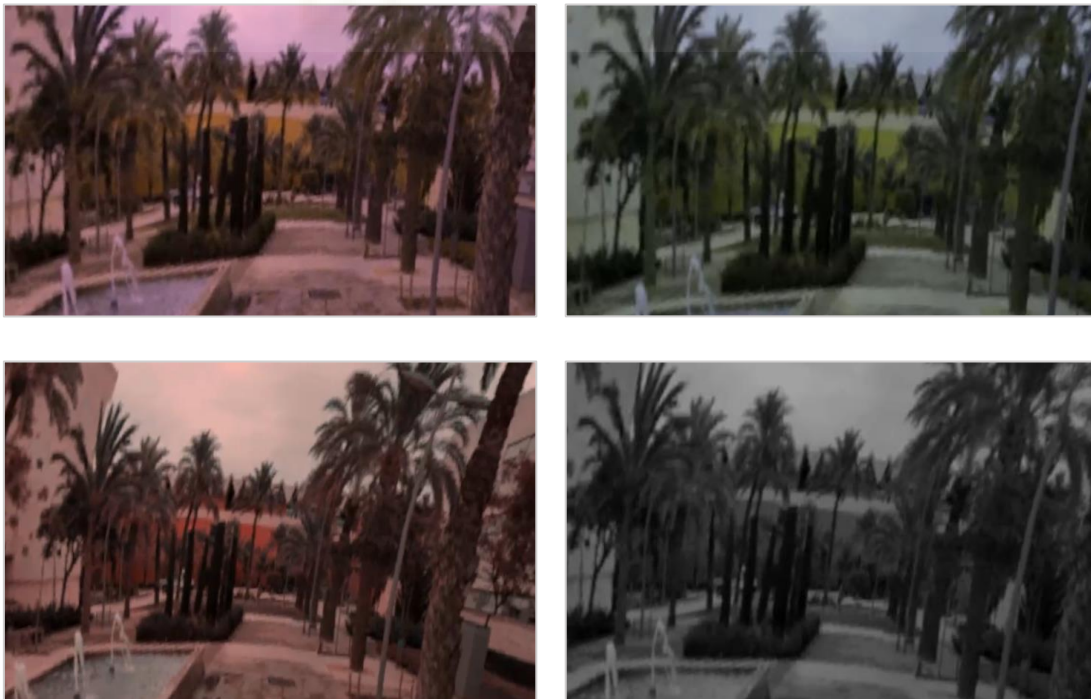


Figura 75. Capturas del Visor de imágenes con la primera imagen simulando a) protanopia, b) deuteranomia, c) tritanopia y d) acromatopsia (imágenes propias)

Usando el menú secundario se permite al usuario pasar a la siguiente imagen como se muestra en la Figura 76.



Figura 76. Menú secundario en escenario Visor 360° (imagen propia)

Como se ha comentado previamente, la aplicación dispone de un segundo entorno 360° de alta calidad como se puede observar en Figura 77, Figura 78, Figura 79 y Figura 80. También se puede observar cómo en este entorno, y con un fin educativo, se le describe al usuario en que consiste cada tipo de daltonismo. Podrá experimentar alternando entre visión normal y visión simulada para comprobar.



Figura 77. Captura del visor 360° con la segunda imagen simulando protanopia (imagen propia)



Figura 78. Captura del visor 360º con la segunda imagen simulando deuteranopia (imagen propia)



Figura 79. Capturas del visor 360º con la segunda imagen simulando tritanopia (imagen propia)



Figura 80. Captura del visor 360º con la segunda imagen simulando acromatopsia (imagen propia)

3.7 VIDEO CON LA DEMO DE LA APLICACIÓN

Se ha preparado un video resumen de donde se muestra un ejemplo de uso de la aplicación y que se encuentra disponible en el enlace: (Rodríguez Ruiz 2022a).



Figura 81. Vídeo con la demo del juego serio desarrollado

4 CONCLUSIONES Y TRABAJO FUTURO

En este trabajo se ha realizado la implementación de un Juego Serio que simule defectos visuales relacionados con el daltonismo en un entorno de Realidad Virtual, para ello se ha usado la metodología GTD que nos ha permitido alcanzar el objetivo principal de este proyecto generando un prototipo funcional desplegado en un dispositivo Oculus Rift S con el que se han podido llevar a cabo algunas pruebas preliminares con usuarios.

El análisis del estado del arte nos permitió definir un marco de trabajo apropiado. Desde el punto de materiales y aplicaciones desarrolladas, el análisis de trabajos relacionados permitió identificar una carencia en cuanto a juegos serios centrados en la simulación del daltonismo en el que el foco principal sea la gamificación frente a la simulación; creemos que el juego aquí desarrollado permite contribuir a este ámbito. Además, el estudio de los fundamentos de color relacionados con los diferentes tipos de daltonismo nos ha permitido aprender y materializarlos en un entorno de RV de forma apropiada. A modo de resumen, hemos identificado materiales útiles como el test de Ishihara o las dos formas de simular estos defectos que se han podido integrar con éxito en nuestra herramienta. Desde el punto de vista más técnico relacionados con el desarrollo de videojuegos, la elección de motores de desarrollo de videojuegos se limitó a elegir entre los dos grandes entornos disponibles en el mercado (Unity o Unreal) y el uso del SDK de Oculus compatible con el dispositivo disponible para probar nuestra solución. En este sentido, su elección también ha sido adecuada ya que nos ha permitido desarrollar el juego serio, siendo finalmente desplegado en dos dispositivos Oculus: las Rift S y las Meta Quest 2.

En cuanto a la funcionalidad alcanzada, el desarrollo de tres escenarios con objetivos diferentes nos ha permitido alcanzar gran parte de los objetivos secundarios planteados en la introducción. Principalmente, ofrecemos a los jugadores un entorno de diagnóstico mediante el test de Ishihara (escenario 1), una experiencia gamificada para empatizar con cuatro tipos de daltonismos que permitirán a los jugadores empatizar con estos problemas de visión (escenario 2), y por último una experiencia más realista basada en fotografías 360° con entornos cercanos a los usuarios en los que poder consolidar su aprendizaje.

Por último, se ha de mencionar que, aunque muy preliminares, hemos podido realizar algunas pruebas con usuario que nos han permitido obtener un *feedback* interesante que podría ser implementado en un futuro. Aunque se han detectado algunos puntos de mejora, todos los usuarios expresaron su satisfacción con la experiencia destacando como acertado las dinámicas propuestas para concienciar sobre este tipo de problemas.

4.1 TRABAJO FUTURO

Aunque algunas de ellas ya se han comentado anteriormente, en este apartado se han detectado algunas líneas de trabajo futuro que enumeramos a continuación:

- **Mejoras de narrativa:** en línea con las conclusiones del análisis de las pruebas con usuarios enumeradas en la sección 3.4 creemos que sería interesante mejorar la narrativa actualmente implementada con tutorial que combinan descripciones de texto e imágenes. Dotarlos de mayor interactividad haciendo uso de animaciones de los objetos 3D sería una mejora interesante.
- **Mejoras de aspectos técnicos:** también en línea de 3.4, se han localizado algunos errores relacionados con el despliegue de la *build* de la aplicación. Aunque error se ha producido con las Meta Quest 2, se debe revisar si pudieran producirse en otros sistemas y como solucionarlos.
- **Aumentar la funcionalidad y los escenarios:** en el contexto de un TFG la funcionalidad de la aplicación desarrollada debe tener un límite. Sin embargo, creemos que el prototipo desarrollado podría ser usado como base para generar nuevos escenarios o incluso niveles de juego que permitan aumentar la experiencia. Para ello sería necesario buscar financiación externa que permita poder abordar un desarrollo mayor.
- **Futuros trabajos de investigación:** sería interesante plantear un escenario completo que permita evaluar la herramienta desde diferentes enfoques. El diseño de pruebas con usuario más complejas y siguiendo

una metodología formal permitiría validar la usabilidad y la capacidad de la herramienta para concienciar sobre este tipo de problemas. Otro trabajo que podría surgir sería el desarrollo de módulos para integrar los símbolos de ColorADD en el juego o incluso integrarlo en entorno reales a través de paradigmas alternativos como la Realidad Aumentada.



5 BIBLIOGRAFÍA

Alomar. 2017. «DALTONISMO | Óptica Alomar Barcelona». Recuperado 6 de diciembre de 2022 (<https://opticaalomar.com/blog/daltonismo-2/>).

AltVR. 2020. «(60) Experience Colorblindness in Virtual Reality! - YouTube». Recuperado 6 de diciembre de 2022 (<https://www.youtube.com/watch?v=u1OpuTUIK8M>).

Amazon. 2022. «Oculus Rift S PC-Powered VR Gaming Headset : Amazon.Es: Videojuegos». Recuperado 8 de diciembre de 2022 (<https://www.amazon.es/Oculus-Rift-PC-Powered-Gaming-Headset/dp/B07PTMKYS7>).

Araujo, Joaquín. 2021. «¡En Otoño: Búscate En El Bosque!» Recuperado 3 de diciembre de 2022 (https://blogs.elconfidencial.com/medioambiente/naturaleza/emboscadas/2021-10-01/joaquin-araujo-buscate-bosque-otono_3299019/).

Área Oftalmológica Avanzada. 2020. «▷ ¿Qué Es El Test de Farnsworth? - Área Oftalmológica Avanzada». Recuperado 6 de diciembre de 2022 (<https://areaoftalmologica.com/terminos-de-oftalmologia/test-de-farnsworth/>).

Aukstakalnis, S., y D. Blatner. 1993. *El Espejismo de Silicio. Arte y Ciencia de La Realidad Virtual*.

Beck, Kent. 2000. «Extreme Programming EXplained : Embrace Change». 190.

Bitbucket. 2022. «Bitbucket | Git Solution for Teams Using Jira». Recuperado 7 de diciembre de 2022 (<https://bitbucket.org/>).

Blender. 2022a. «Blender.Org - Home of the Blender Project - Free and Open 3D Creation Software». Recuperado 7 de diciembre de 2022 (<https://www.blender.org/>).

- Blender. 2022b. «Historia de Blender — Blender Manual». Recuperado 7 de diciembre de 2022 (https://docs.blender.org/manual/es/2.82/getting_started/about/history.html).
- Calleros, González, C. B. ; Guerrero García, Navarro Rangel, Claudia B. González Calleros, Josefina Guerrero García, y Yadira Navarro Rangel. 2019. «Un Juego Serio Para La Solución de Problemas Matemáticos Para Niños Con TDAH». *Campus Virtuales* 8(2):121-40. doi: 10.7821/NAER.2019.7.434.
- CIS Tecnología e Diseño. 2021. «(Cave Automatic Virtual Environment) CIS Tecnología e Diseño CONTENIDO».
- Colblindor. 2021. «Ishihara's Test for Colour Deficiency: 38 Plates Edition – Colblindor». Recuperado 6 de diciembre de 2022 (<https://www.colorblindness.com/ishiharas-test-for-colour-deficiency-38-plates-edition/>).
- Colorite. 2022. «Colorlite - Test de Daltonismo». Recuperado 8 de diciembre de 2022 (<https://www.es.colorlitelens.com/test-de-daltonismo.html>).
- Creative Commons. 2022. «Creative Commons — CC0 1.0 Universal». Recuperado 8 de diciembre de 2022 (<https://creativecommons.org/publicdomain/zero/1.0/deed.es>).
- Cwierz, Halina, Francisco Diaz-Barrancas, Julia Gil Llinas, y Pedro J. Pardo. 2021. «On the Validity of Virtual Reality Applications for Professional Use: A Case Study on Color Vision Research and Diagnosis». *IEEE Access* 9:138215-24. doi: 10.1109/ACCESS.2021.3118438.
- dacangi. 2013. «Oculus Rift: Análisis». Recuperado 3 de diciembre de 2022 (<https://www.vidaextra.com/analisis/oculus-rift-analisis>).
- Darkcritz. 2019. «Spring La Nueva Película Animada de La Comunidad de Blender | Desde Linux». Recuperado 7 de diciembre de 2022 (<https://blog.desdelinux.net/spring-la-nueva-pelicula-animada-de-la-comunidad-de-blender/>).

- David Allen Company. 2022. «Getting Things Done® - David Allen's GTD® Methodology». Recuperado 12 de noviembre de 2022 (<https://getthingsdone.com/>).
- Díaz, Francisco Javier. 2015. «Juegos Serios y Educación». Recuperado 27 de noviembre de 2022 (<http://sedici.unlp.edu.ar/handle/10915/46458>).
- Díaz, Francisco Javier, Claudia Alejandra Queiruga, y Laura Andrea Fava. 2015. «Juegos Serios y Educación».
- Diego Levis. 2006. «(PDF) ¿Qué Es La Realidad Virtual ? | Diego Levis - Academia.Edu». Recuperado 8 de diciembre de 2022 (https://www.academia.edu/2449000/_Qu%C3%A9_es_la_realidad_virtual_).
- Dowling, Mikey. 2020. «Exploring Grounded's Arachnophobia Mode - Xbox Wire». Recuperado 3 de diciembre de 2022 (<https://news.xbox.com/en-us/2020/07/27/grounded-arachnophobia-mode-details/>).
- Elena. 2015. «Eye OK, Una Herramienta Para Jugar y Detectar Problemas Visuales - Educación 2.0». Recuperado 9 de noviembre de 2022 (<https://educacion2.com/eye-ok-una-herramienta-para-jugar-y-detectar-problemas-visuales/>).
- EnChroma. 2022. «EnChroma® Color Blind Test | Test Your Color Vision». Recuperado 8 de diciembre de 2022 (<https://enchroma.com/pages/color-blindness-test>).
- Epic Games. 2022. «The Most Powerful Real-Time 3D Creation Tool - Unreal Engine». Recuperado 6 de diciembre de 2022 (<https://www.unrealengine.com/en-US>).
- Fantasymundo. 2020. «Virtual Boy: Un Fracaso Rojo y Negro (II) - Fantasymundo». Recuperado 3 de diciembre de 2022 (<https://www.fantasymundo.com/virtual-boy-un-fracaso-rojo-y-negro-ii/>).
- Fernández, Yúbal. 2022. «Qué Es El Metaverso, Qué Posibilidades Ofrece y Cuándo Será Real». Recuperado 6 de diciembre de 2022 (<https://www.xataka.com/basics/que-metaverso-que-posibilidades-ofrece-cuando-sera-real>).

- de Fez Saiz, Dolores, M^a Amparo Díez Ajenjo, M^a Carmen García Domene, M^a José Luque Cobija, y Vicente Jesús Camps Sanchís. 2022. «Capítulo 3. Visión Del Color: Anomaloscopia». en *PRÁCTICAS DE PSICOFÍSICA Y PERCEPCIÓN VISUAL*. Universidad de Alicante.
- Flightdemy. 2022. «Link Trainer: El Primer Simulador de Vuelo - Aprende Aviación». Recuperado 3 de diciembre de 2022 (<https://flightdemy.com/blog/primer-simulador-vuelo>).
- FoldIt. 2022. «Foldit». Recuperado 27 de noviembre de 2022 (<https://fold.it/>).
- Frankie MB. 2018. «Cómo Jugar a Videojuegos Siendo Daltónico y “Sobrevivir” En El Intento». Recuperado 10 de noviembre de 2022 (<https://www.xataka.com/videojuegos/como-jugar-a-videojuegos-siendo-daltonico-y-sobrevivir-en-el-intento>).
- Garrido Sotomayor, Sandra. 2022. «Qué Son Las Metodologías Ágiles y Cuáles Son Sus Ventajas Empresariales». Recuperado 10 de diciembre de 2022 (<https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/>).
- Gifreu, Arnau. 2014. «La Realidad Virtual. Como Afectará a Los Webdocs. Parte 2 - Webdocs. Historias Del Siglo XXI». Recuperado 3 de diciembre de 2022 (<https://blog.rtve.es/webdocs/2014/12/la-realidad-virtual-como-afectar%C3%A1-a-los-webdocs-parte-5.html>).
- GitKraken. 2022. «GitKraken Legendary Git Tools | GitKraken». Recuperado 7 de diciembre de 2022 (<https://www.gitkraken.com/>).
- GNU. 2022. «Licencias - Proyecto GNU - Free Software Foundation». Recuperado 8 de diciembre de 2022 (<https://www.gnu.org/licenses/licenses.es.html>).
- González Calleros, Claudia Blanca, Josefina Guerrero García, y Yadira Novarro Rangel. 2019. «Un Juego Serio Para La Solución de Problemas Matemáticos Para Niños Con TDAH | González Calleros | Campus Virtuales». Recuperado 27 de noviembre de 2022 (<http://uajournals.com/ojs/index.php/campusvirtuales/article/view/506>).

- Google. 2022. «Google Cardboard – Google VR». Recuperado 3 de diciembre de 2022 (https://arvr.google.com/intl/es_es/cardboard/).
- Grefer, Alicia. 2022. «¿Cómo Ve Un Daltónico? Experiencia Real Con Gafas Para Daltonismo». Recuperado 10 de noviembre de 2022 (<https://cuidatuvista.com/gafas-daltonismo-colores/>).
- Guitarte, Aitor. 2011a. «Ciencia y Videojuegos Contra El SIDA - Somos Medicina». Recuperado 27 de noviembre de 2022 (<https://www.somosmedicina.com/2011/09/ciencia-y-videojuegos-contra-el-sida.html>).
- Guitarte, Aitor. 2011b. «Jugando Con La Obesidad Infantil - Somos Medicina». Recuperado 27 de noviembre de 2022 (<https://www.somosmedicina.com/2011/04/jugando-con-la-obesidad-infantil.html>).
- Harris, Blake J. 2014. *Console Wars: Sega, Nintendo, and the Battle That Defined a Generation*. HarperCollins.
- Harris, Craig. 2007. «Flash Focus: Vision Training in Minutes a Day - IGN». Recuperado 10 de noviembre de 2022 (<https://www.ign.com/games/flash-focus-vision-training-in-minutes-a-day>).
- hmong. 2022. «Shinobu Ishihara Vida Temprana CarrerayPrueba de Visión Del Color de Ishihara». Recuperado 6 de diciembre de 2022 (https://hmong.es/wiki/Shinobu_Ishihara).
- van Hoek, Susan., y Marion Clayton. Link. 2003. *From Sky to Sea: A Story of Edwin A. Link*. Flagstaff, Arizona: Best Publishing.
- Hugin. 2021. «Hugin - Panorama Photo Stitcher». Recuperado 8 de diciembre de 2022 (<https://hugin.sourceforge.io/>).
- iNFINITE Production. 2018. «Experience: Colorblindness En Steam». Recuperado 8 de diciembre de 2022 (https://store.steampowered.com/app/979100/Experience_Colorblindness/).

- iNFINITE Production. 2022. «Experience: Colorblindness En Steam». Recuperado 3 de diciembre de 2022 (https://store.steampowered.com/app/979100/Experience_Colorblindness/).
- Isaac. 2022. «GitKraken: Conoce Esta Interesante Herramienta Para Desarrolladores | Linux Adictos». Recuperado 7 de diciembre de 2022 (<https://www.linuxadictos.com/gitkraken-herramienta-desarrolladores-git.html>).
- J. Luque, M., M. D. de Fez, y M. A. Díez. 2001. «Directrices Para La Administración y Puntuación Del Test Farnsworth-Munsell de 100 Tonos». *Ver y Oír* 413-20.
- JetBrains. 2022a. «IntelliJ IDEA – the Leading Java and Kotlin IDE». Recuperado 6 de diciembre de 2022 (<https://www.jetbrains.com/idea/>).
- JetBrains. 2022b. «JetBrains: Essential Tools for Software Developers and Teams». Recuperado 12 de noviembre de 2022 (<https://www.jetbrains.com/>).
- JetBrains. 2022c. «Rider: El IDE .NET Multiplataforma de JetBrains». Recuperado 6 de diciembre de 2022 (<https://www.jetbrains.com/es-es/rider/>).
- JetBrains. 2022d. «Rider vs. Visual Studio - Compare | JetBrains Rider». Recuperado 11 de noviembre de 2022 (<https://www.jetbrains.com/rider/compare/rider-vs-visual-studio/>).
- Kickstarter. 2016. «Oculus Rift: Step Into the Game by Oculus — Kickstarter». Recuperado 3 de diciembre de 2022 (<https://www.kickstarter.com/projects/1523379957/oculus-rift-step-into-the-game>).
- Lasa Gómez, Carmen, Alonso Álvarez García, y Rafael de las Heras del Dedo. 2017. «Métodos Ágiles : Scrum, Kanban, Lean».
- Longitud. 2013. «EYE OK Juegos Orientados a La Mejora de La Salud Visual - Longitud de Onda». Recuperado 9 de noviembre de 2022 (<https://longitudeonda.com/eye-ok-juegos-orientados-a-la-mejora-de-la-salud-visual/?cn-reloaded=1>).
- López Escorial, María. 2015. «El Código 'Braille' Para Daltónicos Se Inventa En El Siglo XXI». *Planeta Futuro. El País*. Recuperado 9 de noviembre de 2022

(https://elpais.com/elpais/2015/07/24/planeta_futuro/1437744306_549219.html).

Marcano, Beatriz. 2008. «Juegos Serios y Entrenamiento En La Sociedad Digital». *Education in the Knowledge Society (EKS)* 9(3):93-107.

Márquez, R. 2013. «“The Last of Us”: Análisis». Recuperado 27 de noviembre de 2022 (<https://www.vidaextra.com/analisis/the-last-of-us-analisis>).

Martínez Botella, Pascual. 2022. «Desarrollo de Una Aplicación Para Simulación de Deficiencia Visual Mediante Realidad Virtual». Universidad Miguel Hernández, Elche.

Medellín, Hugo, Germánico González, Raquel Espinosa, Eder Govea, Theodore Lim, H. Medellin, G. Gonzalez, R Espinosa, E Govea, T Lim, M. Ramos, y V. Aguilera. 2014. «Desarrollo de Aplicaciones de Realidad Virtual y Sistemas Hápticos En Ingeniería, Medicina y Arte».

Medicineplus. 2021. «Daltonismo: MedlinePlus Enciclopedia Médica». Recuperado 9 de noviembre de 2022 (<https://medlineplus.gov/spanish/ency/article/001002.htm>).

Meirsschaut, Guillaume. 2022. «El Mejor Casco de Realidad Virtual 2022». Recuperado 3 de diciembre de 2022 (<https://selectos.eu/es/mejores-cascos-rv/>).

Melscience. 2022. «MEL VR Science Simulations». Recuperado 6 de diciembre de 2022 (<https://melscience.com/ES-en/vr/>).

Méndez, Julia. 2016. «Qué Tipos de Daltonismo Existen y Cuáles Son Sus Causas». Recuperado 9 de noviembre de 2022 (<https://www.clinicabaviera.com/blog/salud-visual/tipos-de-daltonismo/>).

Meneses, Nacho. 2021. «La Enseñanza Con Realidad Virtual Puede Multiplicar Por Cuatro La Retención de Conocimientos | Formación | Economía | EL PAÍS». Recuperado 6 de diciembre de 2022 (https://elpais.com/economia/2021/05/27/actualidad/1622109464_578539.html).

- Meta. 2021. «(60) Introducing Meta - YouTube». Recuperado 6 de diciembre de 2022 (https://www.youtube.com/watch?v=pjNI9K1D_xo).
- Meta. 2022a. «El Metaverso Es El Futuro de Las Conexiones Digitales | Meta». Recuperado 6 de diciembre de 2022 (<https://about.meta.com/es/metaverse/>).
- Meta. 2022b. «Meta Quest 2: Gafas Inmersivas de Realidad Virtual Todo En Uno | Meta Store | Meta Store». Recuperado 3 de diciembre de 2022 (<https://www.meta.com/es/quest/products/quest-2/>).
- Microsoft. 2022a. «Visual Studio Code - Code Editing. Redefined». Recuperado 6 de diciembre de 2022 (<https://code.visualstudio.com/>).
- Microsoft. 2022b. «Visual Studio: IDE y Editor de Código Para Desarrolladores de Software y Teams». Recuperado 6 de diciembre de 2022 (<https://visualstudio.microsoft.com/es/>).
- Microsoft. 2022c. «Visual Studio: IDE y Editor de Código Para Desarrolladores de Software y Teams». Recuperado 12 de noviembre de 2022 (<https://visualstudio.microsoft.com/es/>).
- Microsoft. 2022d. «Visual Studio IntelliCode | Visual Studio - Visual Studio». Recuperado 6 de diciembre de 2022 (<https://visualstudio.microsoft.com/es/services/intellicode/>).
- Microsoft. 2022e. «Visual Studio Live Share: Herramienta de Colaboración de Código En Tiempo Real». Recuperado 6 de diciembre de 2022 (<https://visualstudio.microsoft.com/es/services/live-share/>).
- Middleton, Marjorie. 2022. «How (& Why) to Create a Tickler File - HouseAffection». Recuperado 10 de diciembre de 2022 (<https://houseaffection.com/how-and-why-to-create-a-tickler-file/>).
- Mora, Juan Carlos. 2022. «Eye Ok | Edúkame». Recuperado 9 de noviembre de 2022 (<https://edukame.com/eye-ok>).

- Moss, Gabriel. 2019. «How to Play “Beat Saber”: Top 12 Tips & Tricks (From the Pros)». Recuperado 3 de diciembre de 2022 (<https://www.vrfitnessinsider.com/top-12-beat-saber-tips-tricks/>).
- National Academy of Sciences (U.S.). 1993. «Biographical Memoirs. Volume 62». 498.
- Nieva, Miguel. 2010. «ColorADD». Recuperado 10 de noviembre de 2022 (<https://www.coloradd.net/en/>).
- Opeluce Team. 2022. «Las 10 Enfermedades de La Vista Más Comunes | Blog D’Opeluce». Recuperado 9 de noviembre de 2022 (<https://www.opeluce.com.pe/blog/cuales-son-enfermedades-ojos-comunes/>).
- Órbita Laika. 2020. «Historia Del Daltonismo — Cuaderno de Cultura Científica». Recuperado 10 de noviembre de 2022 (<https://culturacientifica.com/2020/12/08/historia-del-daltonismo/>).
- Pau Surribas. 2010. «La Herencia Genética Del Daltonismo | Pau Surribas». Recuperado 3 de diciembre de 2022 (<https://pausurribas.wordpress.com/2010/09/12/la-herencia-genetica-del-daltonismo/>).
- Pingüino Digital. 2020. «EDITORES DE CODIGO: ¿QUE SON Y PARA QUE SIRVEN?». Recuperado 11 de noviembre de 2022 (<https://pinguinodigital.com/blog/editores-de-codigo/>).
- Pixexid. 2022. «Source of Free Images, Wallpapers and Backgrounds - Pixexid». Recuperado 8 de diciembre de 2022 (<https://pixexid.com/>).
- PlayStation. 2020. «The Last of Us Parte II - Accesibilidad (España)». Recuperado 6 de diciembre de 2022 (<https://www.playstation.com/es-es/games/the-last-of-us-part-ii/accessibility/>).
- PlayStation. 2022. «PlayStation®VR2 | La Nueva Generación de Realidad Virtual Para PS5 | PlayStation (España)». Recuperado 3 de diciembre de 2022 (<https://www.playstation.com/es-es/ps-vr2/>).

- Pozzi, Sandro. 2014. «Facebook Compra Oculus | Tecnología | EL PAÍS». Recuperado 7 de noviembre de 2022 (https://elpais.com/tecnologia/2014/03/26/actualidad/1395796446_034242.html).
- Quirell. 2017. «Circulo Cromático». Recuperado 8 de diciembre de 2022 (<https://www.quirell.es/modelos-de-color/>).
- Quiroa, Myriam. 2019. «Modelo Freemium - Qué Es, Definición y Concepto | 2022 | Economipedia». Recuperado 11 de noviembre de 2022 (<https://economipedia.com/definiciones/modelo-freemium-que-es-y-como-funciona.html>).
- Red Hat. 2019. «El Concepto de IDE». Recuperado 10 de noviembre de 2022 (<https://www.redhat.com/es/topics/middleware/what-is-ide>).
- Riera Muñoz, Damián. 2007. «Análisis Training for Your Eyes - NDS». Recuperado 9 de noviembre de 2022 (<https://vandal.elespanol.com/analisis/nds/training-for-your-eyes/7032#p-29>).
- de Rochefort, Simone. 2021. «Assassin's Creed Unity: Ubisoft Can't Help Rebuild Notre-Dame Cathedral - Polygon». Recuperado 6 de diciembre de 2022 (<https://www.polygon.com/features/22790314/assassins-creed-unity-notre-dame-restoration-accuracy>).
- Rodríguez Ruiz, Javier. 2022a. «Carpeta Compartida En Memoria». Recuperado 14 de diciembre de 2022 (<https://drive.google.com/drive/folders/1zXyMZBp4dDo7JqFE0gi1uc26Jf34tkw9?usp=sharing>).
- Rodríguez Ruiz, Javier. 2022b. «Roruja / Tfg — Bitbucket». Recuperado 10 de diciembre de 2022 (<https://bitbucket.org/roruja/tfg/src/master/>).
- Samsung. 2021. «Nuevo Gear VR | SM-R323NBKAPHE | Samsung Empresas España». Recuperado 6 de diciembre de 2022 (<https://www.samsung.com/es/business/wearables/gear-vr-r323/sm-r323nbkaphe/>).

- Shingo, Shigeo, y Andrew P. Dillon. 1989. *A Study of the Toyota Production System from an Industrial Engineering Viewpoint*. Productivity Press.
- Sketchfab. 2014. «Sketchfab Launches a Download Option and Reaches 200k 3D Files - Sketchfab Blog». Recuperado 8 de diciembre de 2022 (<https://web.archive.org/web/20141106003547/http://blog.sketchfab.com/post/100773163909/sketchfab-launches-a-download-option-and-reaches-200k>).
- Sketchfab. 2015. «Facebook Now Supports Sketchfab 3D Embeds! - Sketchfab Blog». Recuperado 8 de diciembre de 2022 (<https://web.archive.org/web/20150504014052/http://blog.sketchfab.com/post/112153509704/facebook-now-supports-sketchfab-3d-embeds>).
- Sketchfab. 2022. «Sketchfab - The Best 3D Viewer on the Web». Recuperado 8 de diciembre de 2022 (<https://sketchfab.com/>).
- Sky Optic. 2022. «Oculus HMC Anomaloscope with Software-Module for the Price 0,00€ Buy in Sky Optic». Recuperado 6 de diciembre de 2022 (<https://sky-optic.com/catalog/oculus-hmc-anomaloskope-with-software-module/>).
- The Witness. 2016. «The Witness – Explore an Abandoned Island.» Recuperado 8 de diciembre de 2022 (<http://the-witness.net/news/>).
- TickTick. 2022. «TickTick». Recuperado 7 de diciembre de 2022 (<https://ticktick.com/webapp/>).
- Tristem, Ben. 2022. «Unity vs. Unreal: Which One Is Best For You? - Udemy Blog». Recuperado 10 de noviembre de 2022 (<https://blog.udemy.com/unity-vs-unreal-which-game-engine-is-best-for-you/>).
- Ubisoft. 2020. «(60) Ubisoft's Notre-Dame de Paris : Journey to the Past in VR - YouTube». Recuperado 6 de diciembre de 2022 (<https://www.youtube.com/watch?v=zfY86nYVki0>).
- UML. 2022. «Welcome To UML Web Site!» Recuperado 4 de diciembre de 2022 (<http://uml.org/>).

- Unity. 2022a. «Plataforma de Desarrollo En Tiempo Real de Unity | Motor de VR y AR En 3D y 2D». Recuperado 6 de diciembre de 2022 (<https://unity.com/es>).
- Unity. 2022b. «Unity - Manual: Cameras». Recuperado 9 de diciembre de 2022 (<https://docs.unity3d.com/Manual/CamerasOverview.html>).
- Unity. 2022c. «Unity - Manual: Panoramic Skybox». Recuperado 9 de diciembre de 2022 (<https://docs.unity3d.com/Manual/shader-skybox-panoramic.html>).
- Unity. 2022d. «Unity - Scripting API: AsyncOperation». Recuperado 11 de diciembre de 2022 (<https://docs.unity3d.com/ScriptReference/AsyncOperation.html>).
- Unity. 2022e. «Unity - Scripting API: PlayerPrefs». Recuperado 10 de diciembre de 2022 (<https://docs.unity3d.com/ScriptReference/PlayerPrefs.html>).
- Versus. 2022. «Valve Index Análisis | 65 Características Detalladas». Recuperado 3 de diciembre de 2022 (<https://versus.com/es/valve-index>).
- Villoria. 2022. «¿Eres Daltónico? Compruébalo Con El Test de Ishihara - Clínica Villoria». Recuperado 8 de diciembre de 2022 (<https://www.clinicavilloria.es/eres-daltonico-comprueballo-con-el-test-de-ishihara/>).
- Wikipedia. 2011a. «ColorADD - Wikipedia, La Enciclopedia Libre». Recuperado 6 de diciembre de 2022 (<https://es.wikipedia.org/wiki/ColorADD>).
- Wikipedia. 2011b. «John Dalton - Wikipedia, La Enciclopedia Libre». Recuperado 10 de noviembre de 2022 (https://es.wikipedia.org/wiki/John_Dalton).
- Wikipedia. 2019. «Brain Age - Wikipedia». Recuperado 9 de noviembre de 2022 (https://en.wikipedia.org/wiki/Brain_Age).
- Wikipedia. 2022a. «Unity (Game Engine) - Wikipedia». Recuperado 10 de noviembre de 2022 ([https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine))).
- Wikipedia. 2022b. «Unreal Engine - Wikipedia». Recuperado 10 de noviembre de 2022 (https://en.wikipedia.org/wiki/Unreal_Engine).
- Wikipedia. 2022c. «Virtual Boy - Wikipedia, La Enciclopedia Libre». Recuperado 26 de noviembre de 2022 (https://es.wikipedia.org/wiki/Virtual_Boy).

Wolds, Stephen. 2020. «The Last of Us 2 Goes beyond Accessibility and Difficulty Levels - Polygon». Recuperado 3 de diciembre de 2022

(<https://www.polygon.com/2020/7/2/21310396/last-of-us-2-accessibility-vision-difficulty-gameplay-opinions>).

Xperimenta Cultura. 2016. «Historia de La Realidad Virtual – Xperimenta Cultura».

Recuperado 26 de noviembre de 2022 (<https://xperimentacultura.com/historia-de-la-realidad-virtual/>).

Zucconi, Alan. 2015. «Accessibility Design: Color Blindness - Alan Zuconni». Recuperado

8 de diciembre de 2022 (<https://www.alanzucconi.com/2015/12/16/color-blindness/>).



ANEXO I: CASOS DE USO

En este anexo se describen los siguientes casos de uso identificados durante el desarrollo del proyecto:

- a) Caso de Uso 01. MENÚ AUXILIAR
- b) Caso de Uso 02. CAMBIO DE ESCENARIO
- c) Caso de Uso 03. SALIR
- d) Caso de Uso 04. TEST ISHIHARA
- e) Caso de Uso 05. PLACA ANTERIOR
- f) Caso de Uso 06. JUEGO SERIO
- g) Caso de Uso 07. CAMBIO DE DIFICULTAD
- h) Caso de Uso 08. MODO LIBRE
- i) Caso de Uso 09. CAMBIO DE DALTONISMO
- j) Caso de Uso 10. VISOR 360
- k) Caso de Uso 11. CAMBIO IMAGEN DE VISOR

CU-01	MENÚ AUXILIAR
Actores	Jugador
Descripción	El jugador activa o desactiva el menú auxiliar
Precondición	
Secuencia normal	1. El jugador presiona el botón "Menu" de su controlador izquierdo
Postcondición	El jugador se encuentra en el escenario del laboratorio
Excepciones	

Tabla A1. Caso de Uso 01

CU-02	CAMBIO DE ESCENARIO
Actores	Jugador
Descripción	El jugador cambia de escenario
Precondición	<ol style="list-style-type: none"> 2. El jugador debe acceder al Menú Auxiliar 3. El debe pulsar el botón “Volver” en la primera pregunta del Test de Ishihara 4. El jugador debe terminar el Test de Ishihara 5. El jugador debe terminar el Juego Serio 6. El jugador debe encontrarse en el escenario de Visor de Imágenes 360
Secuencia normal	<ol style="list-style-type: none"> 1. El jugador presiona uno de los botones que cambian de escenario en función de la precondición que cumpla 2. El sistema cambia el escenario al escogido por el usuario
Postcondición	Se cambia el escenario activo
Excepciones	

Tabla A1. Caso de Uso 02

CU-03	SALIR
Actores	Jugador
Descripción	El jugador cierra la aplicación
Precondición	<ul style="list-style-type: none"> • El jugador tiene activo el Menú Auxiliar • El jugador tiene activo el panel de cambio de escenario en el laboratorio
Secuencia normal	<ol style="list-style-type: none"> 1. El jugador presiona el botón de “Salir” 2. El sistema de cierra
Postcondición	La aplicación se cierra
Excepciones	

Tabla A1. Caso de Uso 03

CU-04 TEST ISHIHARA	
Actores	Jugador
Descripción	El jugador realiza el test de Ishihara
Precondición	El jugador debe encontrarse en el Laboratorio
Secuencia normal	<ol style="list-style-type: none"> 1. El jugador presiona el botón de comenzar 2. El sistema muestra una placa de Ishihara y realiza una pregunta 3. El jugador responde presionando uno de los botones 4. El sistema registra su respuesta 5. El sistema pasa a la siguiente pregunta y vuelve al punto 2. En caso no haber más preguntas muestra una pantalla con los resultados
Postcondición	El jugador ha realizado el test de Ishihara
Excepciones	Si el jugador en lugar de presionar una de las respuestas, presiona el botón de "Volver" el sistema cargará la pregunta anterior, en caso de no haber, por ser esta la primera pregunta, cargará el panel de cambio de escenario

Tabla A1. Caso de Uso 04

CU-05 PLACA ANTERIOR	
Actores	Jugador
Descripción	El jugador intenta volver a la pregunta anterior de la prueba
Precondición	El jugador debe estar realizando la prueba de daltonismo
Secuencia normal	<ol style="list-style-type: none"> 1. El jugador presiona el botón de "Volver" 2. El sistema cambia su estado al de la pregunta anterior, volviendo a mostrar la placa, pregunta y opciones correspondientes
Postcondición	El sistema ha vuelto al estado de la pregunta anterior
Excepciones	Si el jugador hace esto en la primera pregunta, al no haber un estado anterior al que volver, activa el panel de cambio de escenario

Tabla A1. Caso de Uso 05

CU-06	JUEGO SERIO
Actores	Jugador
Descripción	El jugador realiza el juego serio
Precondición	El jugador debe darle al botón para iniciar el juego, que se encuentra en el escenario de Simulación
Secuencia normal	<ol style="list-style-type: none"> 1. El sistema inicia un temporizador de 10s 2. Activa un número de zorros aleatorios determinado por la dificultad 3. Escoge de forma aleatoria a uno de los zorros activos y cambia su color 4. El jugador intenta interactuar el con zorro que tiene color distinto, si falla pierde puntos y si acierta los consigue y el sistema vuelve al paso 2 5. Cuando el temporizador termina, se cambia el entorno para simular el siguiente daltonismo y el sistema vuelve al paso 1 6. Cuando se han realizado todas las simulaciones se muestra el panel de resultados
Postcondición	El jugador ha realizado el juego
Excepciones	

Tabla A1. Caso de Uso 06

CU-07	CAMBIO DE DIFICULTAD
Actores	Jugador
Descripción	El jugador cambia la dificultad del juego serio
Precondición	El jugador se encuentra en el escenario de Simulación justo antes de iniciar el juego o mientras tiene activa la pantalla de resultados
Secuencia normal	<ol style="list-style-type: none"> 1. El jugador presiona en la dificultad deseada dentro del panel de selección de dificultad del Menú Auxiliar 2. El sistema ajusta la dificultad
Postcondición	La dificultad del juego serio se ve modificada
Excepciones	

Tabla A1. Caso de Uso 07

CU-08	MODO LIBRE
Actores	Jugador
Descripción	El jugador disfruta el Modo Libre en el Escenario de Simulación
Precondición	El jugador debe haber terminado el Juego Serio
Secuencia normal	<ol style="list-style-type: none"> 1. El jugador presiona el botón de “Modo libre” presente en el panel de resultados del Juego Serio 2. El sistema desactiva el panel de resultados y activa un panel con información del daltonismo actual, otro de botonería con el selector de daltonismos y todos los zorros 3. El sistema escoge 7 zorros de forma aleatoria y cambia su color, estos zorros tendrán siempre el color especial independientemente del daltonismo actual
Postcondición	El jugador se encuentra en el Modo Libre
Excepciones	

Tabla A1. Caso de Uso 08

CU-09 CAMBIO DE DALTONISMO	
Actores	Jugador
Descripción	El jugador cambia el daltonismo simulado actual
Precondición	<ul style="list-style-type: none"> • El jugador se encuentra en el Modo Libre • El jugador se encuentra en el Escenario de Visor 360
Secuencia normal	<ol style="list-style-type: none"> 1. El jugador presiona el botón del daltonismo que desea simular 2. El sistema actúa en consecuencia bien cambiando los materiales de la escena si se encuentra en el Escenario de Simulación, bien modificando la cámara si se encuentra en el Escenario de Visor 360
Postcondición	Se ha cambiado el daltonismo que se simula
Excepciones	

Tabla A1. Caso de Uso 09

CU-10 VISOR 360º	
Actores	Jugador
Descripción	El jugador ve una imagen 360º y puede modificar su visión para simular un daltonismo con el que ver la imagen
Precondición	El jugador se encuentra en el Escenario de Visor 360
Secuencia normal	<ol style="list-style-type: none"> 1. El jugador vislumbra una imagen 360 2. El jugador puede simular el daltonismo que desee, si esto ocurre se mostrará un texto informativo acerca de este daltonismo
Postcondición	El jugador se encuentra en el Escenario de Visor 360
Excepciones	

Tabla A1. Caso de Uso 10

CU-11	CAMBIO IMAGEN DE VISOR
Actores	Jugador
Descripción	El jugador cambia la imagen 360º del Escenario de Visor 360
Precondición	El jugador se encuentra en el Escenario de Visor 360
Secuencia normal	<ol style="list-style-type: none"> 1. El jugador presiona el botón de “Siguiete imagen” en el Menú Auxiliar 2. El sistema pasa a mostrar la siguiente imagen de la lista
Postcondición	La imagen del Visor 360 cambia
Excepciones	Si no existen más imágenes se vuelve a la primera imagen

Tabla A1. Caso de Uso 11

