

**DESARROLLO DE UN VIDEOJUEGO EDUCATIVO PARA EL MANEJO DE RESIDUOS
SÓLIDOS – AZONI AR**



KEVIN DAVID CASTELLANOS RODRIGUEZ

CÓDIGO: 2010798

IDENTIFICACIÓN: C.C. 1.000.518.779

NICOLAS HENAO VARGAS

CÓDIGO: 2010288

IDENTIFICACIÓN: C.C. 1.000.288.294

NICOLAS BERNAL AYALA

CÓDIGO: 2010355

IDENTIFICACIÓN: C.C. 1.000.953.122

UNIVERSIDAD PILOTO DE COLOMBIA

FACULTAD DE INGENIERÍA

PROGRAMA DE INGENIERÍA MECATRÓNICA

BOGOTÁ, D.C.

2024

**DESARROLLO DE UN VIDEO JUEGO EN REALIDAD AUMENTADA AZONI AR – LA BÚSQUEDA DE LA
EDUCACIÓN DEL RECICLAJE**

KEVIN DAVID CASTELLANOS RODRIGUEZ

CÓDIGO: 2010798

IDENTIFICACIÓN: C.C. 1.000.518.779

NICOLAS HENAO VARGAS

CÓDIGO: 2010288

IDENTIFICACIÓN: C.C. 1.000.288.294

NICOLAS BERNAL AYALA

CÓDIGO: 2010355

IDENTIFICACIÓN: C.C. 1.000.953.122

**PROYECTO DE GRADO PARA OPTAR POR EL TÍTULO DE INGENIERO MECATRÓNICO DE LA
UNIVERSIDAD PILOTO DE COLOMBIA**

DIRECTOR:

M.SC. MARCO ANTONIO JINETE GOMEZ

Ing. Electrónico, Magister Automatización industrial

UNIVERSIDAD PILOTO DE COLOMBIA

FACULTAD DE INGENIERÍA

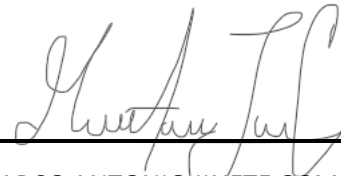
PROGRAMA DE INGENIERÍA MECATRÓNICA

BOGOTÁ, D.C.

2024

NOTA DE ACEPTACIÓN

Una vez realizada la revisión metodológica y técnica del documento final de proyecto de grado, doy constancia de que el (los) estudiante (s) ha cumplido a cabalidad con los objetivos propuestos, cumple a cabalidad con los Lineamientos de Opción de Grado vigentes del programa de Ingeniería Mecatrónica y con las leyes de derechos de autor de la República de Colombia, por tanto, se encuentra(n) preparado(s) para la defensa del mismo ante un jurado evaluador que considere idóneo el Comité de Investigaciones del Programa de Ingeniería Mecatrónica de la Universidad Piloto de Colombia.



MARCO ANTONIO JINETE GOMEZ

Director del Proyecto

DEDICATORIA

Esta tesis está dedicada a nuestras familias, amigos y personas queridas, que nos han apoyado no solo en este trabajo si no en todo el proceso de aprendizaje universitario.

AGRADECIMIENTOS

En esta tesis agradezco a todas esas personas presentes alrededor de la tesis que aportaron su granito de arena para hacer de este una realidad, personas como el profesor líder Marco Jinete y a los estudiantes Fernando Gonzales y José Noreña, siendo base del proyecto y donadores del chasis del robot el cual fue usado en transcurso de todo el proyecto. También agradecer a las personas externas que alimentaron el proyecto con sus consejos y críticas constructivas para que el proyecto avanzara por un buen camino como lo fue el semillero de investigación Open Creator, amigos y allegados, además de un agradecimiento especial a Jairito y a Tina que sin ellos nos estaría aquí.

Nicolás Henao Vargas

Me gustaría agradecer a todas las personas que nos han apoyado en esta tesis, la atención recibida es simplemente increíble, gracias a familiares, amigos, docentes y directivos dentro y fuera de la universidad Piloto por sus palabras de aliento para continuar en la tesis, y un agradecimiento especial a Marco Jinete, José Noreña y Fernando González sin ellos la tesis no sería lo que es hoy por último un agradecimiento a mis padres por ellos seré ingeniero.

Kevin David Castellanos Rodríguez

Me gustaría agradecer a familiares, amigos por sus palabras de aliento a lo largo de este proceso y un especial agradecimiento a mi pareja, sin ella no sería ingeniero.

Nicolas Bernal Ayala

TABLA DE CONTENIDO

	Pág.
NOTA DE ACEPTACIÓN	3
DEDICATORIA.....	4
AGRADECIMIENTOS	5
RESUMEN	9
ABSTRACT	9
 1. GENERALIDADES.....	 10
1.1. PLANTEAMIENTO DEL PROBLEMA	10
1.1.1. DESCRIPCIÓN DEL PROBLEMA.....	10
1.2. JUSTIFICACIÓN	10
1.3. OBJETIVOS	10
1.3.1. OBJETIVO GENERAL	10
1.3.2. OBJETIVOS ESPECÍFICOS.....	11
1.4. DELIMITACIÓN DEL PROYECTO	11
1.4.1. ALCANCES Y LIMITACIONES.....	11
1.5. MARCO REFERENCIAL.....	12
1.5.1. MARCO TEÓRICO.....	12
1.5.2. ESTADO DEL ARTE.....	13
 2. DISEÑO DEL VIDEOJUEGO.....	 16
2.1. HISTORIA.....	16
2.2. HERRAMIENTAS DE TRABAJO	16
2.2.1. UNITY	16
2.2.2. GITHUB.....	17
2.2.3. BLENDER	17
2.2.4. VUFORIA.....	17
2.2.5. PIXEL STUDIO.....	18
2.3. DISEÑO DE ELEMENTOS GRAFICOS.....	18
2.3.1. PALETA DE COLOR.....	18
2.3.2. CALIGRAFÍA	19
2.4. PERSONAJES.....	19
2.4.1. AZONI: EL ROBOT PROTAGONISTA	19
2.4.2. ENEMIGOS.....	20

2.6.1. AZONI	21
2.6.2. RESIDUOS	20
2.6.2. ENEMIGOS	22
2.6.1. RESIDUOS Y REFERENCIAS	21
2.7. MODELOS	23
2.7.1. TEXTURAS	24
2.8. ANIMACIÓN DE PERSONAJES	25
2.9. ESCENARIO	26
2.10. INTERFAZ	28
2.11. ILUMINACIÓN	30
2.11.1. RECURSOS ADICIONALES:	32
2.12. NAVMESH	32
2.12.1. LOGICA Y PROGRAMACION	35
2.12.2. LÓGICA ENTRE ESCENAS	35
2.12.3. REALIZAR LA TRANSICIÓN ENTRE ESCENAS	36
2.12.4. SISTEMA DE PUNTOS	36
3. IMPLEMENTACION DE LA REALIDAD AUMENTADA	38
3.1. PRIMERAS PRACTICAS	38
3.2. ESCALA DE GRISES	39
3.3. HISTOGRAMA	40
3.4. PIXELES CLAVES	41
3.5. MEJORES TARGETS	42
3.6. IMPLEMENTACION	49
3.6.1. SEGUIMIENTO Y DETECCIÓN DE TARGETS	49
3.7. GAMEPLAY	50
4. DESARROLLO DEL ROBOT	53
4.1. DISEÑO Y CONSTRUCCIÓN DEL ROBOT AZONI	53
4.1.1. DISEÑO GENERAL DEL ROBOT	54
4.1.2. ESPECIFICACIONES TÉCNICAS	54
4.1.3. MATERIAL UTILIZADO PARA EL ROBOT Y SUS BENEFICIOS	55
4.1.4. APOYO DEL PROYECTO PADRE AZONI TERB	¡Error! Marcador no definido.
4.1.5. CARCASA DEL ROBOT	56
4.2. SISTEMA DE LOCOMOCIÓN TIPO ORUGA	57
4.2.1. DESCRIPCIÓN DEL SISTEMA DE LOCOMOCIÓN TIPO ORUGA	57
4.2.2. VENTAJAS DEL SISTEMA DE LOCOMOCIÓN TIPO ORUGA	58
4.2.3. DESVENTAJAS DEL SISTEMA DE LOCOMOCIÓN TIPO ORUGA	58

4.2.4. FUERZA Y TRACCIÓN VS. VELOCIDAD	59
4.3. CONTROL DEL ROBOT.....	59
4.4. CONEXION INALAMBIRCA.....	60
5. RESULTADOS Y PRUEBAS DE FUNCIONAMIENTO.....	65
5.1. VALIDACIONES INICIALES.....	65
5.1.1. VALIDACIÓN VIDEOJUEGO	65
5.1.2. VALIDACIÓN REALIDAD AUMENTADA	67
5.1.3. VALIDACIÓN ROBOT	69
5.2. VALIDACIONES EXTERNAS	71
6. CONCLUSIONES	75
7. REFERENCIAS BIBLIOGRÁFICAS	76

RESUMEN

Azoni es un videojuego educativo que tiene como objetivo enseñar a las personas, la importancia de la correcta separación de residuos y cuáles son los contenedores adecuados para cada tipo de desecho. De esta manera, cuando un reciclador vaya a recoger estos desechos, las personas podrán aprovechar los materiales de mejor manera, evitando así perder la mayoría de ellos en los vertederos locales. Con esto, creamos un efecto dominó en el que todos podemos contribuir positivamente al medio ambiente donde tiene diferentes ramas de trabajo, una de ellas es un video juego que busca enseñar al público de una manera diferente a lo convencional, de esta manera nace la idea de AZONI AR una subrama de AZONI TERB el cual busca complementar el video juego con una nueva modalidad, la cual es la realidad aumentada, dando la posibilidad a los jugadores de llevar el videojuego al mundo real haciendo que ellos tengan que moverse por diferentes espacios de su hogar y así puedan estar junto a su familia, también busca no ser complicado para que cualquier tipo de persona mayor de los 12 años y menor de los 50 años pueda usarlo sin ninguna dificultad. Por ahora el proyecto será manejado únicamente por sus desarrolladores, ya que como se menciona en el archivo inicial, este será manejado en espacios controlados.

Palabras Clave: videojuego, realidad aumentada, interactivo.

ABSTRACT

Azoni is an educational video game aimed at teaching people the importance of proper waste separation and which containers are suitable for each type of waste. In this way, when a recycler comes to collect these waste materials, people can make better use of them, thereby avoiding losing the majority of them in local landfills. With this, we create a domino effect in which everyone can positively contribute to the environment. It has different branches of work, one of them being a video game that seeks to teach the public in a different way than conventional methods. Thus, the idea of AZONI AR is born as a sub-branch of AZONI TERB, which aims to complement the video game with a new modality, augmented reality, giving players the possibility to bring the video game into the real world, requiring them to move through different spaces in their homes so they can be with their families. It also aims not to be complicated so that any person over 12 years old and under 50 years old can use it without any difficulty. For now, the project will be managed solely by its developers, as mentioned in the initial file, and will be operated in controlled spaces.

Keywords: video game, augmented reality, interactive.

1. GENERALIDADES

1.1. PLANTEAMIENTO DEL PROBLEMA

La falta de conocimiento sobre cómo clasificar los residuos en categorías orgánicos, no aprovechables y aprovechables, está ocasionando procesos costosos en el aprovechamiento de los residuos.

1.1.1. DESCRIPCIÓN DEL PROBLEMA

En Colombia existe un código de colores para las canecas de la basura esto con el objetivo de separar los residuos en tres categorías orgánicos, aprovechables y no aprovechables, y así hacer más óptimo el procesamiento de residuos sin embargo la mayoría de los colombianos no conocen este código, es más, muchos edificios y espacios públicos no cuentan con este código en consecuencia las basuras se desechan de forma inadecuada dando casos que residuos que se pueden reutilizar se confundan con otros que no además es posible que residuos aprovechables se contaminen o dañen y el proceso para tratarlo se vuelve más largo o complicado, es decir se están generando gastos y procesos que es posible ahorrarse con los conocimientos adecuados, por lo tanto se propone crear un juego que ayuda a identificar como se deben separar los residuos según la norma colombiana y así hacer los procesos más óptimos de reciclaje o reutilización, aportando un granito de arena al cuidado del medio ambiente.

1.2. JUSTIFICACIÓN

Nuestra propuesta es la creación de un videojuego educativo respecto a la separación de basura, con el objetivo de llegar a un público aún más amplio se decidió que sea un videojuego para dispositivos móviles, aparatos que hoy en día casi toda persona cuenta, con el objetivo de brindar una verdadera experiencia educativa el mensaje estará presente en las mecánicas del juego, es decir para poder ganar u obtener la victoria será necesario conocer cómo se separan las basuras, además de ello para poder tener una experiencia aun didáctica se añadió un robot junto a la realidad aumentada, con el fin de no ser un juego monótono, el jugador podrá moverse junto al robot para cumplir las misiones y objetivos esparcidas por el espacio que el usuario decida por ultimo usar la realidad aumentada para poder interactuar con el mundo virtual y el robot, sin olvidar el mensaje que se desea dar.

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

Diseñar e implementar un videojuego interactivo centrado en la enseñanza de la separación de residuos sólidos ordinarios en un entorno de realidad aumentada para dispositivos móviles.

1.3.2. OBJETIVOS ESPECÍFICOS

- Implementar un sistema de locomoción y comunicación que permita la manipulación del robot de forma remota desde el videojuego para facilitar su interacción con el entorno.
- Implementar los algoritmos para la detección de símbolos que permitan la interacción con los modelos virtuales y la realidad vista por el usuario.
- Diseñar y construir el armazón exterior del robot siguiendo los lineamientos del modelo del robot Azoni.
- Desarrollar los diferentes niveles y mecánicas del videojuego además de las interacciones con el robot.
- Validar el funcionamiento del videojuego en condiciones controladas de laboratorio.

1.4. DELIMITACIÓN DEL PROYECTO

La tesis en primer lugar se encuentra delimita en conocimientos, los integrantes no contamos con mucha experiencia en el desarrollo de videojuegos como el uso de la realidad aumentada por lo cual no se aspira a realizar un juego e interacción muy complejas, por otro lado, está la financiación, esta se da por nuestros medios por cual nos veremos limitados en los materiales, ya sean para el robot o en software.

1.4.1. ALCANCES Y LIMITACIONES

1.4.1.1. ALCANCES

- Construir un robot con tracción oruga para moverse en terrenos controlados donde se realizarán pruebas de campo, en superficie plana, su control será manual.
- Este robot será diseñado y elaborado para espacios de interior y donde el terreno sea ligeramente irregular.
- La temática del videojuego será sobre gestión de residuos sólidos específicamente la identificación del código de colores para la separación de residuos sólidos en la fuente establecido en el artículo 4 de la Resolución 2184 de 2019.
- El proyecto estará enfocado a jóvenes entre 13 a 17 años en su mayoría.
- Este robot tendrá su propia fuente de energía mediante baterías.

1.4.1.2. LIMITACIONES

- La mecánica del robot en general se tratará de la posibilidad de movimiento en espacio limitado donde el usuario podrá ver en su dispositivo móvil donde están ubicadas las diferentes estaciones u objetos con los cuales podrá interactuar digitalmente.
- El robot no será autónomo ya que se espera que el jugador interactúe con este todo el tiempo y logre enseñar a reciclar.
- Los brazos del robot físico no tendrán movimiento, ya que la interacción será virtual.
- El sistema se validará en condiciones controladas de iluminación.

1.5. MARCO REFERENCIAL

1.5.1. MARCO TEÓRICO

El marco teórico de los videojuegos es un campo interdisciplinario que abarca una amplia gama de disciplinas, desde la psicología hasta la sociología, pasando por la informática y el diseño de juegos. Aquí hay una breve descripción de algunos aspectos importantes dentro de este marco teórico:

- **Teoría del Juego:** Este campo explora los elementos fundamentales que constituyen un juego, como las reglas, la mecánica y la interactividad. También se examina cómo estos elementos afectan la experiencia del jugador y contribuyen a la jugabilidad.
- **Psicología del Juego:** Se centra en entender cómo los videojuegos afectan el comportamiento humano, incluyendo aspectos como la motivación, la inmersión, la adicción y el aprendizaje. La teoría cognitiva y conductual son fundamentales para comprender cómo los videojuegos impactan en la mente y el comportamiento de los jugadores.
- **Diseño de Juegos:** Esta área se enfoca en los procesos creativos y técnicos involucrados en la creación de videojuegos. Incluye aspectos como el diseño de niveles, la narrativa, la jugabilidad, la estética visual y sonora, así como la usabilidad y la accesibilidad.
- **Estudios Culturales y Sociales de los Videojuegos:** Examina cómo los videojuegos se relacionan con la cultura, la sociedad y la identidad. Esto incluye el análisis de temas como la representación de género, la diversidad cultural, los estereotipos, así como el papel de los videojuegos en la formación de comunidades y subculturas.
- **Economía y Mercado de los Videojuegos:** Se ocupa del estudio de la industria de los videojuegos desde una perspectiva económica, analizando aspectos como la producción, distribución, consumo y monetización de los videojuegos. También aborda cuestiones como los modelos de negocio, la competencia en el mercado y las tendencias de consumo.
- **Ética y Videojuegos:** Explora cuestiones éticas relacionadas con los videojuegos, como la representación de la violencia, el sexismo, la privacidad de los datos, el comportamiento tóxico en línea y el impacto de las micro transacciones en los jugadores.
- **Educación y Videojuegos:** Se centra en el uso de los videojuegos como herramientas educativas, analizando cómo pueden ser empleados para promover el aprendizaje, el desarrollo de habilidades y la resolución de problemas en contextos educativos formales e informales.

Este campo está en constante evolución a medida que los videojuegos siguen siendo una forma de entretenimiento cada vez más popular y relevante en la sociedad contemporánea.

En el ámbito del estudio tenemos:

- **Aprendizaje Basado en Juegos:** Esta metodología educativa utiliza los principios del diseño de juegos para involucrar a los estudiantes en actividades de aprendizaje. Los videojuegos pueden proporcionar entornos inmersivos y motivadores que fomentan la participación y el aprendizaje significativo.
- **Desarrollo de Habilidades Cognitivas y Sociales:** Algunos videojuegos están diseñados específicamente para desarrollar habilidades cognitivas como el pensamiento crítico, la resolución de problemas y la toma de decisiones. Además, los juegos multijugador en línea

pueden promover habilidades sociales como la colaboración, la comunicación y el trabajo en equipo.

- **Simulaciones Educativas:** Los videojuegos pueden simular situaciones del mundo real y proporcionar a los estudiantes la oportunidad de experimentar y practicar habilidades en un entorno seguro y controlado. Por ejemplo, los simuladores de vuelo son utilizados para enseñar a pilotos, mientras que los juegos de gestión empresarial pueden ayudar a comprender conceptos de negocios o como en el caso del proyecto un simulador de la buena separación de residuos.
- **Personalización del Aprendizaje:** Los videojuegos pueden adaptarse a las necesidades individuales de los estudiantes, proporcionando retroalimentación inmediata y niveles de dificultad ajustables. Esto permite un aprendizaje personalizado que se adapta al ritmo y estilo de aprendizaje de cada estudiante.
- **Motivación y Compromiso:** Los videojuegos suelen ser altamente motivadores y pueden aumentar el compromiso de los estudiantes con el contenido educativo. La posibilidad de alcanzar objetivos, desbloquear logros y competir con otros puede aumentar la motivación intrínseca de los estudiantes para aprender.
- **Narrativa y Emoción:** Los videojuegos a menudo cuentan historias emocionantes y envolventes que pueden captar el interés de los estudiantes y crear conexiones emocionales con el material educativo. Esto puede facilitar la retención de información y el aprendizaje a largo plazo.
- **Evaluación Formativa:** Los videojuegos pueden utilizarse como herramientas de evaluación formativa, permitiendo a los educadores recopilar datos sobre el progreso y el desempeño de los estudiantes de forma continua y no intrusiva. Esto puede ayudar a identificar áreas de mejora y personalizar la instrucción.

Integrar los videojuegos en la educación requiere un enfoque equilibrado que tenga en cuenta tanto los objetivos educativos como las necesidades de los estudiantes. Cuando se implementan de manera efectiva, los videojuegos pueden ser una herramienta poderosa para mejorar la experiencia de aprendizaje y alcanzar los resultados educativos deseados.

1.5.2. ESTADO DEL ARTE

1. Chanchí-Golondrino, G.E., Gómez-Álvarez, M.C.y Sierra-Martínez, L. M.(2022).Directrices para el diseño y la construcción de videojuegos serios educativos. Revista Colombiana de Educación, 1(84), 1-22. <https://doi.org/10.17227/rce.num84-12759>

Usando como base el videojuego de lightbot, un juego educativo enfocado en enseñar programación de software, el artículo proponer un total de 51 directrices para el desarrollo y diseño de videojuegos educativos serios, estas matrices se enfocan principalmente en brindar una experiencia sencilla, accesible e interactiva, el artículo crea tres grupos claves de directrices para el desarrollo del juegos, el primero es la Usabilidad, en resumen es la relación usuario interfaz y gráficos, las directivas de este grupo en un principio proponen la creación de interfaces simples y vistosas que dejen bien claro el propósito de estas además de ello propone incluir elementos motivacionales para incentivar al jugador, llamar su atención y no aburrir.

Por otro lado, tenemos el grupo de Educación, enfocado en la guía del usuario, por ejemplo:

“DJ9. El videojuego serio debe dar la posibilidad de superar los retos de cada nivel con diferentes grados de cumplimiento o niveles de puntaje con el fin de que el jugador tenga una mayor motivación de superarse.”

Implicando dar mensajes claros respecto a lo que está haciendo, felicitando al jugar o bien corrigiendo con el fin de que el jugador entienda lo que debe hacer y se sienta retado, aumentado su interés.

Por último, el grupo de Juego, este se enfoca en dar una experiencia grata al jugador, es decir que no tenga solo un reto sino un puntaje a superar acompañada de una realimentación positiva no solo para divertir al usuario sino también, y principalmente, para motivar a continuar con el juego.

Aunque las pautas están enfocadas en juegos para el aprendizaje de programación de software algunas directrices pueden ser de utilidad como guía para poder desarrollar un videojuego en verdad enfocado en la educación.

2. Ortiz-Clavijo, L. F. y Cardona-Valencia, D. Tendencias y desafíos de los videojuegos como herramienta educativa. Revista Colombiana de Educación, 1(84), 1-17. <https://doi.org/10.17227/rce.num84-12761>

En artículo en cuestión cita una gran variedad de autores los cuales destacan las tendencia y los desafíos a la hora de usar videojuegos para enseñar entre las tendencias destaca cómo a partir de la pandemia el uso de tecnologías para enseñar se resaltó, demostrando como este a pesar de ser muy estudiado es poco aplicado dando a conocer la poca veracidad de la efectividad de este “nuevo método de educación” destacando también la malas perspectivas que se tiene por delante, entrando a la parte de los desafíos, el más importante a destacar es respecto al tiempo, teniendo como problema el tiempo que se debe dedicar a un juego para poder dar con el objetivo, enseñar.

Se menciona que en el documento “Computer Games in Education” (Mayer, 2019) como siempre hay una comparativa de este método con el tradicional destacando la relación jugador-videojuego donde radica en verdad el reto, ya que el jugador puede enfocarse en el apartado “divertido” y dejar de lado el educativo, dando a entender que el mensaje debe de estar fuertemente relacionado con la diversión para brindar una buena experiencia educativa.

3. Isodri Gomez, “Diseño y desarrollo de una aplicación para dispositivos móviles de realidad aumentada”, Tesis, Instituto politécnico nacional (2016). <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKewjYp5b3vbb9AhUAjLAFHeAxCQUQFnoECBgQAQ&url=https%3A%2F%2Ftesis.ipn.mx%2Fbitstream%2Fhandle%2F123456789%2F24066%2FTesis%2520Maestr%25C3%25ADa%2520PTA%2520G%25C3%25B3mez%2520Vargas%252C%2520Isidro.pdf%3Fsequence%3D1%26isAllowed%3Dy&usg=AOvVaw1G7crCzx4QO5ySHmUUph9u>

“Esta aplicación se diseñó con la finalidad de que sirva como material didáctico en un curso formal de geometría básica.”

La Figura 1-1, ilustra el funcionamiento de la aplicación ejemplo de realidad aumentada, donde se observa figuras geométricas a partir de un target o imagen la cual al ser observada por la cámara esta activara la forma geométrica.

El trabajo para titulación presenta una idea para el desarrollo de aplicaciones móviles usando aplicaciones de bajo costo como Vuforia, un kit de desarrollo de software de licencia libre para móviles enfocado en realidad aumentada, además de usar Unity 3D ambos basados en códigos en C#.

El desarrollo consiste en crear una relación entre Unity y Vuforia, Unity cumple la tarea de ser una plataforma, es decir, gracias a las funciones implementadas para desarrollo fue posible hacer cosas como exportar el proyecto a dispositivos móviles, acceder a la cámara del móvil, generar interfaces de forma rápida y generar las instrucciones, por otro lado, Vuforia se encarga de la generación de los objetos 3D.

La programación se basó en primer lugar hacer de Vuforia para obtener una base de datos de los objetos 3D, el sitio web oficial permite dar con esta base en 3D o 2D, seguido generar una cámara RA, con la base datos una vez conseguida y la cámara configurada se debe enlazar el SDK con Unity, la licencia y reemplazar la pantalla principal por último crear código dentro de Unity, con la capacidad de conocer los vértices de la figura, generarla y deformarla esto a la orden del usuario. este fue el resultado:

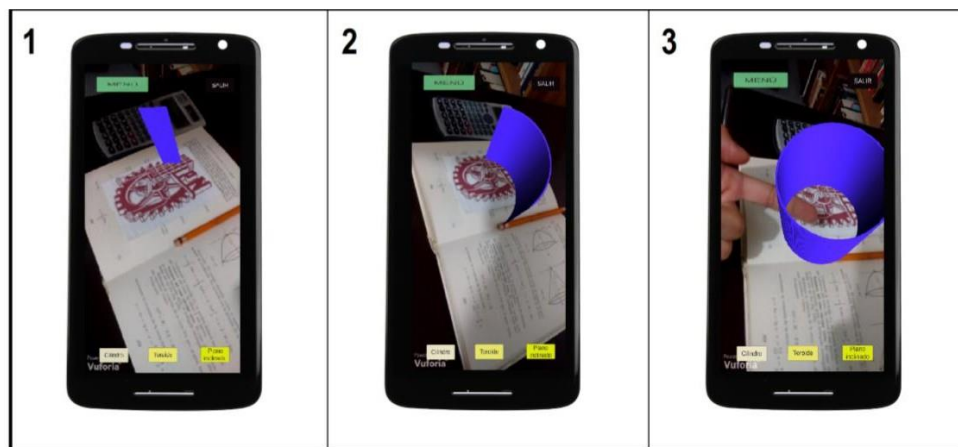


Figura 1-1 Funcionamiento de aplicación de realidad aumentada. Imagen tomada de *Construcción en RA de un cilindro* [Fotografía] Sanchez, D. Garcia, R (2016) [1]

2. DISEÑO DEL VIDEOJUEGO

2.1. HISTORIA

El videojuego "AZONI" se desarrolla en un mundo postapocalíptico de alta tecnología, pero devastado por la contaminación y la acumulación masiva de desechos tóxicos que han dado origen a seres conocidos como "desechos vivientes". Estos seres destructivos amenazan la existencia de la humanidad, y la única esperanza reside en el robot de guerra Azoni, creado por el científico Antonio Gómez Giraldo. Los jugadores asumen el rol de héroes como Ana y Juan, quienes deben viajar a través de diversas regiones colombianas, enfrentando monstruosas criaturas y recolectando muestras genéticas de desechos para encontrar una manera de detener esta amenaza. La narrativa del juego se desarrolla en tres actos: el establecimiento del contexto y personajes, la confrontación con desafíos crecientes, y la resolución con un enfrentamiento final contra la amenaza principal. La misión culmina en Cartagena, donde el destino de la humanidad depende de su éxito en la eliminación de los desechos vivientes. Con una mezcla de elementos de aprendizaje, exploración y combate, el juego busca educar sobre la importancia del manejo de residuos mientras ofrece una experiencia inmersiva en un entorno caótico y visualmente impactante.

Video: <https://youtu.be/8nm7wTm-iG8>

La dinámica del videojuego se centra en el robot Azoni, inventado por el doctor Antonio Gómez. Azoni deberá eliminar los diferentes residuos que han mutado debido a la contaminación, lanzándole un rayo purificador con la capacidad de devolverlos a su estado original. Después de esto, Azoni deberá llevar estos residuos a una caneca, donde serán separados en sus respectivas categorías dentro del mundo de realidad aumentada. Luego, Azoni continuará su labor explorando el mapa y regresando a través de un portal para seguir con su misión.

2.2. HERRAMIENTAS DE TRABAJO

Para la elaboración del proyecto se han necesitado diferentes programas Unity, Github, Blender y Visual Studio Code, Adicionando usamos extensiones de Unity, pero la más destacada es Vuforia, también se han usado otro tipo de aplicaciones, pero por muy corto tiempo, como lo son Pixel Studio y Krita.

2.2.1. UNITY

Unity es un motor de desarrollo en tiempo real que te permite crear experiencias interactivas en el Editor de Unity que se utiliza para la creación de videojuegos. Estos se pueden publicar en diversas plataformas como PC, videoconsolas, móviles, etc.

Gracias a su flexibilidad es una herramienta que también se usa en diferentes industrias como arquitectura, ingeniería, automotriz y de entretenimiento. (Unity 2024)

Como se menciona en la definición extraída de la página oficial de soporte de unity esta nos permite dar vida al proyecto ya que está diseñada para la elaboración de videojuegos, también como podemos leer esta aplicación nos permite la exportación del proyecto a la plataforma de móviles, gracias a esto podemos llevar el proyecto a los dispositivos más usados por las personas y así mostrar el proyecto de una manera más rápida y eficientemente.

Con esta herramienta podremos exportar todos los modelos realizados en Blender o cualquier tipo de aplicación que esta permita, también códigos y archivos multimedia, como imágenes, animaciones o texturas.

Obtenido de: <https://support.unity.com/hc/es/articles/7642130833812-Unity-Qué-es-y-cómo-funciona->

2.2.2. GITHUB

Github es un portal creado para alojar el código de las aplicaciones de cualquier desarrollador, y que fue comprada por Microsoft en junio del 2018. La plataforma está creada para que los desarrolladores suban el código de sus aplicaciones y herramientas, y que como usuario no sólo puedas descargarte la aplicación, sino también entrar a su perfil para leer sobre ella o colaborar con su desarrollo. (Xataka 2024)

Github es la herramienta que ha venido utilizando para guardar el proyecto en una nube. Gracias a esto el proyecto lo podemos abrir desde diferentes lados sin necesitar el mismo dispositivo y además evitamos el riesgo de que se pierda la información si algo les pasa a los computadores encargados para el desarrollo.

2.2.3. BLENDER

Con Blender, es posible crear visualizaciones 3D, imágenes fijas, animaciones 3D y efectos visuales. Es posible incluso editar videos. Se adapta muy bien a individuos independientes y estudios pequeños que se benefician de su canalización unificada y su proceso de desarrollo receptivo. (Blender 2024)

Blender se escogió en el proyecto ya que además de ser una herramienta gratuita, es una de las más fuertes y usadas en el mercado indie o de trabajo independiente, como se menciona en el texto anterior esta es la herramienta con la cual fueron diseñados los diferentes enemigos, personaje y elementos del mapa.

2.2.4. VUFORIA

Vuforia es una plataforma de desarrollo de aplicaciones de Realidad Aumentada (AR) y Realidad Mixta (MR) multiplataforma, con seguimiento robusto y rendimiento en una variedad de hardware (incluyendo dispositivos móviles y monitores de realidad mixta montados en la cabeza (HMD) como Microsoft HoloLens). La integración de Unity en Vuforia le permite crear aplicaciones y juegos de visión para Android y iOS utilizando un flujo de trabajo de creación de arrastrar y soltar. (Unity Documentation 2024)

Esta es una extensión de Unity el cual se incluyó en el proyecto para activación del AR o realidad aumentada, agregando el reconocimiento y seguimiento de imagen en tiempo real lo que nos permite crear interacción en objeto e imagen, este proceso es explicado con más detalle más adelante.

2.2.5. PIXEL STUDIO

Pixel Studio es un nuevo editor de pixel art para artistas y desarrolladores de videojuegos. Simple, rápido y portátil. No importa si eres un principiante o un profesional. (Pixel Studio 2024)

Como se menciona esta es una aplicación de diseño pixel art, en el proyecto se incluyen este tipo de imágenes para la implementación de diferentes estilos de arte para conectar de una manera amigable con el usuario.

2.3. DISEÑO DE ELEMENTOS GRAFICOS

2.3.1. PALETA DE COLOR

En el diseño del videojuego se busca que se vea futurista y que demuestre las máquinas y robots en este, por esta razón se opta por usar el tema ciberpunk.

El ciberpunk es un subgénero de la ciencia ficción, conocido por reflejar visiones distópicas del futuro en las cuales se combinan la tecnología avanzada con un bajo nivel de vida. (Wikipedia)

Con esta aclaración de que es ciberpunk llegamos a la conclusión de usar diferentes colores neón que nos acompañaran la parte de creación de interfaz y caligrafía del proyecto. Estos colores nos darán una identidad visual coherente y atractiva, que refleje la esencia de nuestro proyecto.

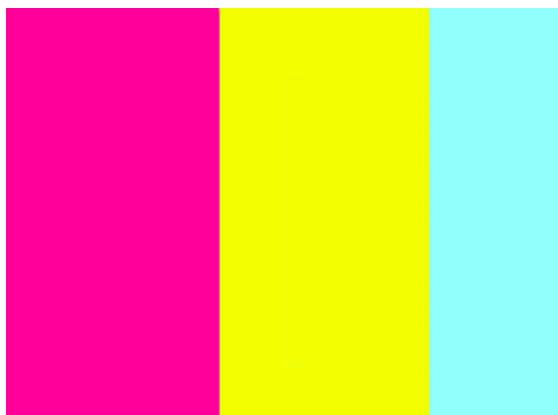


Figura 2-1 Paleta de color designada para el proyecto. Imagen tomada de *Krita paleta de color* [Ilustración] Azoni AR (2024) [2]

Diseño elementos gráficos (introducir paleta de colores, ejemplos de escenarios, luces y referencias al diseño)

2.3.2. CALIGRAFÍA

Luego de haber definido la paleta de colores en la Figura 2-1, se inicia la creación de nuestro logo. La caligrafía utilizada en el logo se ha seleccionado cuidadosamente para demostrar lo futurista en este, garantizando una apariencia armoniosa y profesional. Este proceso es fundamental para establecer una imagen de marca sólida y memorable.

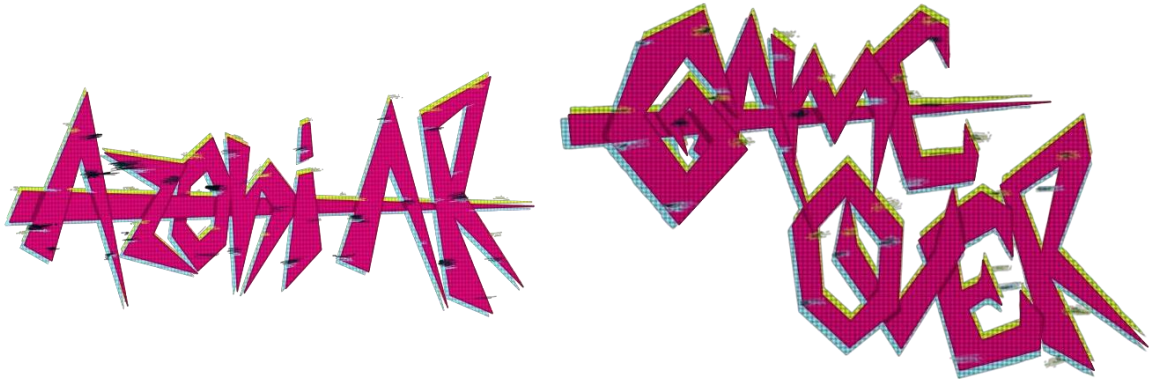


Figura 2-2 Imagen tipo logotipo del proyecto Azoni AR. Imagen tomada de *Krita logotipo Azoni AR* [Ilustración] Azoni AR (2024) [3]

2.4. PERSONAJES

2.4.1. AZONI: EL ROBOT PROTAGONISTA

Como protagonista de la historia tenemos a Azoni Figura 2-3, un robot que, con ayuda del jugador, tendrá como objetivo eliminar a los residuos.

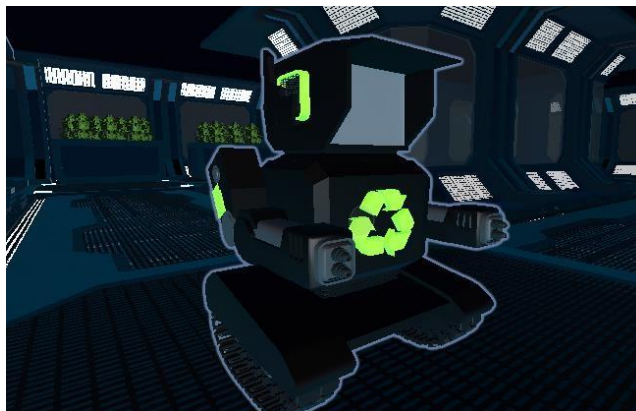


Figura 2-3 Robot principal – personaje jugable. Imagen tomada de *Azoni: El robot protagonista* [Pantallazo] Azoni AR (2024) [4]

2.4.2. ENEMIGOS

En el apartado de enemigos tenemos tres tipos, una caja, un pitillo y una acumulación de botellas, observar Figura 2-4. El motivo de estos residuos son uno de los más frecuentes en el día a día.



Figura 2-4 Enemigos – residuos mutantes.

- El enemigo de cartón representando a todos los residuos tipo cartón.
- El enemigo con forma de pitillo en representación a esos residuos pequeños del día a día.
- El enemigo de la unión de botellas y residuos representando a uno de los desechos más comunes como son las botellas y además de los cúmulos de basura.

Estos enemigos tienen como propósito contagiar más basura e invadir el mundo destruyendo el medioambiente, pero también de huir de Azoni el robot de limpieza, por este motivo en el videojuego podremos ver como los enemigos huyen de este.

2.5. RESIDUOS

También vemos la selección de los diferentes residuos en la Figura 2-5 que se utilizarán en el proyecto donde vemos una botella plástica como resibo aprovechable una manzana como residuo orgánico una madera con tablones para residuos no aprovechable y también una bolsa de cartón para residuo aprovechable.

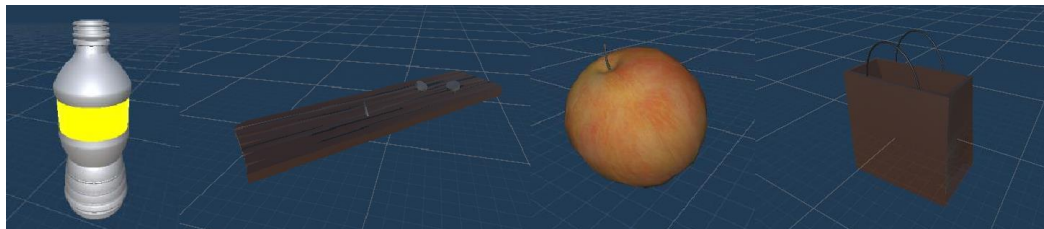


Figura 2-5 Residuos – Orgánicos, Aprovechables y No aprovechables

Este es el modelo de la caneca destinada a la separación de residuos donde será Re coloreada con verde, negro y blanco; Esto será para seguir el código de colores establecido por el gobierno colombiano.



Figura 2-6 Modelo base para la creación de las canecas, Imagen tomada de *Modelo caneca* [Modelo .fbx] Azoni TERB (2023) [5]

Los modelos y residuos fueron tomados del del videojuego Azoni desarrollado en marco del proyecto TERB, cuyo grupo de desarrollo está compuesto en parte por autores de este trabajo de grado.

2.6. BOCETOS Y REFERENCIAS

2.6.1. AZONI

El desarrollo de Azoni, nuestro robot protagonista, comenzó con una serie de bocetos que buscaban capturar su esencia como un defensor del medio ambiente en un mundo futurista y caótico. Estos primeros diseños se centraron en crear una figura que fuera tanto funcional como amigable, capaz de conectarse con los jugadores con su diseño. Se realizaron múltiples iteraciones, ajustando detalles como sus extremidades, sus herramientas de limpieza, hasta lograr un diseño que reflejara su propósito.

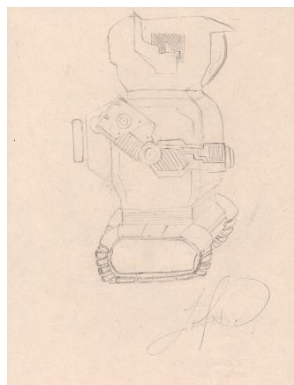


Figura 2-7 Boceto inicial de Azoni.

2.6.2. ENEMIGOS

Paralelamente, se diseñaron los enemigos del juego: los mutantes de basura. Estos villanos representan residuos contaminados que han adquirido formas y comportamientos agresivos debido a la contaminación extrema. Cada enemigo se conceptualizó con diferentes materiales desechados, como plásticos, cartones y otros, para ilustrar la variedad y peligrosidad de la contaminación. Los bocetos de estos enemigos muestran características grotescas y deformadas, destacando el contraste entre la pureza de Azoni y la corrupción de la basura mutada.

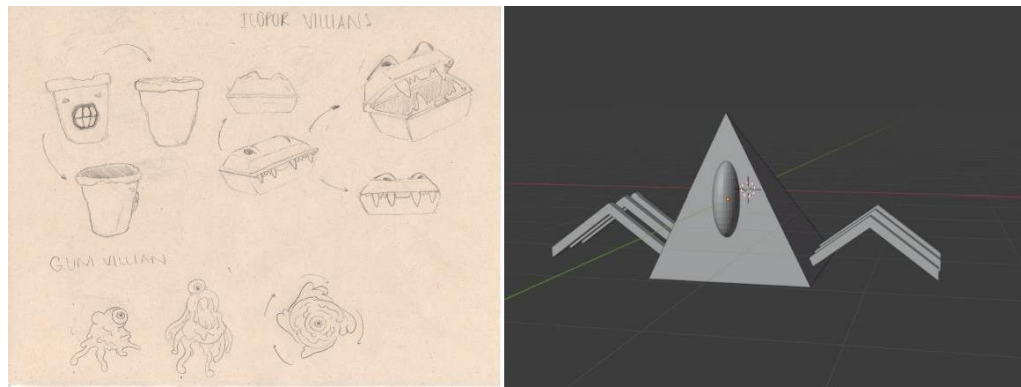


Figura 2-8 Ideas iniciales de enemigos **Figura 2-9** Boceto prueba enemigo

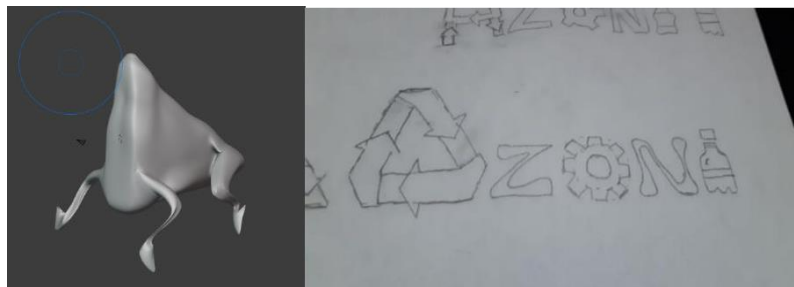


Figura 2-10 Boceto prueba enemigo 2 **Figura 2-11** Boceto ideas logotipo Azoni

2.6.1. RESIDUOS

Para asegurarnos de que los residuos en el juego fueran representados de manera realista y educativa, tomamos referencias de desechos comunes y sus efectos en el medio ambiente. Investigamos cómo ciertos materiales, cuando no son gestionados adecuadamente, pueden causar graves daños ecológicos. Esta investigación nos permitió sensibilizar a los jugadores sobre la importancia del reciclaje y la correcta disposición de residuos.



Figura 2-12 Ejemplo grafico de botella de agua. Tomado de *Agua mineral* [Fotografía] Núñez, A. (2023) [6]

Figura 2-13 Ejemplo grafico de botella de tabla con puntillas. Tomado de *Uña oxidada en madera de raja* [Fotografía] Pongsak14 (2016) [7]

Figura 2-14 Ejemplo grafico de botella de manzanas en mal estado. Tomado de *Una manzana podrida* [Fotografía] López, A. (2014) [8]

En conjunto, los bocetos de Azoni, los enemigos mutantes de basura y las referencias de residuos crean un universo visualmente coherente y temáticamente significativo, enriqueciendo la narrativa y la experiencia de juego.

2.7.MODELOS

Gracias a los bocetos realizados, podemos crear los modelos tridimensionales de Azoni, los enemigos mutantes y los residuos, Figura 2-15. Estos modelos permitirán dar vida a nuestro robot protagonista en un entorno interactivo y visualmente atractivo.

Del mismo modo, los enemigos mutantes de basura serán modelados a partir de los bocetos conceptuales. Esto nos permitirá representar de manera efectiva las grotescas deformaciones y características únicas de cada tipo de residuo contaminado.

Además, las referencias tomadas para los residuos nos ayudarán a desarrollar modelos realistas y variados de basura. Cada modelo de residuo estará diseñado para reflejar con precisión los diferentes materiales.



Figura 2-15 Ejemplo con modelo de enemigo sin texturizar

2.7.1. TEXTURAS

Una vez realizados los modelos 3D, es hora de añadir color y profundidad mediante la aplicación de Blender que nos da la paleta de pintura para que nosotros podamos pintar a mano los diferentes modelos, Figura 2-16 y Figura 2-17.

Para Azoni, nuestro robot protagonista, las texturas se enfocarán en resaltar su aspecto futurista y amigable.

Los enemigos mutantes de basura, por otro lado, requerirán texturas que reflejen su naturaleza corrupta y contaminada. Utilizaremos colores oscuros. Las texturas detalladas mostrarán superficies rugosas y deformaciones, enfatizando su origen de residuos mutados y su peligrosidad.

En cuanto a los residuos, las texturas serán fundamentales para diferenciar los distintos tipos de materiales. Usaremos referencias del mundo real para crear texturas que representen fielmente plásticos, cartones, productos químicos y otros desechos.

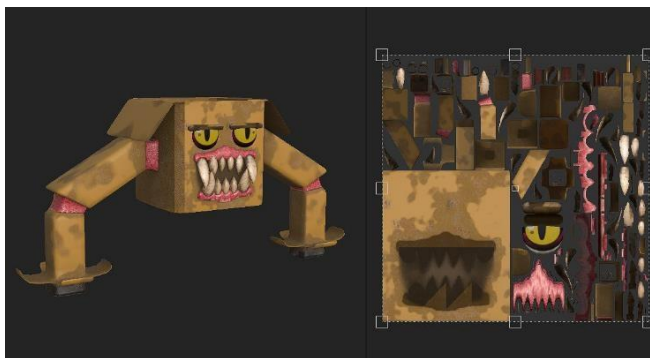


Figura 2-16 Ejemplo enemigo luego de crear su textura e implementársela al modelo

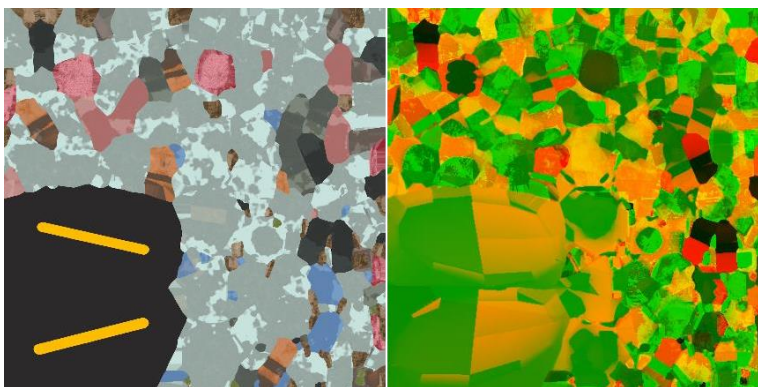


Figura 2-17 Ejemplo de texturas y brillos

2.8. ANIMACIÓN DE PERSONAJES

A cada personaje se le proporcionó una animación que fue creada moviendo el modelo parte por parte (frame por frame), Figura 2-18, donde se crean las animaciones de correr de los enemigos el movimiento de Azoni, las animaciones necesarias para los diferentes objetos que sean seleccionados.

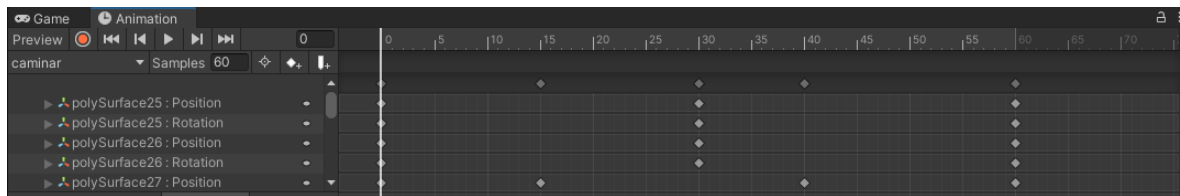


Figura 2-18 Tabla de edición de animación

Este apartado de animación nos sirve para darle vida a los personajes, objetos y cualquier tipo de elemento disponible en la jerarquía de unity, Figura 2-19.

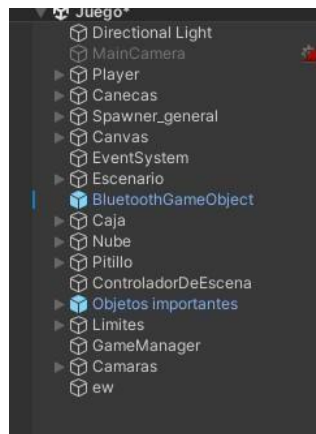


Figura 2-19 Jerarquía

Para el control de estas animaciones, Figura 2-20, tenemos el Animator o animador, con este podemos crear “mapas” y así añadir varias animaciones a un objeto.

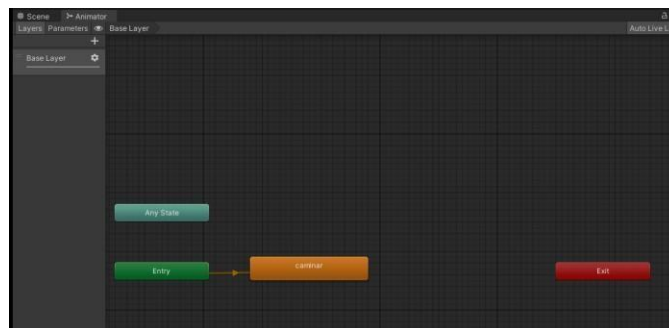


Figura 2-20 Animator – creador de mapas de animación

Para esto usamos los parámetros, Figura 2-21, estos junto a las flechas que vimos en los mapas dan vía o un camino a las animaciones, como es, ejemplo: de estado idle o inicial a luego caminar; este vemos dos estados idle y caminar entonces en la parte de layer pondremos idle (se puede poner cualquier nombre).

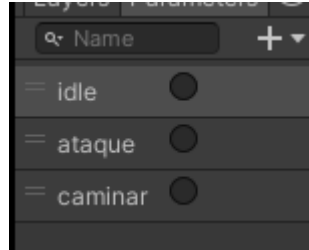


Figura 2-21 Parámetros de animación

y en el inspector, Figura 2-22, le asignamos su respectivo parámetro para que ya lo podamos activar desde el código.

`boxAnimator.SetTrigger("idle");`

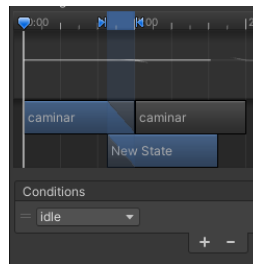


Figura 2-22 Inspector - apartado animación

2.9.ESCENARIO

Podemos observar como el escenario ha avanzado pasando de calidad de iluminación y también diseño de escenario agregándole más vida a este también podremos observar los diferentes elementos en el mapa esparcidos para que el jugador se motive a explorar las diferentes secciones de este escenario, Figura 2-23:28.

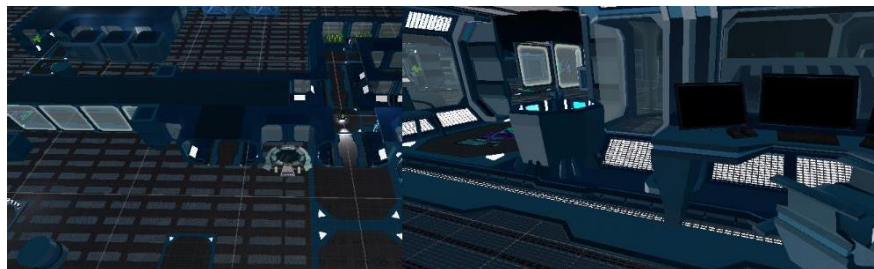


Figura 2-23 y Figura 2-24

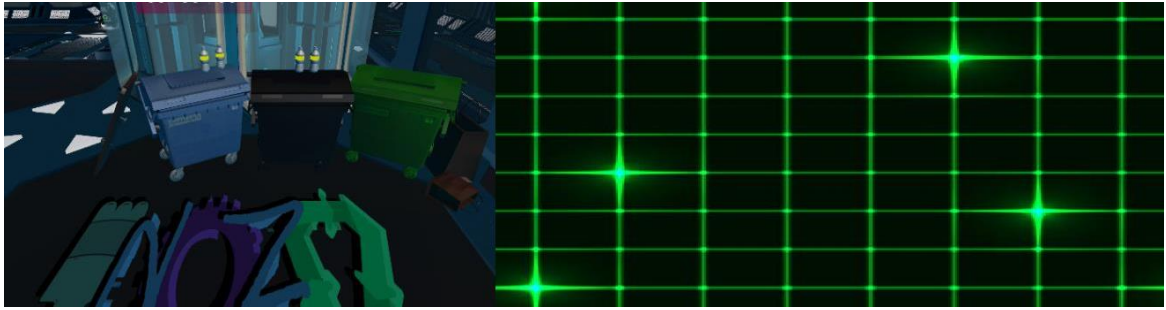


Figura 2-25 y Figura 2-26

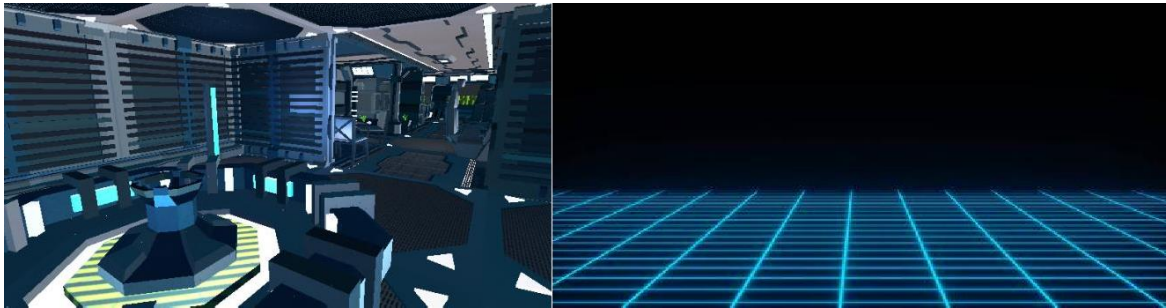


Figura 2-27 y Figura 2-28

Muestras de escenario, se observan pasillos, fondos, iluminación, residuos y elementos como sillas y computadores que se diseñaron con el propósito de enriquecer el escenario.

Video de muestra: En el siguiente video se observara un poco mas a fondo el escenario del nivel para poderlo ver con mas detalle. Tomado de *Azoni AR Escenario* [Video] Azoni AR (2024) <https://youtu.be/pqQo6wDjGAQ> [9]

En el videojuego se busca que el jugador explore, elimine y obtenga residuos, por eso se hace un mapa que pone al jugador en un semi laberinto, buscando que la persona busque cómo puede crear una estrategia para atrapar enemigos antes de que huyan.

Para el mapa tenemos un diseño estilo oficinas y laboratorio, Figura 2-29, con un fondo estilo Matrix, el cual hace pensar al usuario que está en un mundo virtual y que cuando tenga que ir al apartado AR este encienda la cámara y pueda interactuar con el robot físico desde el videojuego.

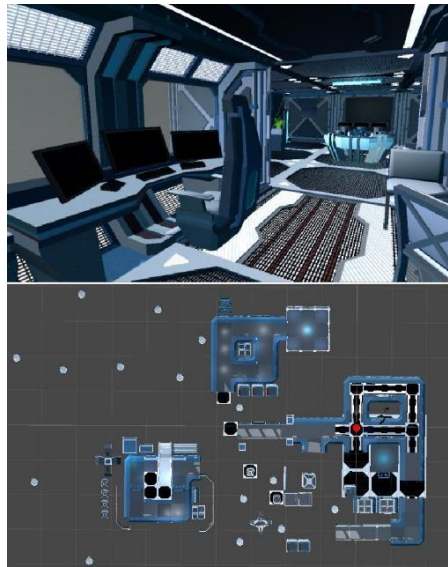


Figura 2-29 Muestra de mapa y escenario al mismo tiempo

En esta parte, Figura 2-30, se ve el escenario donde se consideró el diseño futurístico de este, un control de luz adecuado para que el jugador se sienta cómodo y una decoración interna que apoya la parte estética del escenario, donde se pueden ver sillas mesas muebles equipos tecnológicos como computadores, también se ven plantas y objetos diseñados para acompañar el escenario y este se vea más completo.

Podemos observar que estos fueron diseñados a detalle para el mayor disfrute del juego

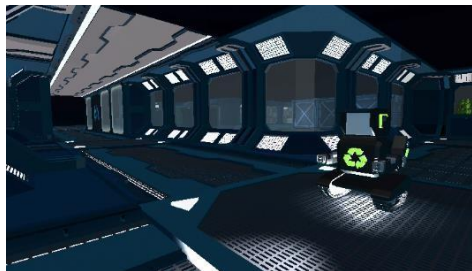


Figura 2-30 Escenario desde una nueva perspectiva, pero junto al protagonista Azoni

2.10. INTERFAZ

En nuestra interfaz, Figura 2-31, observamos cómo ha evolucionado, destacándose por sus colores característicos del estilo ciberpunk, con tonos neón. También hemos puesto especial énfasis en la ubicación de cada botón, logrando así una experiencia cómoda para el jugador.

Contamos con dos cámaras, una que nos ayuda con la visualización del mapa y otra que proporciona una nueva perspectiva, apoyando la exploración del jugador. En la parte superior, se encuentra el botón de menú, desde el cual se puede acceder a diversas configuraciones, como el control de

iluminación y el control de audio, donde se pueden disfrutar de las canciones 100% originales producidas por nuestro equipo.

Para esta sección podremos escuchar las canciones creadas por el equipo en donde podremos ver que estos se crearon con el objetivo de acompañar al jugador durante su aventura mediante sonidos suaves juguetones, indicios de aventura y reto.

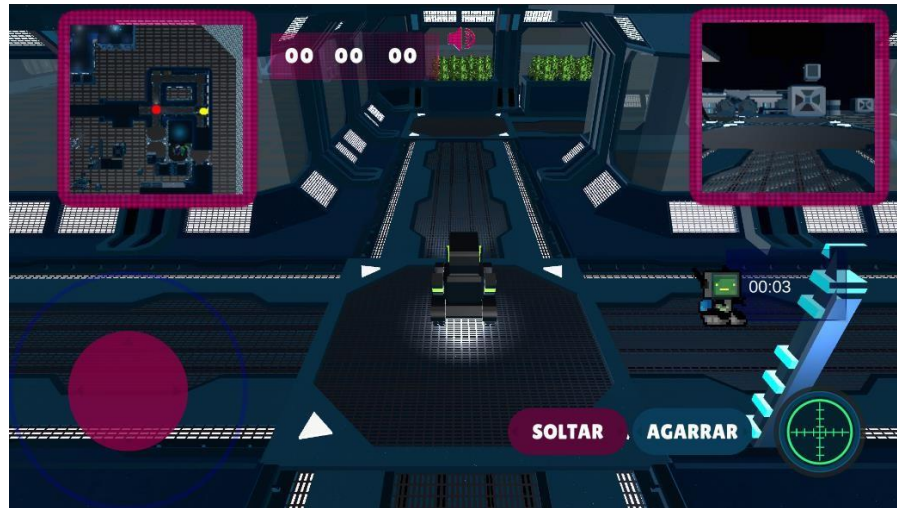


Figura 2-31 Interfaz de nivel virtual

Menú de opciones: Figura 2-32, Aquí podremos visualizar, un controlador de brillo para que el usuario se sienta cómodo al momento de jugar y pueda cuadrarlo a su gusto, el variador de volumen para la música del videojuego y un seleccionador de las canciones disponibles echas por el equipo.



Figura 2-32 Menú de opciones

Canción original número 1. Tomada de *Azoni AR – Canción 1* [Música] Azoni AR (2024) [10]
<https://youtu.be/3QxvzozPgmw>

Canción original número 2. Tomada de *Azoni AR – Canción 2* [Música] Azoni AR (2024) [10]
<https://youtu.be/NK7KISpgrYU>

Canción original número 3. Tomada de *Azoni AR – Canción 3* [Música] Azoni AR (2024) [10]
<https://youtu.be/0Gqyw0o0Q88>

2.11. ILUMINACIÓN

- ✓ Fuentes de Luz:
 - Luz Ambiental: Es la luz que llena todo el entorno, proporcionando una iluminación base sin sombras.
 - Luz Puntual: Representa una fuente de luz que emite luz en todas direcciones desde un punto específico.
 - Luz Direccional: Simula la luz del sol y emite luz en una dirección específica.
 - Luz Spot: Emite luz en una dirección específica, similar a un foco.
- ✓ Sombreado:
 - Sombreado Difuso: Determina la cantidad de luz que llega a una superficie y se dispersa en todas direcciones.
 - Sombreado Especular: Controla los reflejos especulares, que son los destellos de luz en superficies brillantes.
- ✓ Modelo de Reflectancia de Lambert:
 - Describe cómo la luz se refleja en una superficie mate.

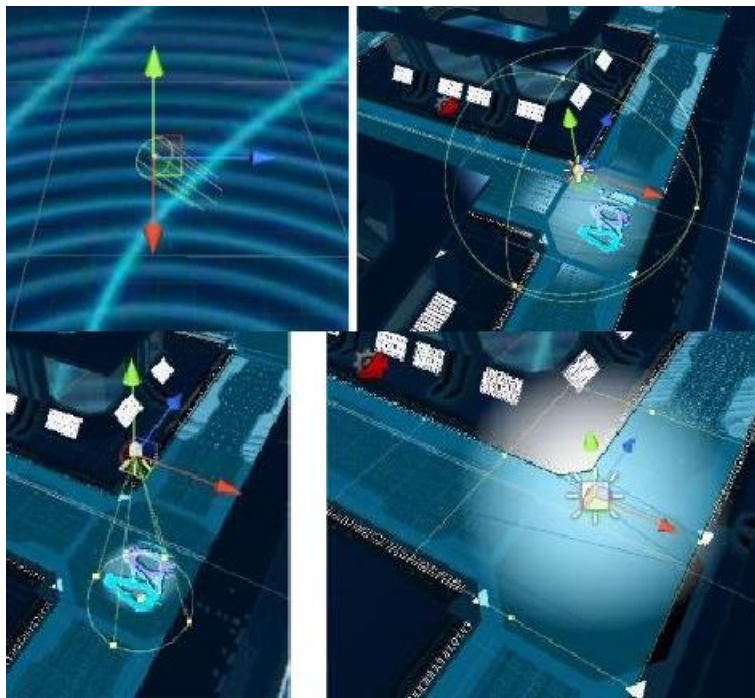


Figura 2-33 Tipos de fuente de iluminación

- ✓ Tipos de Iluminación:
 - Iluminación en Tiempo Real: Calcula la iluminación en cada cuadro, adecuada para escenarios dinámicos.
 - Iluminación Baked (recalculada): Calcula la iluminación de antemano y la almacena, mejor para entornos estáticos.
- ✓ Son Sistemas de Iluminación en Unity:
 - Unity Global Illumination (GI): Permite el cálculo de la iluminación global en tiempo real.
 - Realtime Global Illumination: Iluminación global en tiempo real basada en la posición y el color de las luces.

Iluminación sobre Objetos en Unity:

- ✓ Materiales y Shaders:
 - Propiedades de Material: Los materiales en Unity tienen propiedades que afectan cómo reaccionan a la luz.
 - Shaders: Programas que determinan cómo se renderizan los objetos, incluyendo la iluminación.
- ✓ Luces en Unity:
 - Configuración de Luces: Puedes ajustar la intensidad, color y otros parámetros de las luces en Unity.
 - Sombras: Las luces pueden arrojar sombras, y puedes personalizar su calidad y resolución.
- ✓ Baking (Cocción) de Luces:
 - Iluminación Baked: Unity permite cocinar la iluminación para mejorar el rendimiento en tiempo real.
 - Lightmapping: Proceso de precalcular cómo la luz afecta a la escena y almacenar esta información para su uso posterior.
- ✓ Post-Processing:
 - Efectos Post-Procesamiento: Unity admite efectos visuales después de la renderización, como desenfoque, tonemapping y más.

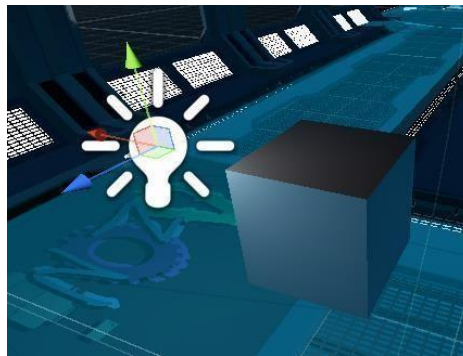


Figura 2-34 Procesamiento de luz en superficies u objetos

2.11.1. RECURSOS ADICIONALES:

- Libros y cursos específicos sobre iluminación en Unity.
- Documentación oficial de Unity:
- Iluminación – Unity. Tomada de UNITY DOCUMENTATION [Pagina Web] Unity (2024) [11]

2.12. NAVMESH

El análisis de NavMesh es crucial para establecer áreas de navegación dentro del entorno del juego, lo que permite que los personajes controlados por la inteligencia artificial (IA) se muevan de manera coherente y sin problemas a lo largo del terreno. Esta funcionalidad garantiza una interacción fluida entre los personajes y su entorno, contribuyendo así a la inmersión del jugador en el mundo del juego.

Cuando se ha tenido la necesidad de que algún personaje del videojuego llegue a un lugar mediante diferentes caminos, ya sea el camino más largo o el camino más corto o también se podría tener el caso de que tenemos la necesidad que un personaje solo se mueva en un espacio asignado como un patrullaje, para estas necesidades se usa la herramienta de NavMesh o Sistema de navegación de Unity.

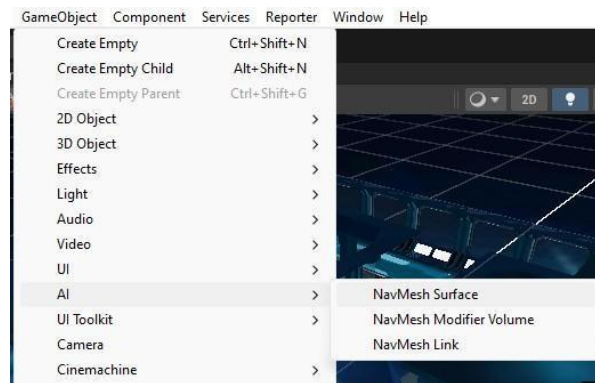


Figura 2-35 Búsqueda de NavMesh

Para este tipo de navegación necesitada en el proyecto se usa la navegación de superficie (NavMesh Surface)

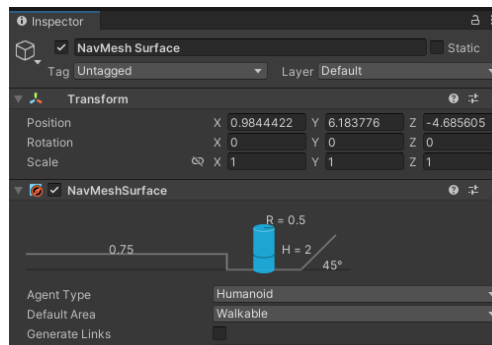


Figura 2-36 Inspector – Sección NavMesh

En esta opción, Figura 2-37, podremos observar que viene para un agente tipo humanoide el cual viene por defecto, podremos observar el radio del personaje, su altura, el salto o escalón que puede subir, y la inclinación que de igual manera podrá escalar. Estos valores podremos cambiarlos según la necesidad del momento.

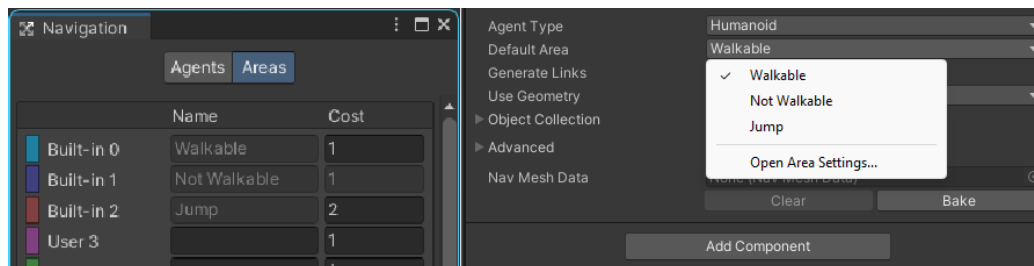


Figura 2-37 Inspector – Configuración de áreas

También podremos encontrar la configuración de áreas, estas nos servirán para definir donde puede y no caminar el personaje que se le ha asignado la navegación.

Como vemos tenemos, Figura 2-38; Walkable (Caminable), Not Walkable (No caminable) y Jump (Salto). Estas son las opciones por defecto, pero nosotros podremos agregar más configuraciones de áreas en el apartado de navegación.

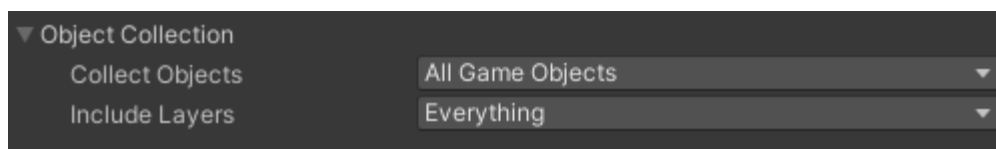


Figura 2-38 Inspector – Configuración de áreas

Ahora tenemos la colección objetos donde se eligen los diferentes puntos disponibles para la navegación, que en este caso sería nuestro mapa, Figura 2-39, aunque solo se elegirá el suelo y objetos que estén en el mapa para que este los tome en cuenta al momento de desplazarse.

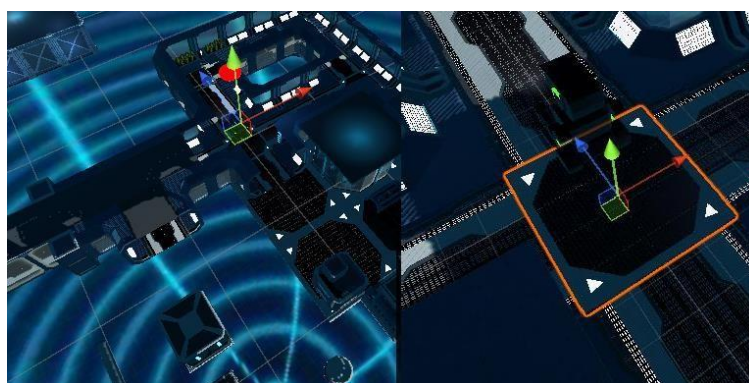


Figura 2-39 Escenario seleccionando una de las plataformas preparadas para el NavMesh

Luego de seleccionar todos los suelos y objetos daremos uso a la opción de “Bake” o cocinar, Figura 2-40, lo que hará que nuestra superficie se torne de un color de azul claro y nos lo habilite para la navegación en esta.

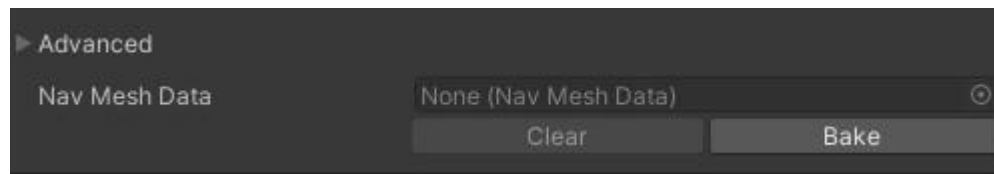
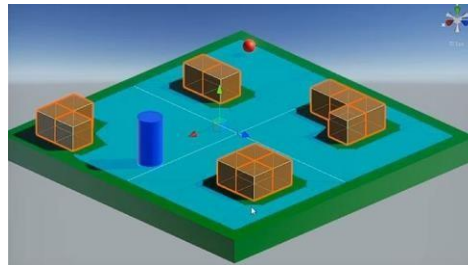


Figura 2-40 Inspector – botón de cocinado “baked”



Navemesh cocinado “baked”. Tomado de NavMesh Primeros pasos [Imagen referencia] - GamerGarage (2024) [12] <https://youtu.be/G6OTzfcyNc>

En nuestro caso quedaría así Figura 2-41.

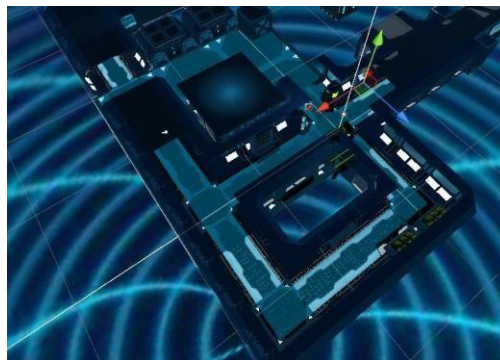


Figura 2-41 Superficie de mapa con sistema de NavMesh implementado

dándonos en este caso un rango dependiendo al que elijamos, por defecto viene con 0.5 de radio respecto al centro del personaje Figura 2-42.

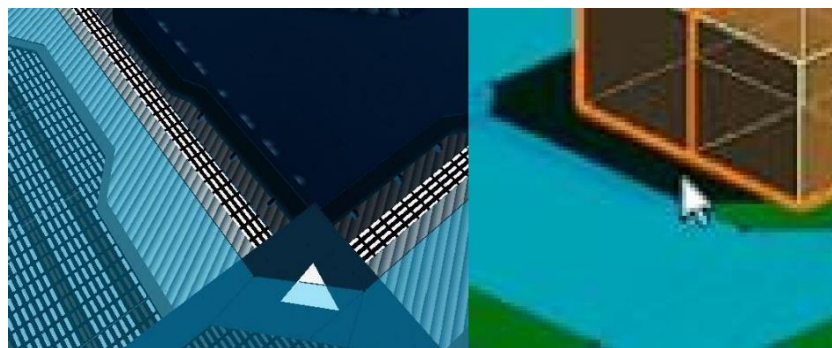


Figura 2-42 Configuración de Nav Mesh – radio de interacción con objetos [12]

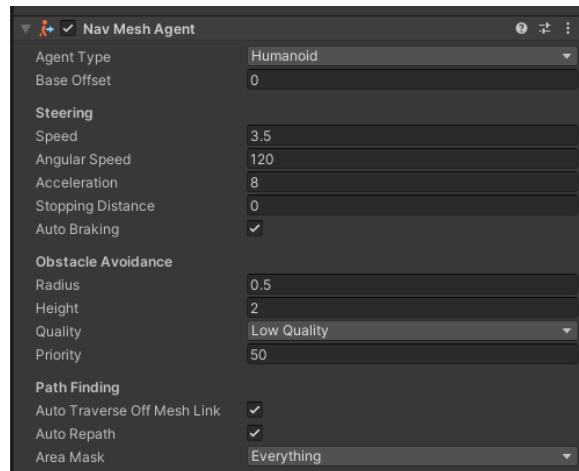


Figura 2-43 Configuración de Nav Mesh

Luego de que nosotros tengamos listo nuestro mapa o sistema añadiremos el agente de navegación al inspector de nuestro personaje Figura 2-43.

Aquí podremos configurar:

- Aceleración
- Velocidad angular
- Velocidad
- El radio respecto al objeto
- La calidad del personaje al encontrar la mejor o peor ruta
- Auto breaking para que pueda pararse
- Auto Travers y repath que nos permiten encontrar nuevos caminos
- Area Mask la cual es para las áreas antes definidas

Ya luego con esto configurado se procede a crear su respectivo script o código que nos permita usar nuestro personaje deseado, donde ya pondremos la lógica de los enemigos mencionada anteriormente.

2.12.1. LOGICA Y PROGRAMACION

2.12.2. LÓGICA ENTRE ESCENAS

Mantener objetos o cualidades

```
var noDestruirEntreEscenas =
FindObjectsOfType<EntreEscenas>()); if
(noDestruirEntreEscenas.Length > 1)
{
    Destroy(game
    Object);
    return;
```

```
}
```

```
DontDestroyOnLoad(gameObject);
```

Este código nos permite filtrar y categorizar los objetos y variables que queremos conservar a lo largo de las escenas implementadas en el juego, este código realiza la selección de los elementos aplicando el criterio de si estos se encuentran dentro del objeto Entre del game object que le asignemos, este código está pensando para el menú de opciones, la puntuación de los residuos y podría ser utilizado para aspectos como música brillo perspectivas, etc.

2.12.3. REALIZAR LA TRANSICIÓN ENTRE ESCENAS

En este apartado contamos con el siguiente código

```
public enum Levels { BluetoothScene = 0, Juego = 1, Opciones = 2,
Menuinicial = 3
}

public void CambiarNivel(Levels level)

{
SceneManager.LoadScene((int)level);

}
```

Este nos permite asignarle un valor fijo a la escena o canvas, Figura 2-44, al que queremos que el jugador acceda después de cierta acción, este código es fácil de realizar y configurar, ya que permite agregar cuantos valores deseemos por ende no está limitado a una cierta cantidad de escenas, y he de aclarar estos valores que asignamos a las escenas deben ser iguales al orden que están organizadas en la opción build settings.

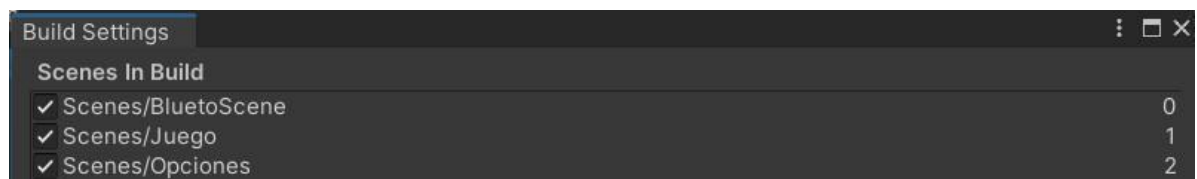


Figura 2-44 Activación de escenas y asignación de orden

2.12.4. SISTEMA DE PUNTOS

Game Manager(Admin)

```
public void ResiduoA()
{
```

```

        Residuo1 = Residuo1 + 1;
        Residuo1Text.text =
        Residuo1.ToString();
    }
    public void ResiduoB()
    {
        Residuo2 = Residuo2 + 1;
        Residuo2Text.text =
        Residuo2.ToString();
    }
    public void ResiduoC()
    {
        Residuo3 = Residuo3 + 1;

        Residuo3Text.text = Residuo3.ToString();
    }

```

Este código nos permite realizar toda la lógica de puntuación durante la escena asociada o pertinente que requiramos, en este se encuentra todo el proceso que es necesario para la correcta ejecución del sistema en un solo código, ya que en este se encuentran los 3 TMP_Text los cuales son los encargados de reflejar el valor correcto de cada uno de los residuos recogidos por el jugador, y los tres valores como tal de la puntuación de estos residuos, en estos en donde se ven reflejadas las respectivas sumas y por medio de la conversión de este valor a una variable de tipo string es reflejado en los textos TMP, esto viene acompañado de unos subcódigos asignados en cada uno de los “objetivos”, este game manager es quien se encarga de actualizar el puntaje cada vez que el jugador recicle unos de los residuos.

Cada uno de los residuos cuentan con un código el cual permite llamar o realizar el proceso que se encuentra descrito en el código implementado en el game manager.

```

if (other.CompareTag("Recolector"))
{

    FindAnyObjectByType<GameManager>().ResiduoA();

    Destroy(gameObject);
}

```

Donde cada uno de estos códigos compara los diferentes collider y así poder realizar el debido proceso de puntuación dependiendo del objeto busca el void que este asignado en el código de cada objetivo en el game Manager y así poder tener el correcto seguimiento.

3. IMPLEMENTACION DE LA REALIDAD AUMENTADA

3.1. PRIMERAS PRACTICAS

Con el objetivo de tener más claro cuáles fueron los errores cometidos se describirán las primeras practicas realizadas para la detección de símbolos que forma parte de la realidad aumentada. Lo primero debe ser enseñar las imágenes con las que se hicieron las primeras pruebas, la cuales fueron generar objetos 3D sobre las imágenes una vez identificadas, las imágenes usadas fueron las siguientes:



Figura 3-1 Personaje animado con alto nivel de detalle. Imagen tomada de *wonder egg priority* [Ilustración] Nakayama, N. (2021) [13]

Figura 3-2 Universidad Piloto de Colombia. Imagen tomada de *unipiloto* [Logotipo] Universidad Piloto de Colombia (2024) [14]

Figura 3-3 Universidad Piloto de Colombia. Imagen tomada de *unipiloto* [Logotipo] Universidad Piloto de Colombia (2024) [14]

Más adelante se probó hacer uso de los multi targets opción que brinda vuforia que no es otra cosa varias imágenes juntas para formar un cubo u objeto de 6 caras tridimensional, esta opción se consideró cuando veíamos problemas con los targets actuales, pensando que el reconocimiento sería mejor, para ello se debía tener también el objeto tridimensional en la vida real por lo cual se recurrió a una imagen que permite armar “muñecos” con doblar y pegar.



Figura 3-4 Personaje animado con la capacidad de hacer modelo 3D. Imagen tomada de *BMO adventure* [Ilustración] Gilian, N. (2024) [15]

La imagen del personaje “BMO”, Figura 3-4, permitió crear una figura tridimensional fácilmente lo siguiente fue recortar la imagen e ingresarla en la base de datos que brinda Vuforia con la licencia gratuita para esto lo que debemos hacer es recortar cada una de las caras, conocer su medida en metros e ingresar esta información en la cara respectiva en base de datos.

Sin embargo, los targets presentaban el mismo problema, después de ser identificadas la primera vez, es decir se generaba su respectivo modelo recientemente abierta la aplicación, los modelos tendían a moverse de su posición inicial y al enfocar el target el modelo tenía una posición errona, flotando en una posición ilógicas o incluso fuera de la pantalla.

Con el objetivo de optimizar este proceso se indago respecto a que tipo de imágenes son las mejores para poder usar, lo primero fue leer el documento de Vuforia con esta identificamos el sistema que utiliza para identificar las imágenes, el dispositivo pasa la imagen en escala de grises, práctica común en el procesamiento de imágenes, después de eso pasa a identificar pixeles claves, son los pixeles que contienen caracterizas destacables de la imagen lo que facilita su identificación, es decir no vamos a buscar toda la imagen, vamos a buscar estas características con el fin de que el proceso sea rápido por ultimo teniendo en cuenta todos los pixeles podemos saber la posición de la imagen y su rotación.



Figura 3-5 Imagen ejemplo de Vuforia para entender que es una imagen con alto nivel de detalle y buenos pixeles claves. Obtenido de *Best practices for designing and Developing Image-Based Targets* [Pagina Web] Vuforia Engine (2024) [16]

3.2. ESCALA DE GRISES

Para entender el concepto de escalas de grises primero debemos de entender que una imagen en una pantalla no es otra cosa que un grupo de x cuadros de colores agrupados en filas y columna, cuantos más cuadros más clara es la imagen, y estos cuadros se denominan pixeles, los pixeles se pueden traducir a un número o arreglo de números, dependiendo del número de bits del dispositivo, frecuentemente se usan 8 bits es decir 0 a 255, 255 implica alta frecuencia de un color y 0 nula presencia, y para dar color del pixel se pueden sacar canales que describen la presencia de colores principales, frecuentemente se usa RGB, el pixel tiene tres números de 8 bits, uno para el rojo o RED, verde o el GREEN y azul o BLUE, Figura 3-6, gracias a combinar estos canales tenemos infinidad de colores.

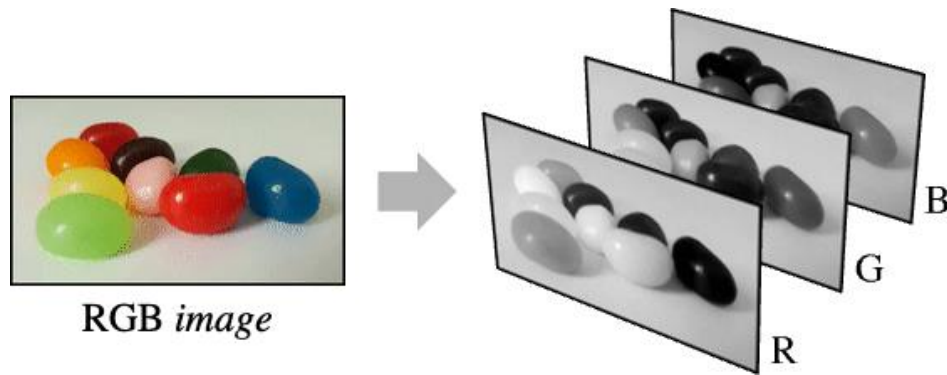


Figura 3-6 Imagen demostrativa de como funciona el RGB. Imagen tomada de *Image channels* [Ilustración] Wolfram (2024) [17]

La escala de grises se hace para ahorra recursos ya que es más rápido analizar un grupo de pixeles de un solo canal a un grupo de más canales, lo que ocurre con los canales es que se convierten en uno solo, un valor de 0 a 255 por ejemplo, donde 0 represente el negro puro y el 255 el blanco por lo tanto lo que queda en medio son una gran cantidad de grises, la mayoría indistinguibles para el ojo humano, pero no para la máquina.

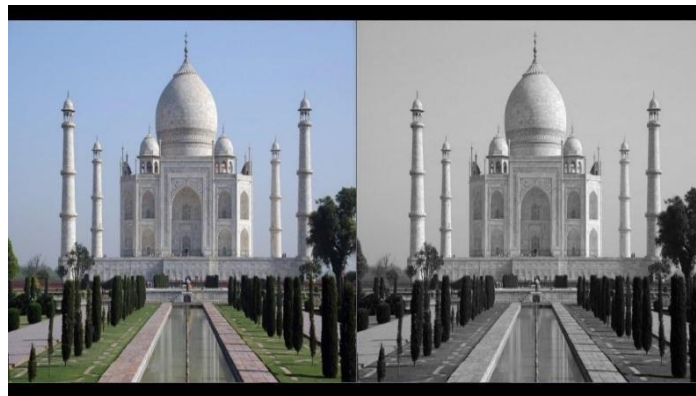


Figura 3-7 Imagen de ejemplo en su estado de color y escala de grises. Imagen tomada de Taj Mahal [Fotografía] Wikimedia (2024) [18]

3.3. HISTOGRAMA

El histograma no es una gráfica de nivel de gris contra cantidad, Figura 3-8, usada en imágenes para evaluar los diferentes canales, lo que hace el histograma es leer los bits de cada uno de los pixeles de la imagen e ir contando cuantos pixeles tiene el mismo valor y con eso generar una sola tabla que permite fácilmente ver que los colores más frecuentes en la imagen.

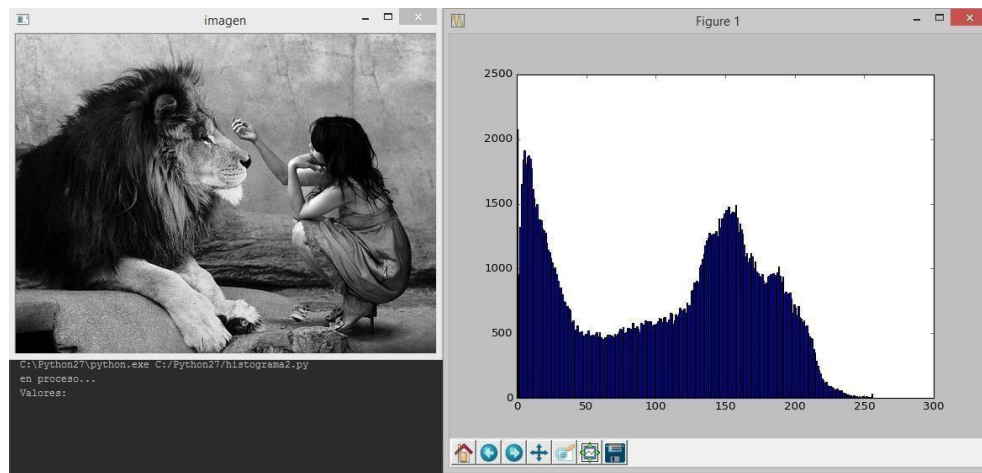


Figura 3-8 Histograma – de imagen a cantidad de pixeles

Véase como el histograma tiene más picos en la parte izquierda es decir la parte donde está el 0 por lo tanto el histograma nos permite saber que hay fuerte presencia de colores oscuros en la imagen y también hay una cantidad considerable de gris con un valor de 150.

3.4. PÍXELES CLAVES

Como se dijo anteriormente los pixeles claves son los puntos que contienen características útiles para identificar el target, la manera en que define cuales son los pixeles con información suficiente es muy sencillo lo que hace es revisar los vecinos del píxel, para explicar mejor esto véase la siguiente imagen que es una imagen de 3x3 pixeles Figura 3-9.



Figura 3-9 Pixeles y su conversión a grises

Si el pixel del medio, en la posición 2x2, es el candidato para ser clave se debe mirar que tan distinguible es respecto a sus vecino y a su vez, en el grupo, que tan diferentes son los vecinos, por ejemplo en la primera imagen en escala de grises vemos que respecto al centro no hay una gran variación en el tono en consecuencia ese segmento puede ser fácilmente confundido con otro, en cambio veamos el de su derecha, el pixel del medio tiene un tono distinguible de sus vecinos, es decir este puede ser fácilmente reconocido en la imagen grande, por lo tanto puede ser un pixel clave.

Lo que hace Vuforia y otros programas de reconocimiento de imágenes es recurrir a estos píxeles con características distinguibles, en nuestro caso lo primero que hace es recorrer la imagen y decidir que píxeles puede usar es decir se define las características o píxeles claves, cuando están elegidos son almacenados, y se buscan como un patrón, pienses como ver una constelación en el cielo con el fin de identificar los puntos cardinales.

En las imágenes se busca el patrón y además de identificar la coincidencia es capaz de entender la rotación y posición del target. cabe aclarar que un target con varios puntos claves será más fácil de reconocer y mantener en posición su respectivo objeto en cambio imágenes con pocos píxeles claves dará error ya sea a la hora de identificar la imagen en diferentes ángulos y posiblemente confundir con otros elementos lo que resulta en interacciones erróneas



Figura 3-10 Búsqueda de píxeles clave en dos diferentes imágenes [14] [16]

3.5. MEJORES TARGETS

Ahora entra la duda de cómo puedo tener un buen target, ya que como se dijo anteriormente los píxeles claves no se pueden elegir directamente por el usuario, pero podemos tener en cuenta cuales son las características de estos píxeles y así elegir las imágenes.

En primer lugar, miremos las recomendaciones que da el mismo Vuforia.

Atributo	Ejemplo
Rico en detalles	Escenas callejeras, grupos de personas, collages y mezclas de elementos y escenas deportivas son buenos ejemplos.
Buen contraste	Las imágenes con regiones brillantes y oscuras y áreas bien iluminadas funcionan bien.
Sin patrones repetitivos	Emplee características únicas y gráficos distintos que cubran la mayor parte posible del objetivo para evitar la simetría, patrones repetidos y áreas sin características.
Formato	Debe ser de 8 o 24 bits PNG y JPEG formatos; menos de 2 MB de tamaño; JPG debe ser RGB o escala de grises (no CMYK).

Figura 3-11 Recomendaciones de Vuforia para la selección de imágenes. Imagen Tomada de *Pose status and Status info in Unity* [Tabla-Imagen] Vuforia Engine (2024) [19]

El primer parámetro son los detalles, afirma que una imagen con una variedad considerable de elementos y detalles distinguibles puede proporcionar facilitar la identificación del target, como primera recomendación para esto, se recomienda el uso no solo varios colores sino también tener una buena distinción entre ellos, una forma para poder evaluar la cantidad de color de una forma más teórica es usando el histograma de la escala de grises.

Vuforia recomienda que “el histograma de la imagen es estrecho y puntiagudo, no es probable que sea una buena imagen de destino” en contra parte de este párrafo expresa más adelante dice “si el histograma es ancho y plano, es una buena primera indicación de que la imagen contiene buena distribución de características útiles”, esto último indica no solo que la escala de grises es un factor para considerar, sino que en cambio es el primer indicativo sobre la utilidad de la imagen, a continuación se muestra un ejemplo de imágenes con su respectivo histograma, Figura 3-12.

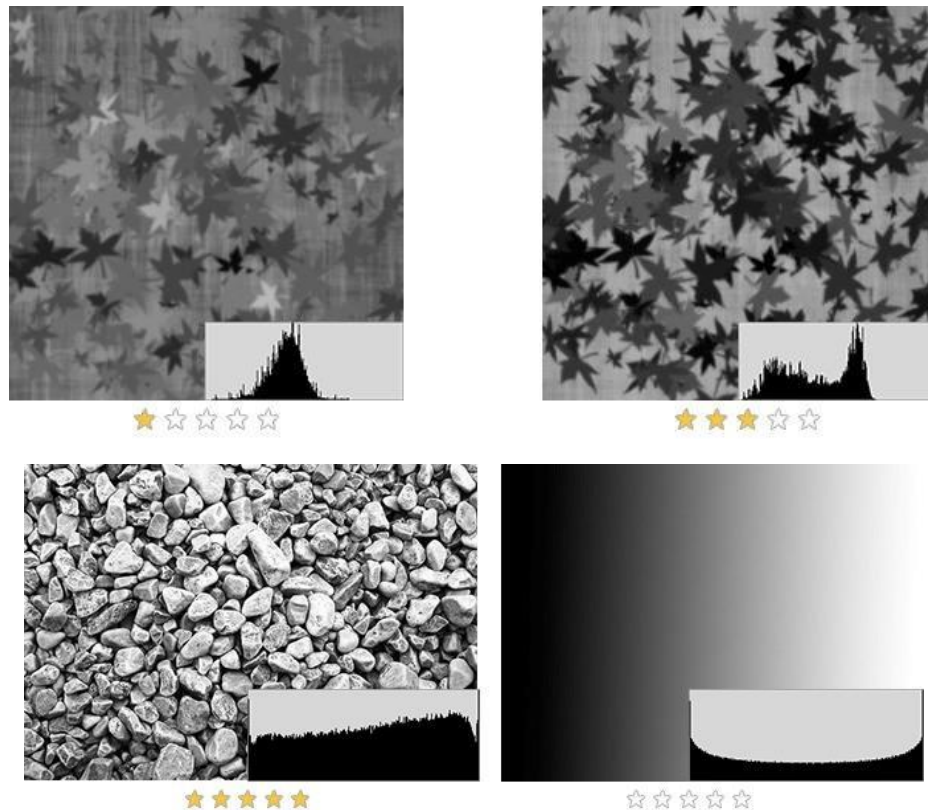


Figura 3-12 Clasificación de imágenes con Vuforia. Imagen tomada de *Image Targets Optimization Techniques* [Fotografía] Vuforia Engine (2024) [20]

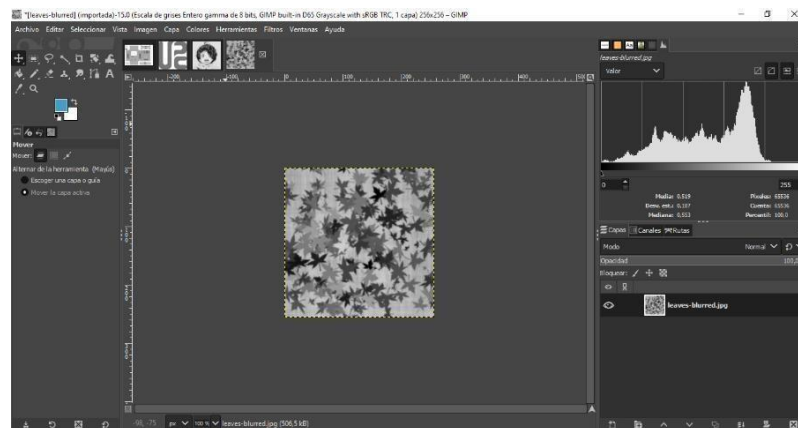
Para entender mejor que imágenes son óptimas, retomemos lo que dice la documentación un histograma ancho y plano es el mejor, en primer lugar cuando se habla de ancho se habla

de una gran cantidad de colores entre blanco y negro presentes en la imagen y cuando se habla de plano hablamos de equilibrio de color, el alto en el histograma se define por cantidad de pixeles por lo tanto para obtener un histograma plano debe haber una cantidad casi igual de pixeles por color, con esto claro vimos que tal eran los histogramas de nuestra primeras prácticas para ellos recurrimos al programa de GIMP y este fue el resultado:



Gracias a poder ver los histogramas definimos que las imágenes de las primeras practicas no son nada útiles para este propósito, en la parte superior derecha vemos el histograma, el cual se comporta de solo un par de líneas visibles ahora comparece con un histograma de una imagen saca de la documentación de Vuforia

Nota: la imagen encontrada y que se mostrará a continuación se describe como un 3 de 5 estrellas, es decir, no es la más conveniente para lo que se quiere hacer, pero puede usarse con errores.



El histograma, con un comportamiento destacable, hay variedad en la gráfica, pero no se busca, ya que el objetivo es histograma grueso y plano. Nuestra pauta es que el color tenga una cantidad considerable de color.

Regresando a la recomendación de Vuforia, donde recomienda fotos de calles, varias personas y varios elementos podemos afirmar que son imágenes con varios colores y varios elementos sin embargo para poder aplicar debidamente esta recomendación debemos conocer el contraste antes.

Continuando con las recomendaciones de Vuforia encontramos el contraste, esto hace referencia a la diferencia entre la intensidad de los colores, en cierto modo que tan distinguibles son los detalles u objetos que componen la imagen para entender mejor esto recuerde los pixeles claves estos buscan lugares con diferencias únicas y gracias a tener un buen contraste podemos resaltar detalles y obtener más pixeles claves.



Figura 3-15 Clasificación de imágenes con Vuforia. Imagen tomada de *Image Targets Optimization Techniques* [Fotografía] Vuforia Engine (2024) [20]

Por ejemplo, véase la imagen, Figura 3-15, en ella vemos que la primera imagen el tono del fondo se confunde con las hojas haciendo más difícil encontrar detalles para ser pixeles claves, pero en cambio la segunda imagen, el fondo deja ver con claridad más hojas lo que permite tener más detalles es decir tiene un mejor contraste y puede tener más pixeles claves, en conclusión, el target debe tener detalles fácilmente visibles y de preferencia con diferencia de tonos destacable.

Por último, ignorando la última recomendación que es más técnico, tenemos el hecho de evitar lo patrones repetidos, recordemos cómo funciona el reconocimiento de imágenes, almacena los pixeles claves y luego realiza una búsqueda, al usar patrones repetidos, Figura 3-16, es muy probable que el programa tenga problemas a la hora de reconocer la imagen puesto que al tener puntos tan parecidos el programa se le dificulta identificar desde que punto se mira.



Figura 3-16 Clasificación de imágenes con Vuforia – Patrones repetidos. Imagen tomada de *Image Targets Optimization Techniques* [Fotografía] Vuforia Engine (2024) [20]

Otra recomendación que hace Vuforia es no usar tantas imágenes redondas o mejor dicho detalles curvos poco visibles, recordemos que una imagen es un grupo de cuadrados entonces para hacer una curva se requieren varios pixeles y muchas imágenes estas curvas de pixeles no son muy distinguibles entre ellas.

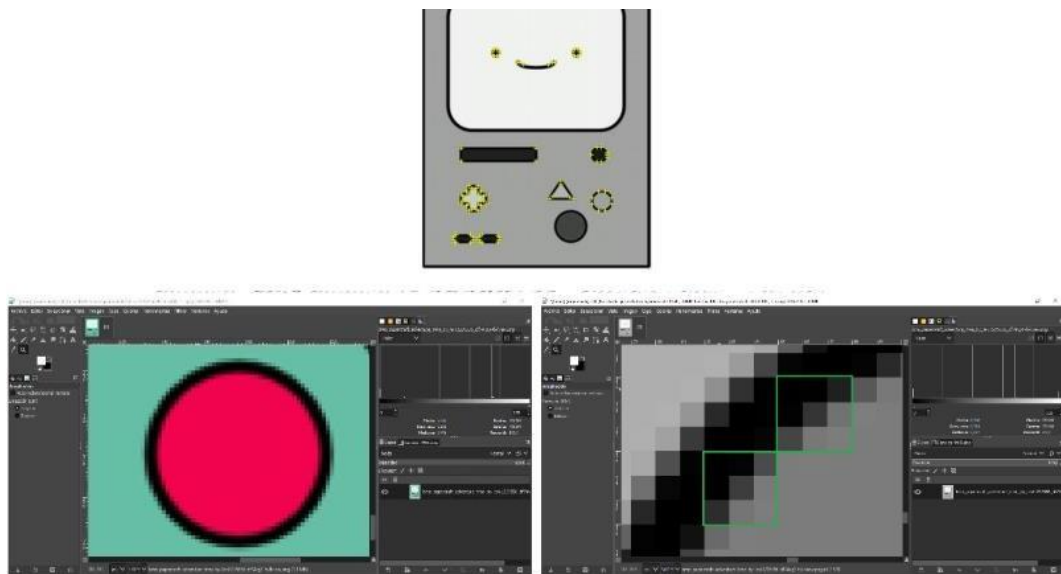


Figura 3-17 GIMP – Histograma. Imagen tomada de *GIMP* [Pantallazo] GIMP (2024) [21]

En la imagen se muestra una parte de la imagen de BMO como en una figura curva se pueden encontrar patrones repetidos los cuales Vuforia prefiere ignorar, del mismo modo en eso ocurre en otras partes como los bordes redondeados, sin embargo, no se afirma hacer uso de solo figuras rectangulares o picudas, siempre y cuando el borde este bien definido la imagen puede usarse a pesar de tener una forma irregular.

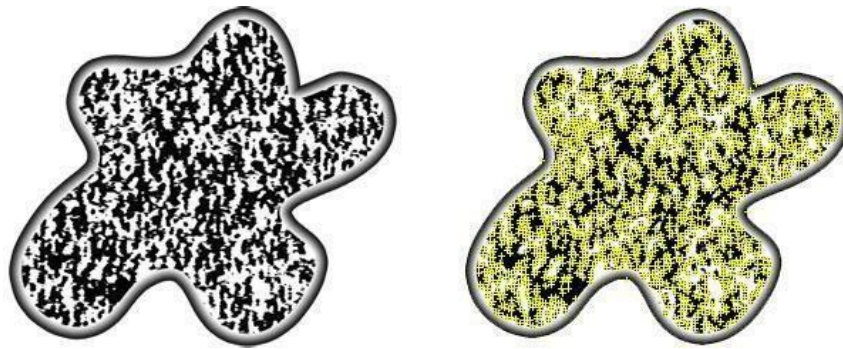


Figura 3-18 Clasificación de imágenes con Vuforia – Imagen irregular. Imagen tomada de *Image Targets Optimization Techniques* [Fotografía] Vuforia Engine (2024) [20]

Por último, debemos destacar que Vuforia crea una margen invisible del 8% de la imagen, esto hace porque en los bordes naturales es muy común encontrar patrones repetidos por lo tanto opta por crear un área de estudio para poner los pixeles claves se recomienda poner uno mismo este borde en blanco solo si el área de trabajo no tenga características en sí, es decir no se puede crear una confusión entre pixeles.



Figura 3-19 Clasificación de imágenes con Vuforia – Borde. Imagen tomada de *Image Targets Optimization Techniques* [Fotografía] Vuforia Engine (2024) [20]

Con esta información nos dispusimos a crear imágenes que cumplieran con las pautas para ser buenos targets, para esta tarea se utilizó la inteligencia artificial disponible en línea Leonardo.AI, con la licencia gratuita de la IA creadora de imágenes pudimos realizar varias imágenes.

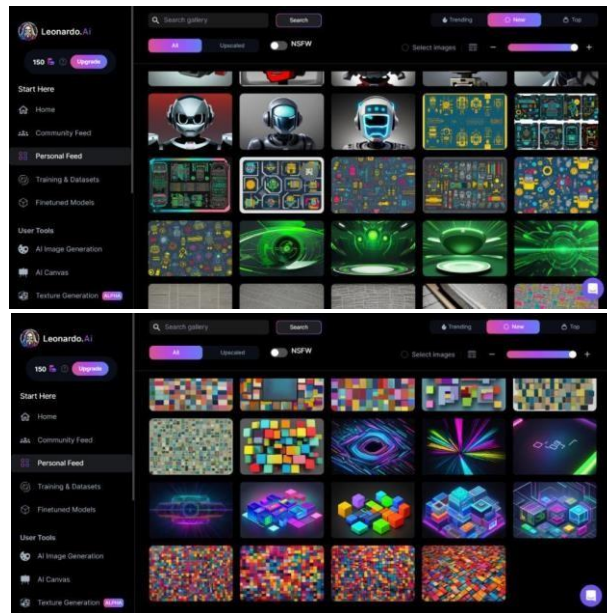


Figura 3-20 Interfaz de leonardo IA. Imagen tomada de *Image Leonardo IA* [Fotografía] Leonardo Interactive Pty Ltd (2024) [22]

Con algunas de estas imágenes se realizaron montajes en GIMP, con el objetivo de probar si en un futuro podemos realizar imágenes con logos más acorde a la temática del proyecto, Figura 3-21, las conclusiones de la práctica tenemos que los logos no deben ocupar gran parte de la imagen, es decir deben dejar ver muchos detalles, del mismo modo ayuda que dentro del logo hallan patrones útiles para ser usado como pixeles claves por último se recomienda que los trazos de los logos sean con colores que generan un contraste fuerte con el resto de la imagen de este modo podemos crear más pixeles claves con los bordes de los logos y añadir utilidad a esta práctica.



Figura 3-21 Imágenes creadas a partir de los conocimientos obtenidos. Imagen tomada de *Image Leonardo IA* [Fotografía] Leonardo Interactive Pty Ltd (2024) [22]

Una vez finalicemos la configuración de nuestros objetivos, podremos iniciar el nivel de realidad aumentada.



Figura 3-22 Sección de realidad aumentada de la aplicación del proyecto

En la sección de realidad aumentada, encontraremos una interfaz más pequeña y con menos elementos en pantalla para asegurar la comodidad del jugador al monitorear al robot, sus interacciones y un mejor disfrute al momento de ver los modelos en pantalla. Un botón aparecerá cuando se pueda interactuar con los residuos.

3.6.IMPLEMENTACION

En primer lugar, para conocer qué tan buenas son las imágenes creadas se subieron a la base datos de Vuforia en todas era 5 de 5, luego pudimos probar las imágenes dentro de una aplicación, el resultado es mucho más preciso, la detección de imágenes era mucho mejor, los objetos ya no tendían a moverse ni a aparecer en posiciones ilógicas incluso pudimos rotar las imágenes y la aplicación sin mucho problema podía seguir movimientos, los objetos también se movían o rotaban, demostrando la utilidad de haber realizado esta investigación y práctica de estos targets.

3.6.1. SEGUIMIENTO Y DETECCIÓN DE TARGETS

Una configuración muy importante, que en un principio fue ignorada, es el seguimiento del dispositivo, en pocas palabras es la reacción del programa cuando un target activo ya no es enfocado en la cámara, debe mantener su posición o debe desactivarse, para configurar debidamente esto se ofrece tres opciones.

En la configuración de acciones de la pantalla de Unity se nos permite usar “Tracked”, “tracked or Extended tracked” y “tracked or Extended tracked or Limited”, en primer lugar esta “Tracked” que significa ligeramente “rastreado” que hace referencia a encontrar el target cuando lo encuentra realiza la interacción que tenga programada, si se desea seguir ejecutando esta programación cuando el target salga de la cámara, se usa el “Extended tracked”, el programa usa diferentes métodos para calcular la posición del dispositivo y tener una idea de donde se encuentra el target cuando sale de cuadro, cuando se tiene problemas para hacer este proceso se recurre a “Limited” que sería el mismo proceso pero mucho menos preciso.

La solución a este problema es cambiar a “Tracked” con esto los objetos desaparecen a la hora de no ser visibles lo que permite un mejor control con el juego que tiene gran movimiento en la cámara.

3.6.1.1. GAMEOBJECT Y TARGETS

Con la detección de símbolos debidamente ajustada pasamos a trabajar en la interacción que tiene los objetos que depende de los targets, algo que notamos es que estos no dejan de ser GameObjects de unity por lo tanto pueden tener todo lo que tenga un objeto en un videojuego 3D, lo importante es saber aplicar las opciones de unity para nuestro objetivo.

Lo primero es verificar cómo se detectan los GameObject entre ellos cuando los targets estén cerca, cuando la cámara puede ver dos objetos virtuales, lo primero que se pensó era en hacer uso de sistemas de coordenadas, pero las prácticas demostraron que la posición de los objetos siempre será 0, en todos sus puntos cardinales y rotación, se mueva el target o no, por lo tanto la idea del seguimiento de coordenadas quedo completamente descartada, sin embargo como se dijo con anterioridad no dejan de ser GameObject, por lo tanto con la misma configuración que se usa para un videojuego normal, se puede aplicar a los colliders y el rigidbody, de este modo podemos crear interacciones entre los objetos, sin embargo hay un par de configuraciones para que el juego no presente muchos inconvenientes:

1. Colliders de gran tamaño, cuando se hicieron las primeras prácticas los objetos se tenían que juntar demasiado para poder interactuar entre ellos, en algunos casos interfiere con los targets por lo cual se optó por usar colliders de gran tamaño para poder detectar el objeto sin necesidad de acercarse tanto.
2. Usar triggers, lleno de la mano con el punto anterior es recomendable ajustar el collider como trigger u objetos no sólidos, esto para asegurar que las físicas de los objetos no interfirieran con la posición del objeto en la cámara.
3. Anclar el rigidbody, cuando se usó por primera vez el objeto por las físicas de Unity este tendía a caer y sin un suelo rápidamente dejó de ser visible, por lo tanto, se optó que su coordenada no variará, sin embargo, el objeto a veces tiende a deslizarse por lo cual se recomienda que todas sus coordenadas y rotación se mantengan constantes o ancladas.

3.7. GAMEPLAY

Con la detección de símbolos debidamente implementada se debe trabajar en el gameplay de esta sección, la idea del juego consta de recorrer un terreno real pasando por las canecas y el portal, donde su posición la elige el jugador, lo que se debe hacer en el juego es reconocer que tipo de residuo tiene el robot y con eso ir a depositarlo en la caneca adecuada y una vez termine dirigirse al portar terminando la parte de realidad aumentada.

Lo primero que se hizo fue crear la mecánica de que se debe pasar por tres targets y luego salir, en primeras versiones estos targets que se llamaran estaciones tenían el modelo de los enemigos además en pantalla saldrían tres puntos rojos para poder indicar los objetivos, además de un botón que aparecerá cuando el robot se acerque lo suficiente a las estaciones y con este dar por completado el objetivo.

Las estaciones se hicieron de forma que cada vez que se cargara la escena siempre fueran aleatorios, es decir, si en el target 1, salía el monstruo con forma de caja, al volver a cargar el monstruo podría estar en el target 2, 3 o incluso repartir el 1, para saber que modelo está hay que ir al target, para lograr esto se creó un algoritmo que contenía modelos y tenía referencia de los targets, y una vez iniciada la escena este ordenaría de forma aleatoria los modelos y luego los

enviaría a los target como hijos y en la posición 0, de tal forma que cuando se detecte el target el modelo salga sobre él de forma automática y se quede ahí siguiendo el target aunque se mueva.

Sin embargo, el sistema de detección es algo limitado, y solo nos permite interactuar con funciones específicas y esta configuración se debe ajustar antes de ejecutar el juego por lo cual se recurrió a crear un código que pudiera activar lo GameObject sin necesidad de esta referencia previa, este código estará presente en las canecas, cuando inicie el juego buscaremos que objetos tienen de hijo y lo guardaremos como variable y además tienen una función enfocada en esta variable su tarea será variar el estado del objeto, es decir si está apagado y se llama la función el código prenderá el hijo que es un enemigo y viceversa, sin embargo se lidia con el problema que si el código que asigna a los hijos se ejecuta antes del que revisa los hijos dará un error ya que no encuentra que debe encender o apagar, la solución es aprovechar la jerarquía de código de Unity, resulta que por defecto al crear un código nuevo se crea una función llamada “start” que es la función que por defecto se ejecuta de primeras, y una sola vez, siempre antes de ejecutar la función “Update” que es una función loop que se ejecuta cada frame, sin embargo hay una función igual que “start” además se ejecuta antes que esta y es la llamada “awaked”. Por lo tanto, debemos de poner la función que encarga de repartir los hijos en Awaked y revisar los hijos en start, de esta forma podemos asegurar que los hijos serán asignados y correctamente referenciados, con esto solo resta configurar que al detectar el objeto recurra a la función que hace variar el estado de los modelos 3D.

Ahora para poder decir si el objetivo se completó se implementó el botón, se añadió al código de los enemigos poder detectar cuando el jugador esté cerca y con eso activar un botón que este recurrirá a otro GameObject que maneje toda esa lógica, este sistema no es aleatorio, tenemos tres variables bool que representan las estaciones, al oprimir el botón estas cambiarán a ser verdadero diciendo que se completó la tarea, estas son fijas para los targets, los objetos 3D siguieran apareciendo en el target pero el botón no, además estas variables bool también se usaron para saber si ya se completaron las tareas y poder dar fin, el medio para salir de la escena varió con el tiempo primero fue un botón igual a los ya mencionados, más adelante se implementó los botones de realidad aumentada pero estos fueron deshabilitados luego de un par de actualizaciones y nuestro intento de replicarlos no dio frutos pero la solución fue implementar un portal para añadir un poco más de coherencia con el juego, para poder salir de la escena el robot debe de entrar a un portal, como si estuviera viajando por mundos.

En versiones más cercanas a la actual los modelos de los enemigos se cambiaron por el de las canecas y se implementó el inventario del videojuego digital, ahora lo que se recoja en el mundo virtual se verá en el mundo en realidad aumentada, para poder implementar eso debidamente lo primero es dar a conocer al jugador que residuo debe tirar por lo tanto se añadieron los modelos de los residuos como hijos del modelo del robot, como con los targets aleatorios, pero debe darse un orden de cuando aparece por lo que se volvió a implementar el algoritmo aleatorio para hacer variar el orden de los residuos.

Ahora para leer que se tenía en el inventario primero se debió ajustar esta variable como algo que no se destruyera una vez de cambiar la escena y sin embargo esto es suficiente, ya que los objetos deben de tener una referencia de este y al no destruir el objeto cuando se cambia de escena se destruye el que si estaba antes, para poder hacer las pruebas se requirió un GameObject que contuviera esta información y por la lógica de cambio de escena cuando se inicia el objeto nuevo de la escena se pregunta si ya existe uno de escenas anteriores y si es verdad se destruye, perdiendo

referencia que se habían hecho con anterioridad para solucionar esto lo único que hay que hacer es que los objetos que lo requieran lo busquen según su nombre y hacer creen una referencia, esta práctica si se aplica para muchos objetos genera problemas de rendimiento sin embargo como es solo un objeto no hay problema.

Con los residuos ya visibles y el inventario ajustado se debió implementar la lógica con su respectiva caneca y desactivar/activar los residuos que no se han reciclado, para poder definir que residuo debe ir en cada caneca recurrimos a usar el sistema de etiquetas o “tags” de Unity, el residuo compartirá etiqueta con su respectiva caneca sin embargo este método supone que cada residuo tenga parámetros como el del robot para ser detectado, ya con la experiencia previa no fue problema, lo único que resta es ajustar los códigos de las estaciones, para poder activar el botón de la interacción además de entrar en el collider debe cumplirse que las etiquetas de los objetos sean iguales, de lo contrario no se hará visible el botón, y cuando sea posible ahora el botón en lugar de cambiar una variable bool, recurrirá al inventario y restará los residuos desechados, una vez el inventario este vacío el botón no saldrá más.

Para realizar la rotación de los objetos se implementó un código que contara las acciones del jugador, es decir cuando un residuo del inventario llegara a cero se suma una acción, hay completar 3 acciones para poder terminar al juego, gracias a este número entero podemos definir cuál de los tres residuos estará encendido mientras los demás apagados, por último con este algoritmo implementado creamos el portal, este debe revisar que las acciones sean 3 y así salir del mundo virtual cuando detecte que el robot está cerca, de lo contrario el portal no hará nada.

4. DESARROLLO DEL ROBOT

Un robot móvil debe tener un método claro y adecuado de movimiento que defina cómo se desplaza, la superficie sobre la cual opera y otras características importantes. En el diseño, es crucial considerar las propiedades físicas como el peso, tamaño y método de transporte, así como la estabilidad del equipo.

El sistema de movimiento de un robot móvil se ve principalmente afectado por su peso, ya que este factor determina su movilidad y la potencia necesaria para los motores que lo desplazan. El peso del robot depende del chasis, los motores, las baterías y los componentes electrónicos. Cuando un robot es demasiado pesado, se deben elegir motores adecuados, seleccionar materiales livianos para el chasis y considerar métodos para reducir su peso.

Reducir el peso es a menudo abordado mediante estrategias como optimizar el diseño de la fuente de energía, aunque esto puede afectar la autonomía del robot. Esto implica diagramar los requerimientos de energía inmediatos y establecer prioridades de alimentación según la aplicación específica.

En el aspecto mecánico, los sistemas de movilidad del robot están vinculados con el sistema de transmisión mecánica, los cálculos de velocidad y la resolución espacial mecánica necesaria para definir sus características. Los sistemas de movilidad principales se pueden clasificar en cuatro tipos: diferencial, similar a las orugas de los tanques militares; automóvil, que utiliza un motor de tracción y otro de dirección; triciclo, común en experimentos de robótica con estabilidad asegurada por ruedas locas; y omnidireccional, avanzado y tecnológicamente complejo, que utiliza un motor para cada extremidad para lograr movimientos multidireccionales.

4.1. DISEÑO Y CONSTRUCCIÓN DEL ROBOT AZONI

4.1.1. DISEÑO GENERAL DEL ROBOT

Para poder realizar el diseño del robot necesitamos tener en cuenta las diferentes necesidades que nos exigen el proyecto siendo estas las siguientes:

- Facilidad al controlar
- Versatilidad
- Simplicidad
- Robustez
- Equilibrado

Se evalúa elaborar un robot lo más similar al del proyecto original, conocido como Azoni, este robot se categoriza como un robot móvil de tracción diferencial.

El equipo debido a esto pide apoyo al proyecto grande sobre materiales de este robot, donde se es proporcionada la base del robot junto a sus llantas, su material es de Aluminio en su gran mayoría lo cual ayuda a tener poco peso con una alta resistencia.

El aluminio es ligero pero resistente debido a su baja densidad, estructura cristalina cúbica centrada en las caras que proporciona orden en la disposición de átomos, una excelente fuerza específica que equilibra la resistencia y el peso, la capacidad de formar aleaciones como el aluminio- litio y aluminio-magnesio para mejorar sus propiedades, y la formación natural de una capa delgada de óxido de aluminio en su superficie que le confiere resistencia a la corrosión. Estas características hacen del aluminio un material versátil, utilizado en una amplia variedad de aplicaciones, desde la industria aeroespacial hasta productos cotidianos. (Boyfre 2023).

Y en la parte de las llantas están hechas con plástico donde son compuestas con poleas lo suficientemente gruesas para la correa de las llantas, este tipo de llantas son las llamadas llantas oruga. Este tipo de llantas están diseñadas para un ambiente diferente a lo convencional, como lo son caminos rocosos, montañosos, difícil movilidad y un buen agarre al terreno, lo malo es que este tipo de sistemas ganan fuerza a costa de velocidad.

Evidenciamos que el modelo en base al proyecto original se ajusta a la perfección con las necesidades del proyecto, ya que con la base fabricada en aluminio encontramos las cualidades de robustez y equilibrio requeridos, y por medio de las llantas oruga contamos con la versatilidad, la facilidad al controlar y la simplicidad.

4.1.2. ESPECIFICACIONES TÉCNICAS:

- Tensión nominal: 6 V
- Rango de voltaje de funcionamiento: 2 ~ 7,5 V
- Relación de reducción de engranajes: 45:1

- Diámetro del eje de salida: 4 mm
- Velocidad sin carga: 133 RPM @ 6 V
- Corriente sin carga: 0,13 A
- Par de rotor bloqueado: 4,5 kg.cm
- Corriente de rotor bloqueado: 2,3 A
- Capacidad de carga: 3 kg
- Dimensiones ensambladas (largo x ancho x alto): 225 x 220 x 108 mm / 8,86 x 8,66 x 4,25 pulgadas
- Peso neto: 1,3 kg

4.1.3. MATERIAL UTILIZADO PARA EL ROBOT Y SUS BENEFICIOS

El chasis del robot Azoni está fabricado principalmente de aluminio, un material conocido por su ligereza y resistencia. Estas son algunas de las características que hacen del aluminio una excelente elección:

Ligereza: El aluminio es baja densidad, reduciendo significativamente el peso total del robot.

Resistencia: Su estructura cristalina cúbica centrada en las caras proporciona una excelente relación entre resistencia y peso.

Aleaciones: El aluminio puede formar aleaciones con otros metales como el litio y el magnesio, mejorando aún más sus propiedades mecánicas.

Resistencia a la corrosión: La formación natural de una capa de óxido de aluminio en la superficie del material le confiere una protección eficaz contra la corrosión.

4.1.4. IMPLEMENTACIÓN DE CHASIS

El diseño y construcción del robot Azoni refleja una cuidadosa consideración de los materiales y la configuración mecánica para maximizar su rendimiento en terrenos desafiantes. La elección del sistema de locomoción tipo oruga, el uso de aluminio en su estructura, y el diseño robusto de las llantas y orugas, aseguran que Azoni sea capaz de operar eficazmente en una amplia variedad de condiciones, proporcionando una solución versátil y confiable, Figura 4-1.

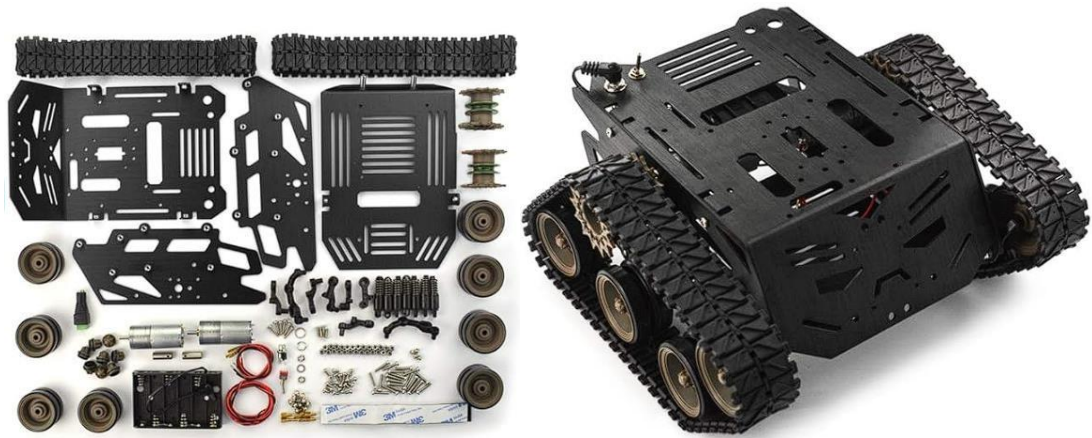


Figura 4-1 Chasis del robot antes y después de armarlo. Imagen tomada de *Tanque devastador – plataforma* [Fotografía] DFROBOT (2024) [23]

4.1.5. CARCASA DEL ROBOT

Una vez realizada las validaciones anteriores necesitamos proponer la manera que este robot físico se muestre como el robot Azoni en la realidad aumentada a implementar en el sistema, la propuesta es utilizar una caja de cartón que además de tener la función de esta “interconexión” cuente con la funcionalidad de protección para el circuito eléctrico utilizado para controlar los motores, así como la comunicación con el dispositivo móvil. Esta no cuenta con una decoración en específico ya que se quiere enfatizar la transformación de este robot en la realidad aumentada, teniendo como base que algo tan sencillo como una caja de cartón puede llegar a generar este modelo 3d de un robot futurista.

Se optó por una estructura simple y sin muchos detalles. La razón principal es que facilita la integración de la realidad aumentada del proyecto. Cuando detecta el target de realidad aumentada la cámara del celular, esta carcasa permite proyectar de manera precisa los elementos virtuales del robot. Este enfoque permite que el usuario experimente un impacto visual significativo, al transformar una carcasa simple en un diseño virtualmente enriquecido y detallado, realzando la experiencia de interacción con el robot.

Además, a su facilidad para ser personalizada y su ligereza. Esta elección no solo reduce el peso total del robot, sino que también facilita la integración de targets de la realidad aumentada elaborados en el proyecto.

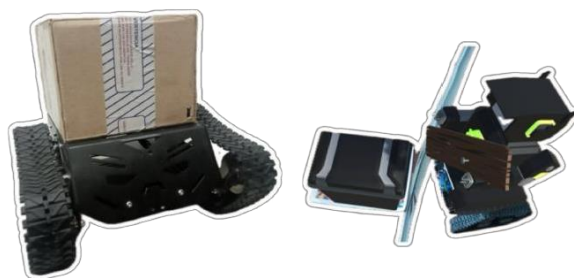


Figura 4-2 Robot Azoni en la realidad y en la sección virtual visto desde un dispositivo móvil [23]

4.2. SISTEMA DE LOCOMOCIÓN TIPO ORUGA

Azoni es un robot diseñado con un sistema de locomoción tipo oruga, el cual se caracteriza por su capacidad de mejorar la tracción y la fuerza en diversos terrenos. Este sistema, aunque sacrifica algo de velocidad, ofrece numerosas ventajas en términos de estabilidad y movilidad en superficies difíciles, Figura 4-2.

4.2.1. DESCRIPCIÓN DEL SISTEMA DE LOCOMOCIÓN TIPO ORUGA

El sistema de locomoción tipo oruga consiste en un conjunto de ruedas conectadas mediante una banda continua (la oruga) Figura 4-1 que rodea todo el perímetro del conjunto. Esta configuración permite que el robot distribuya su peso de manera uniforme sobre una superficie mayor, reduciendo la presión sobre el suelo y mejorando su capacidad de atravesar terrenos irregulares.



Figura 4-1 Chasis del robot. Imagen tomada de *Tanque devastador – plataforma* [Fotografía] DFROBOT (2024) [23]

4.2.2. VENTAJAS DEL SISTEMA DE LOCOMOCIÓN TIPO ORUGA

1. Tracción Mejorada:

Las orugas ofrecen mayor superficie de contacto con el suelo que las ruedas convencionales, lo que supone una tracción mejorada. Esto es especialmente útil en terrenos blandos, como barro o arena, donde las ruedas podrían hundirse.

2. Estabilidad:

La mayor superficie de contacto no solo mejora la tracción, sino que también contribuye a una mayor estabilidad del robot. Azoni puede mantener su equilibrio con mayor facilidad en terrenos irregulares o inclinados.

3. Capacidad Todo Terreno:

Las orugas permiten a Azoni superar obstáculos como rocas, raíces y otros elementos que serían difíciles de sortear con ruedas convencionales. Este tipo de sistema es ideal para aplicaciones en exploración, rescate y operaciones en terrenos difíciles.

4.2.3. DESVENTAJAS DEL SISTEMA DE LOCOMOCIÓN TIPO ORUGA

1. Velocidad Reducida:

Una de las principales desventajas del sistema de orugas es la velocidad. Debido a la mayor fricción y al peso adicional de las orugas, Azoni no puede alcanzar las mismas velocidades que un robot equipado con ruedas convencionales.

2. Mayor Complejidad Mecánica:

El sistema de orugas es más complejo desde el punto de vista mecánico. Requiere más componentes móviles y, por lo tanto, puede ser más susceptible a fallos mecánicos y desgaste. Además, su mantenimiento puede ser más exigente.

3. Consumo de Energía:

La fricción adicional y la masa de las orugas pueden llevar a un mayor consumo de energía, lo que podría reducir la autonomía del robot si no se gestiona adecuadamente.

4.2.4. FUERZA Y TRACCIÓN VS. VELOCIDAD

El diseño de Azoni se centró en maximizar la fuerza y la tracción, considerando que su principal objetivo es operar en espacios controlados donde se puede presentar una superficie irregular y además de un espacio pequeño de operación, donde la velocidad no es el factor más crítico. Las orugas permiten distribuir la fuerza de tracción a lo largo de una mayor área de contacto, lo que resulta en una fuerza de tracción superior que la de las ruedas convencionales.

La implementación del sistema de locomoción tipo oruga en el proyecto Azoni ha sido una decisión estratégica basada en un análisis detallado de las necesidades operativas del robot. Aunque se sacrifica algo de velocidad, las ventajas en términos de tracción, estabilidad y capacidad para sortear terrenos irregulares controlados hacen que este sistema sea ideal para el propósito de Azoni. La combinación de estas características asegura que Azoni pueda operar de manera eficiente y efectiva.

4.3. CONTROL DEL ROBOT

Al iniciar el proyecto ya se contaba con una estructura para el robot, esta consiste en un armazón con dos motores con ruedas tipo oruga, es decir ya se contaba con una estructura principal al inicio, además de eso se contaba ya con un controlador, un Arduino con un Shield Driver L293D que es un dispositivo tipo puente H para cuatro motores adaptado para Arduino, por lo tanto ya se contaba con todos los materiales para poder iniciar con la programación, el Shield cuenta con su propia librería que permite fácilmente manipular la velocidad y sentido de giro.

En primer lugar es muy importante definir qué puertos de la Shield se van a usar para poder definirlos en el programa como una variable y poder aplicar los comandos, para manipular la velocidad del motor se usa un números entre 0 y 255, en términos técnicos un número en 8 bits, esto define cuánto voltaje saldrá del puerto siendo que 255 es el máximo que son 5 voltios y en contraparte 0 es apagar el puerto, el segundo comando es para poner andar el motor y definir el sentido de rotación, el comando directamente abre pasa de voltaje por el puerto pero lo que se debe tener en cuenta es la dirección la cual si se puede manipular para ello se debe tener claro el uso de FORWARD y BACKWARD, que literalmente significa “Hacia adelante” y “Hacia Atrás” respectivamente, lo que hace el comando es invertir la polaridad de la salida del puerto, es decir para ir adelante, usar FORWARD, implica que los voltios serán positivos mientras que usar BACKWARD o ir hacia atrás genera una salida negativa, de 0 a -5 voltios, siendo acorde de la naturaleza de los motores.

```
#include <AFMotor.h> AF_DCMotor Variable(N puerto);
```

```
Variable.setSpeed(0 a 255);
```

```
Variable.run(FORWARD o BACKWARD);
```

La programación no presentó mayor problema sin embargo hay aspectos importantes a tener en cuenta cuando se trabaja con esta Shield, en primer lugar cuando solo se trabajan con dos puertos

es recomendable usarlos en “X”, es decir si se elige el puerto 1 se debe usar el puerto 3 y con el 2 el 4, esta configuración permite que los dos motores tengan en su máxima velocidad los 5 voltios, fuimos testigos de cómo uno de los puertos pierde voltaje al no seguir esta recomendación, siendo que el puerto puede dar tan solo 2.5 voltios.

Otro aspecto de suma importancia es cómo se va a mover, es decir específicamente que deben hacer los motores a la hora de que el robot debe girar, la solución para esto es muy sencilla y es reducir la velocidad del motor según se quiere voltear por ejemplo si se desea girar a la derecha el motor que maneje la rueda derecha debe tener menos fuerza que el de la izquierda, de este modo la rueda que está quieta o con menos fuerza se convierte en un eje de giro, la fuerza del motor solo está de un lado lo que genera la rotación esto es natural en los vehículos con ruedas tipo oruga.

4.4. CONEXION INALAMBIRCA

Uno de los pilares del juego era la conexión con el robot, la tarea consistía en enviar comandos desde el teléfono y que el robot actuará en base a estos, por ahora solo sería la manipulación de dos motores a través de un joystick. Esta conexión inalámbrica se puede realizar por los siguientes métodos

- Bluetooth: es una tecnología que permite la comunicación inalámbrica de voz y datos entre dispositivos a través de un enlace de radiofrecuencia esto reduciendo los problemas generados por el cableado, así como dar seguridad entre los equipos vinculados a los montajes y dando facilidad para la creación de pequeñas redes inalámbricas
- Wi-Fi: Una conexión wifi usa ondas de radio y tecnología inalámbrica, brindando a los usuarios unos elevados niveles de movilidad y flexibilidad para sus conexiones

En primer lugar se optó por usar bluetooth debido a que todos los teléfonos traen esto lo que hace accesible esta opción, por lo tanto lo primero es hacer una conexión vía Bluetooth entre el Arduino y Unity, para esto se evaluó los diferentes periféricos disponibles en el mercado para realizar ese tipo de conexión cumpliendo con la compatibilidad de los periféricos utilizados en el control del robot; encontrando así dos módulos conocidos por amplia disponibilidad en el mercado, su bajo costo, compactos y versatilidad, estos siendo los siguientes:

Modulo HC-05: La transmisión se realiza con transparencia al programador, conectando directamente a los pines seriales de nuestro microcontrolador preferido. Este viene configurado de fábrica para trabajar como esclavo, es decir, preparado para escuchar peticiones de conexión, pero podemos configurarlo para trabajar con Maestro utilizando comandos AT. Por otra parte, si nuestro objetivo es conectar dos proyectos, necesitaremos utilizar un módulo HC-05 configurado como maestro y un HC-06 (esclavo) o un HC-05 configurado como esclavo.

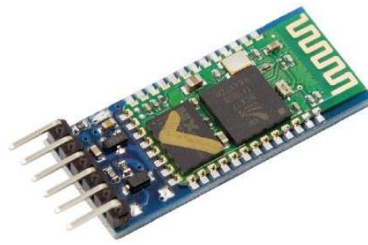


Figura 4-2 Modulo HC-05. Imagen obtenida de *MecaBot* [Fotografía] MecaBot (2024) [24]

Modulo HC-06: La transmisión se realiza con transparencia al programador, conectando directamente a los pines seriales de nuestro microcontrolador preferido. El módulo Bluetooth HC-06 viene configurado de fábrica para trabajar como **esclavo**, es decir, preparado para escuchar peticiones de conexión. Por otra parte, si nuestro objetivo es conectar dos proyectos, necesitaremos utilizar un módulo HC-05 configurado como maestro y un HC-06 (esclavo).



Figura 4-3 Modulo HC-06. Imagen obtenida de *Mercado Libre* [Fotografía] Vistronica (2024) [25]

Físicamente los dos módulos son muy parecidos y la mejor manera de diferenciarlos es por los pines de E/S (Entradas y salidas) que nos ofrecen, en este caso encontramos 4 pines para el módulo HC-06 y 6 pines para el HC-05. A continuación os describo la función de cada uno de los pines:

VCC: Alimentación del módulo entre 3,6V y 6V.

GND: Masa del módulo.

TXD: Pin para la transmisión de datos.

RXD: Pin para la recepción de datos. Trabaja a un voltaje de 3,3V, aunque la mayoría de los módulos ya trae convertidor de niveles lógicos integrado, con lo que lo podemos conectar directamente al pin TX de Arduino (5V) sin tener que recurrir a HARDWARE externo para bajar el voltaje.

EN: Pin para entrar en el modo de configuración del módulo (solo en el modelo HC-05).

STATE: Pin de estado (salida) para conectar un led y visualizar cuando se está comunicando el módulo (solo en el modelo HC-05).

Aunque no hay diferencias significativas en el hardware entre ambos modelos, sí hay diferencias en el firmware, es decir, en la programación interna del módulo. El HC-06 entra en modo de programación (comandos AT) siempre que lo enciendes y no esté conectado a él ningún dispositivo a través de Bluetooth. En cambio, el HC-05 es un poco más complicado de configurar en modo comandos AT y requiere un modo de arranque específico.

Ya que conocemos las aplicaciones y características de cada uno de los módulos, evaluamos cada uno de estos según la necesidad que encontramos en el proyecto, siendo estas un dispositivo al que realicemos la conexión y de esta manera poder realizar el envío de la información se escoge el módulo HC-06 por la facilidad en el encendido además de la facilidad que ofrece para cambiar entre modo de esclavo o maestro.

Usó el módulo HC-06 que es un dispositivo que permite a Arduino enviar y recibir información por los puertos RT y TX, recurrimos a un plugin gratuito dado por el usuario Ahmed BENTALEB de YouTube, gracias a este programa pudimos enviar y recibir datos del tipo string o dicho de otra manera información en texto que es el formato que puede manejar el módulo Bluetooth.

Módulo Esp8266: Sin embargo también se exploró la opción de transferir los datos por internet para esto se usó un módulo ESP8266 presentado como un wemos mini D1 que tiene capacidad de conectarse con a internet vía wifi además este módulo también puede trabajar por comunicación serial pero esta se da solo a 3,3v siendo que el Arduino maneja 5v podría quemar el módulo por lo que se optó que solo Arduino recibiera información del ESP8266 y no viceversa, para poder comunicar unity con wemos debimos ajustar este último como un servidor local, para enviar los datos utilizaremos los enlaces, mediante el link podemos enviar datos por ejemplo podemos acceder por navegador a este enlace [http://\[Modulo\]/action?X=1&Y=20&Z=FORW](http://[Modulo]/action?X=1&Y=20&Z=FORW) y cambiando X, Y, y Z enviar variables a al módulo wifi.



Figura 4-4 Modulo Esp8266. Imagen obtenida de *Colombianizate* [Fotografía] colombianizate (2024) [26]

Ahora resta enviar datos desde Unity al servidor para esto se presentan dos opciones la primera es usar Unity WebRequest, que es una librería de unity que permite obtener el HTML de un sitio web entre otras opciones, esto mediante una variable tipo string que representa el enlace, sin embargo esta no es la mejor opción, en primer lugar siempre que se use descargar el HTML a pesar de que se

puede borrar es mucho esfuerzo para la memoria, la información que vamos a enviar será casi cada frame, en segundo lugar tenemos el problema de compatibilidad con Android, las versiones más actuales de Android exigen que los enlaces a trabajar sean https, por temas de seguridad, y no http que es el formato que usa el servidor.

Por lo tanto nos queda la segunda opción el uso de Sockets, que es la comunicación entre dos procesos para intercambiar datos localmente y entre redes, las ventajas de usar este sistema son que no requiere un enlace http o https por lo cual no presenta problemas a la hora de portarlo a Android, este proceso requiere la IP que le dio el internet a el módulo y del mismo modo no descarga nada, realiza un proceso de acceso vía get o post por lo cual es solo enviar información permitiendo no esforzar mucho la memoria resultado en un método más práctico.

Lo siguiente es manejar el formato para que enviar los datos, lo primero es convertir la información del joystick en velocidad y dirección para el motor, para ello debemos sacar las coordenadas del joystick, este funciona por un sistema de coordenadas X y Y que va de -1 a 1, con esto podemos definir qué es lo quiere hacer el usuario, si quiere ir adelante sube el joystick y sus coordenadas son (1,0), para retroceder (-1,0) y para girar (0,1) o (0,-1).

Fácilmente podemos deducir si el usuario desea ir hacia adelante o hacia atrás gracias a su coordenada en x, si es positiva hacia adelante y si es negativa hacia atrás, lo que resta es saber a qué velocidad la cual es una ecuación muy simple para convertir los números entre 0 y 1 a números entre 0 y 255, esto se debe a que podemos normalizar la velocidad y con otro comando definir el sentido, por lo cual podemos tomar el valor absoluto del eje x y trabajar.

```
int velocidad motor = abs(dato1 unity)*255;
motor1.setSpeed(velocidad motor);

if(direccion >= 0){
motor1.run(FORWARD)
;
motor2.run(FORWARD)
;
}else{
motor1.run(BACKWARD);
motor2.run(BACKWARD);
}
```

Ahora para poder girar lo que se requiere es que un motor tenga más fuerza que el otro, de este modo el motor más débil se vuelve un eje de giro lo que permite que el frente cambie de dirección, pero añadir fuerza no sería posible si el usuario ya está en máxima velocidad por lo tanto la solución más práctica es reducir la fuerza del motor en base de la dirección, por ejemplo si se dese girar a la izquierda el motor en la izquierda debe perder velocidad para que la fuerza del motor derecho genere el giro, para poder definir si el usuario volteara usaremos el valor de rotación o el del eje horizontal del joystick, cuando este es mayor a 0 está en la derecha y viceversa para la izquierda, una vez definida cual motor debe perder velocidad, ahora para definir cuál es el valor de la velocidad ahora, usaremos el valor de la rotación para hacer una conversión donde el 0 valdrá 225 y el 1 tendrá

el valor de 0, de este modo cuando el usuario quiera girar muy poco, moverá un poco el joystick es decir el valor será cercano a 0 y con la ecuación anterior el giro en los motores será también pequeño al no perder mucha fuerza, pero si el usuario quiere hacer un giro más brusco usara el joystick en un valor cercano al 1, con la ecuación la velocidad sería muy poco lo que permite que el giro será más pronunciado.

```
if(rotación > 0)///derecha
    velocidad motor 1= 255 * 1-abs(rotación); else
if(rotación < 0)///izquierda
    velocidad motor 2= 255 * 1-abs(rotación);
```

Esta es una manera de imitar los controles llamados coloquialmente “tipo tanque”, que son vehículos que cuentan con ruedas oruga al igual que nuestro robot.

Ahora bien, con la programación ya definida podemos ya implementar el formato para enviar los datos, lo primero es que el teléfono al tener mejor procesador va a hacer los cálculos previos para que el Arduino solo lea y actúe, por lo tanto, unity debe de enviar tres datos, dos velocidades y la dirección, el formato a manejar es:

vel1/vel2/FORW o BACK

Este dato se puede enviar directamente vía bluetooth y Arduino dividir la información y leerla directamente, por otro lado, si es por internet primero se debe enviar vía enlace con eso el módulo ESP enviara la información vía serial ya con el formato establecido.

5. RESULTADOS Y PRUEBAS DE FUNCIONAMIENTO

El resultado final del proyecto integra la parte del videojuego del mundo virtual junto con la realidad aumentada, el cambio entre estos mundos en principio es fluido ya que depende principalmente del equipo que se use, en su mayoría los teléfonos de hoy en día son capaces de ejecutar el juego sin mayor problema.

Por un lado el mundo “virtual” cumple con las expectativas iniciales, se implementó de forma correcta el mapa y modelos, por partes de las mecánicas de juego o gameplay también se logró implementar de forma satisfactoria, la pantalla de juego brinda la información necesaria, con su mini mapa y cámara de ayuda, el joystick y botones funcionan de forma óptima y cómoda para el jugador, ayudando con las mecánicas de juego o gameplay, este al final consiste en eliminar los enemigos, estos dejan un residuo, tomar este residuo y por último dejarlo en el espacio elegido para poder añadirlo al inventario que se usará en la segunda sección del juego. estas mecánicas fueron satisfactoriamente aplicadas.

En la segunda sección del juego el resultado final consta del movimiento del robot, identificación de los targets e interacción entre los modelos de realidad aumentada, empezando por los targets la identificación resulta de forma satisfactoria solo presenta problemas con imágenes pequeñas pero con el tamaño inicial, aproximadamente una hoja carta, funciona correctamente, por otro lado las interacciones funcionan de forma eficaz, los objetos sin mucha dificultad pueden detectar que otro objeto está cerca y poder realizar la mecánica, esta consiste en revisar el residuo que tenga el robot e ir a la caneca correspondiente para poder depositarlo, al hacer esta acción se restará el residuo del inventario y una vez llegue a cero, se cambiará el residuo e iniciara de nuevo, cuando se terminen los residuos se debería ir al portal para regresar al mundo virtual y recolectar todos los nuevamente residuos, este ciclo se efecto de forma apropiada.

Por parte de la locomoción del robot esta se realiza de forma rápida, aunque un poco confusa al inicio, sin embargo, la reacción del robot a la hora de enviar las señales es muy rápida lo que permite un juego fluido además las ruedas demostraron ser prácticas, en ninguna oportunidad se tuvo problemas con estas además de que, en suelos ligeramente rugosos, como pisos en ladrillo, su motricidad no se ve afectada.

5.1. VALIDACIONES INICIALES

5.1.1. VALIDACIÓN VIDEOJUEGO

En el proceso de creación de la sección virtual del videojuego Azoni, se realizaron diversas pruebas exhaustivas. Se probaron diferentes interacciones entre objetos para garantizar una jugabilidad fluida y coherente. Además, se elaboraron modelos detallados para ambientar el nivel, cuidando cada aspecto del entorno virtual y siguiendo la estética del ciberpunk, asegurando que esta temática se evidenciara claramente en cada validación realizada.

Se prestó especial atención a la calidad de la iluminación, el ideal con esto es crear una atmósfera inmersiva y agradable para el usuario. La lógica de programación fue revisada para asegurar que todas las mecánicas del juego funcionaran correctamente, sin errores.

5.1.1.1. Proceso de Testeo y Validación

El proceso de testeo de la aplicación abordó múltiples aspectos del juego para garantizar su calidad y rendimiento. Los desarrolladores como testers se encargaron de buscar errores y bugs, forzando el juego en diversas situaciones para asegurarse de que todos los posibles fallos fueran detectados y corregidos. Este enfoque incluyó intentar trabar el juego de diferentes maneras, como interactuar con múltiples objetos al mismo tiempo, explorar los límites del escenario, y realizar acciones fuera del comportamiento esperado del usuario.

5.1.1.2. Pruebas de Estrés

Se sometió al juego a pruebas de estrés, y a revisar errores gráficos y sonoros en diferentes dispositivos, al iniciar el juego se buscaba explorar en el mapa donde lográramos ver cómo interactuaba la luz con los diferentes objetos del mapa y para el sonido se probaba que la música del nivel sonara en las diferentes escenas para ver que no se trabara.

5.1.1.3. Pruebas de Compatibilidad

Se llevaron a cabo pruebas de compatibilidad en diferentes dispositivos móviles disponibles por el equipo, con diferentes sistemas operativos. Esto fue para observar el juego funcionara sin problemas en diversos celulares, donde se instalaba la aplicación y se veía su efectividad al momento de que funcionara, para saber si el juego estaba optimizado de la mejor manera.

5.1.1.4. Pruebas de Usabilidad

Se realizaron sesiones de pruebas de usabilidad con usuarios reales para identificar y solucionar problemas de interfaz y jugabilidad. Estas pruebas permitieron asegurarse de que la interfaz del juego fuera intuitiva y accesible, y también nos permitió identificar errores que el equipo no hubiera identificado. Los comentarios de los usuarios se utilizaron para realizar ajustes y mejoras. Mas adelante, se discutirán estos aspectos en mayor detalle.

5.1.1.5. Pruebas de Funcionalidad

Cada característica del juego fue probada individualmente para asegurarse de que funcionara según lo previsto. Esto incluyó probar las mecánicas de juego, como el movimiento del jugador y su interacción con el entorno, para evitar problemas como la capacidad de atravesar objetos u otros errores similares.

Gracias a estas pruebas y ajustes continuos, la calidad general del videojuego aumenta, asegurando que ofrezca una experiencia de juego rica y envolvente, fiel a la temática ciberpunk y técnicamente sólida.

5.1.2. VALIDACIÓN REALIDAD AUMENTADA

El primer objetivo a la hora de implementar la realidad aumentada es que el programa este en la capacidad de detectar símbolos e interactuar, esto se logró sin tanto problema sin embargo se presentaban varios inconvenientes al respecto, partiendo del antes mencionado posiciones ilógicas, los objetos 3D o GameObjects generados sobre los targets tendían a moverse de su posición inicial, ya sea porque un objeto se interponía entre los targets y el programa detectaba como si el símbolo se movía o por la configuración de los objetos, al mover el dispositivo este trataba de guardar la posición del target cuando no era visible en la cámara pero este proceso era poco preciso además se debe tener en cuenta la cantidad de GameObject que podían ser visibles al mismo tiempo, para solucionar estos problemas lo primero fue la creación de targets o símbolos mejor preparados para esta tarea, los resultados fueron imágenes muy llamativas y para probar su funcionamiento se hicieron las acciones que provocaban problemas, para empezar eclipsar el target, se interpusieron objetos entre la cámara y la imagen, ya sean cosas simples como la mano o bien otros targets, con las imágenes más elaboradas resulto más complicado mover el objeto 3D por este método, si el objeto no ocupaba mayor parte de la imagen no lo afectaría por otro lado si se cubría la mayor parte el objeto directamente desaparecía.

Sin embargo, aun teníamos posiciones ilógicas o no aparecían, para entender por qué se daba esto pusimos un texto en pantalla que nos ayudara a revisar el estado de GameObjects y con ello pudimos notar el problema con el seguimiento, se pudo conocer que los Objetos 3D se quedaban activos a pesar de no ser enfocados y cuando se regresaba al target al ya estar cargado no se reiniciaba su posición y en consecuencia salía en un posición que el dispositivo creía que había quedado la imagen después de salir de cámara, con frecuencia errónea, algo que se intento es implementar botones para “Forzar” el reinicio de los GameObject pero esta posición quedaba almacenada y este método sería difícil de implementar en el juego final. Indagando de cómo solucionar este problema no requería targets más elaborados, en cambio la solución es configurar debidamente el seguimiento por la mecánica del juego no era necesario guardar la posición de los targets por lo cual podemos mantener el seguimiento simple si está en cámara, active el GameObject de lo contrario desactívelo, Con la configuración echa tuvimos que probar moviendo el dispositivo suavemente y bruscamente, la configuración repara el problema de las posiciones ilógicas, cada vez la imagen no era visible inmediatamente el GameObject se desactivaba y al volver a activarlo, enfocar el target, este salía en la posición adecuada por más frenético que se moviera el dispositivo.

La siguiente tarea de la realidad aumentada era la creación de interacción entre GameObject para esto se debían de configurar con Colliders y RigidBody, en un principio al poner Colliders no se detectaban entre ellos ya que se requiero un RigidBody para poder detectarse, pero al implementar este último volvimos a tener el problema de las posiciones ilógicas, pero esta vez el problema era que el Objeto caía, en cuestión de segundos el GameObject aparecía y desaparecía, no era fácil de ver por lo rápido que sucedía pero se notó que este siempre iba para el suelo, resulta que al implementar el RigidBody se implementa un cuerpo rígido que se ve afectado por las físicas, que sería poder chocar con otros objetos y ser afectado por fuerzas como la gravedad, este parámetro de Unity funciona como una fuerza que atrae los cuerpos al suelo, como en la vida real, lo que ocurría en nuestro caso es que la gravedad afectaba al GameObject generado y al no tener un suelo preparado este caería infinitamente, por fortuna este parámetro es posible ajustarlo para que

tenga, mayor, menor o nula fuerza para nuestro caso optamos que no tuviera fuerza de tal forma que los objetos jamás cayeran al vacío.

Esta solución al problema de no ver el GameObject sin embargo había otros parámetros del Rigidbody a tener en cuenta, y es cuando se toca con otros objetos igual configurados, no solo probaría ser empujado además sería posible que rotara sobre sí mismo, esto se notó principalmente cuando se hacían pruebas con objetos esféricos, nuevamente por la configuración del Rigidbody, para ello se debe “anclar” o dejar fijos los parámetros de posición y rotación, es decir por más fuerzas que se ejecuten sobre el objeto este jamás se deja su posición y rotación, esto dejando algo obsoleto la solución anterior, sin embargo se dejó así, ya que de ese modo no hace tantos cálculos, otro detalle que nos dimos cuenta es que no es necesario que todos los objetos tengan un Rigidbody, la interacción depende de que el robot deba ir a los diferentes puntos por lo cual el único que debe tener un cuerpo sólido es el robot, los demás objetos pueden tener solo un Collider además deben estar configurados como Trigger, básicamente ajustar los parámetros para tener un área de detección y poder saber cuándo el robot está cerca de las canecas.

Como prueba final se movieron las imágenes con la mano de diferentes formas para comprobar si las físicas ya no aplican y pudimos presenciar que efectivamente no se presentan variedad en su posición ni en su rotación, además se configuró para tener solo dos GameObject, y se acercó el target del robot a los de la caneca está perfectamente podían detectar al robot sin afectar ningún otro parámetro, lo único es que se debía de acercar demasiado en algunos casos estaba tan cerca que algún target escondía el otro pero esto se solucionó fácil poniendo un gran tamaño al collider.

Las pruebas para verificar el funcionamiento del gameplay en la sección de realidad aumentada se realizaron principalmente con los targets en moviéndose en manos, lo primero es verificar si entre los GameObject se detectan para ello se dispuso a crear un texto que dijera “tocando: ” la última parte cambiaría en base a cuál es el nombre del objeto que lo está tocando con esto pudimos ver se requiere una detección de gran tamaño para interactuar sin problema, con la interacción debidamente hecha se debe probar la lógica del juego.

Se debió asegurar que el jugador pasara por las diferentes estaciones para terminar el juego para ello nos apoyamos en elementos gráficos, que son tres puntos presentes en la pantalla, estos serán rojos cuando el objetivo no se haya cumplido y verde cuando se cumpla, las pruebas se basaron en mover el robot por los diferentes targets y entre estos movimientos tratar de salir de la escena pero no era posible hasta que todos los puntos fueran de color verde, esta prueba se aplicó para las primeras versiones en versiones más actuales donde se contaba con un inventario y un portal.

Por parte del inventario se optó que el GameObject que no se destruyera fuera un canvas, es decir una capa de la pantalla donde se muestran las interfaces, esta indica la cantidad de residuos en ambos mundos, lo primero es verificar que la información se pase debidamente, se recogieron los residuos y paso varias veces a la realidad aumentada, siempre variando la cantidad, a veces 0 en clase, el inventario en la interfaz se pasaba sin problema.

A continuación se debía probar si al ejecutar la mecánica de desechar los residuos estos si se restaban del inventario original al principio no se daba debido a que se perdió a la referencia, entre el cambio de escena se destruía el GameObject referenciado y no era posible hacer la resta, para solucionarlo los objetos debían de buscar el inventario y crear la referencia con esto logrado se probó cómo funciona, se recogieron residuos en el mundo virtual, la prueba tenía que hacer residuo

por residuo, debíamos volver al mundo virtual mínimo tres veces, la primera con el inventario solo con residuos orgánicos, y los demás en cero, aprovechables y no aprovechables 0, la siguiente vez orgánicos 0, aprovechables mayor a 1 y no aprovechables 0, por ultimo solo con no aprovechables, esta prueba sirvió para ajustar debidamente la lógica, ya que algunos casos al oprimir el botón estando con orgánicos restaba en no aprobables por poner un ejemplo, esto debido a ciertos parámetros pasados por alto en un principio, y claro verificar que al regresar al mundo virtual el inventario resultara vacío, luego de un par de pruebas y ajustes quedo como se deseaba incluso quedo con una condición, si se entra con algún residuo en 0, aprovechables por ejemplo, obligatoriamente debe de pasar por la caneca de aprovechables, esto con el objetivo de que el jugador aproveche mejor las veces que entre en la realidad aumentada, que se prepare con muchos residuos para que valga más la pena.

5.1.3. VALIDACIÓN ROBOT

Cuando se nos entregó el almacén lo primero fue probarlo, no se tenía nada para poder manejarlo, era solo el almacén con los motores y llantas, pero por fortuna se contaba con un Arduino y una app de un trabajo anterior para realizar pruebas de movimiento sencillo y verificar el estado, las pruebas con la app se hicieron en un espacio cerrado y controlado, se comprobó como avanzaba hacia adelante, hacia atrás y como giraba esto sin ningún tipo de obstáculo, esto demostró que efectivamente los motores funcionaban pero uno tenía menos potencia que el otro y la comunicación era algo lenta pero como prueba inicial nos fue muy útil.

Lo siguiente fue la creación de una app de comunicación propia desarrollada en Unity, recurrimos a un par de librerías externas y gratuitas para poder hacer una comunicación sencilla entre la app y el módulo HC-05, la razón de porque este módulo es porque se quería recibir información de regreso, la primera app fue desarrollada esta tenía la tarea de enviar y recibir string de un Arduino, las pruebas consistían en enviar cualquier palabra y el Arduino regresarla, así podemos ver que recibió por el puerto serial de PC, con esta prueba pudimos apreciar varios parámetros a tener en cuenta a la hora de desarrollar el código como son los espacios en blanco y saltos de líneas que se presentan cuando se mandan string, con un par de comando se solucionó estos inconvenientes.

Con la comunicación echa se debe hacer el procesamiento de datos para poder mover el robot, lo primero fue definir un formato para poder trabajar con la primera app podemos enviar el formato de la forma más clara, de en un principio eran dos números con decimales que simularan el joystick, y el Arduino debía procesar la información, las primeras pruebas consistían repetir la primera prueba, movimientos sencillos, ir adelante, ir hacia atrás y girar, estas pruebas para demostrar que se logró la comunicación con la app y el Arduino funciona, pero teníamos los mismos problemas de antes, un motor con menos fuerza y comunicación lenta.

Lo primero fue tomar lectura de los voltios en los motores cuando se les pedía ir a máxima velocidad se puede ver que mientras en un o era de 5v en el otro era de 2,5 v, indagando de porque se da esto es porque la placa pueda trabajar con 4 motores en nuestro caso son solo dos, y se recomienda trabajarlo en X, realizamos el cambio, tanto físico como de código, y volvimos a probar enviando a los motores a su máxima velocidad, primero solo, solo el motor de la derecha a toda velocidad nos daba 5v, apagamos este y encendemos el otro, en el motor izquierdo leímos 5v, ahora se encienden los dos al mismo tiempo, y 5v en ambos, la nueva configuración arreglo el problema de la diferencia de potencia de los motores.

Para el problema de lo lento que sentía enviar información primero se implementó un cronómetro que permitiera saber en cuanto tiempo está listo el Arduino para procesar más información, básicamente la app envía el dato y espera a recibir un "OK" para enviar otro, este se implementó en un principio por el miedo de saturar la señal, ahora la app iniciara a contar desde que envía el dato y se detiene cuando llegue el "OK", con esto pudimos ver que el tiempo de respuesta era de aproximadamente 2.4 s, se teorizo que el problema podría ser que el Arduino hace la mayor parte del trabajo, se cambió su configuración y ahora el Arduino solo tenía el trabajo de leer datos, ya no tenía que transformar la información tipo joystick para los motores, la app se encargaría y el solo debe leer, no obstante no hubieron cambios significativos, además con la primera presentación pública del proyecto pudimos ver el corto alcance que se tiene al usar bluetooth.

Por lo cual se optó por cambiar el medio de comunicación por wifi, gracias al semillero de investigación pudimos tener un módulo ESP8266, la primera etapa consistía en mediante este módulo encender un led por un navegador web, a continuación, se vía una app creada en Unity, esta demostró ser más rápida e incluso con mayor alcance, ya que depende principalmente de una señal wifi.

Esto sin usar Arduino ya que con este teníamos la complicación de que la comunicación serial del EPS se da por 3,3v y la de Arduino 5v, lo que significa que al no tener cuidado podríamos quemar el módulo de wifi, en principio se intentó conseguir un adaptador pero esta tarea no fue posible, los datos no salían de este por lo cual se intentó que solo el EPS enviara datos al Arduino con el miedo de que la información al no tener misma corriente sea errónea pero estábamos equivocados el EPS puede enviar datos al Arduino no obstante no probamos viceversa por seguridad, la prueba era igual a la del Bluetooth un dato por la app y que Arduino lo dejara ver en pantalla, se demostró que el resultado era correcto, la comunicación se daba sin inconvenientes.

El método usado fue vía sockets un método que era comunicación directa mediante una IP en una red local incluso teníamos, sin programarlo un tiempo de respuesta que daba el mismo método de comunicación, demostrando que el sistema ahora respondía a aproximadamente 2 ms, una reducción de tiempo muy considerable respecto al módulo bluetooth, en principio al probarlo con el modem de casa no se presentó mayor inconveniente sin embargo a la hora de probarlo usando zona wifi, que es una función de los teléfonos inteligentes donde el teléfono funciona como modem se nos presentaban problemas graves de rendimiento, en un teléfono que servía tanto como modem como ejecutor del juego no sería fluido en la parte de la realidad aumentada cuando se movía el joystick, o cuando se mandaba señales al ESP.

Se probó usando otro teléfono como modem, el resultado era el mismo, sin embargo, cuando se probó con un teléfono aún más potente el resultado si fue fluido, el teléfono tenía una recepción de señal más fuerte lo que permitía que el sistema se ejecutara correctamente.

Por ultimo nos percatamos de la capacidad de las ruedas de oruga, por un lado las primeras pruebas que se hicieron que fueron en un piso liso no hubo mayor problema con el tiempo se fueron haciendo pruebas en diferentes suelos por ejemplo en uno más rugoso, lleno de piedritas o bien uno echo en ladrillo donde los obstáculos eran un copo más intensos sin embargo lograba avanzar sin mucho problemas, incluso puede subir obstáculos no muy altos pienses en tapetes o pequeños escalones, demostrando la capacidad de las ruedas tipo oruga para pasar por ciertos obstáculos con suma facilidad.

5.2. VALIDACIONES EXTERNAS

En los resultados del proyecto vemos cómo ha avanzado durante el tiempo empezando desde un escenario previamente diseñado con diferentes errores (bugs) y detalles que no eran los mejores dando errores de iluminación errores de interacción y demás que afectaba en la experiencia, también podemos observar la interacción temprana de la realidad aumentada con sus diferentes objetos virtuales 3D que eran simplemente una esfera y un cubo donde podíamos ver la capacidad de estos para tener comunicación entre ellos, aunque en tiempos tempranos tenían errores para ubicarse debido a la inexactitud y maltratamiento de los targets.



Figura 5-1 Pruebas de validación de la interacción de objetos en la RA

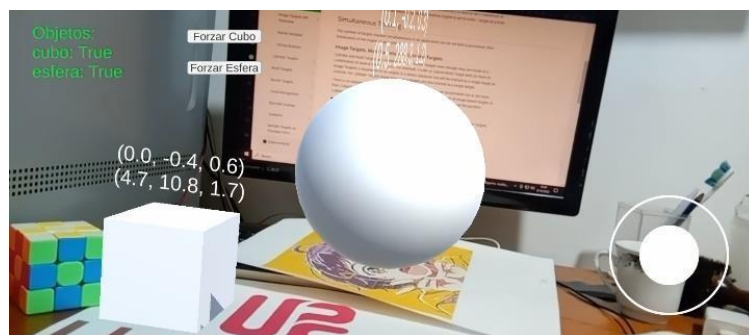


Figura 5-2 Pruebas de validación de la interacción de objetos en la RA

También pudimos observar cómo se dio a conocer el proyecto en sus primeras versiones en el taller mecatrónico de 2023, Figura 5-3 obteniendo grandes resultados que alimentaron el proyecto con críticas constructivas además que se observa que al público gustó tanto que se obtuvo la premiación a mejor proyecto de este evento.

En esta versión mostrada el proyecto contaba con dos aplicaciones, una para el manejo del robot y otra para la sección virtual. En la aplicación del robot observábamos una interfaz simple donde lográbamos controlar el robot, también con esta app podíamos tener la interacción de objetos con la realidad aumentada, donde al acercar la cámara del dispositivo móvil esta mostraba a los enemigos del juego, también mostrábamos a Azoni el robot protagonista en su versión virtual. En la

app de la sección virtual no se contaba con menús iniciales, ni música y tampoco control de SFX efectos de sonido y el brillo.

Contábamos con una versión temprana del juego donde ya teníamos el escenario, pero este tenía varios errores gráficos en diferentes puntos lo que hacía que objetos interactuaran de forma errónea con este, no contábamos con la facilidad actual del mapa, el agarre de los objetos era tosco y erróneo, y diferentes errores que se observaron ese día gracias a las personas presentes, de esta manera tuvimos una gran retroalimentación de parte de la audiencia presente en el evento.



Figura 5-3 Presentación en el Taller mecatrónico 2023 de la universidad Piloto de Colombia

Nuevamente este proyecto tuvo la oportunidad de participar en el taller mecatrónico 2024, Figura 5-4, con una de las últimas versiones del proyecto donde ya se resolvieron diferentes errores de iluminación aumento de detalles en los elementos del escenario la integración de la realidad aumentada junto a la sección virtual además de un funcionamiento del robot en tiempo real, además que se observa nuevamente que a las personas les sigue atrayendo el proyecto generando nuevos caminos para el este y seguir avanzando.



Figura 5-4 Presentación en el Taller mecatrónico 2024 de la universidad Piloto de Colombia

También se tuvo la oportunidad de llevar el proyecto al colegio San Patricio, Figura 5-5, donde se crearon encuestas para evaluar el conocimiento de las estudiantes presentes en este día donde también se generaron diferentes actividades como la rivalidad entre equipos para cumplir los diferentes objetivos del juego en un corto tiempo donde estos eran premiados, de esta manera logramos observar que el juego tenía la capacidad de divertir a un público joven atrayéndolos a que puedan aprender al reciclaje sin tener que ser forzados a ello, gracias a los buenos resultados de

este día fuimos invitados a seguir exponiendo y demostrando en diferentes colegios de Bogotá el proyecto Azoni AR, gracias a esto el proyecto podrá crecer aún más.



Figura 5-5 Presentación en el Colegio Sant Patric

Añadiendo al proyecto, se tiene la oportunidad de participar presentando una ponencia y un artículo del proyecto en el congreso internacional de ingeniería mecatrónica y automatización cima 2024 mostrando el proyecto a diferentes estudiantes y personas presentes en este congreso.



Figura 5-6 Foto con uno de los profesores presentadores del evento



Figura 5-7 Discurso de apertura de parte del doctor Jaime Duran



Figura 5-8 Logotipo de evento CIIMA 2024. Imagen tomada de *Red CIIMA* [Logotipo] RED CIIMA (2024) [27]

6. CONCLUSIONES

Se realizó una beta interactiva para mostrar los avances en el taller mecatrónico, lo primero fue preparar el robot, con una caja pudimos hacer una carcasa para ocultar los circuitos y más importante aún, tener una base para tener un target para que el programa identifique cual es el robot o jugador además se añadió que generará el mismo modelo del juego virtual así podemos estar seguros de que el target funciona bien y en consecuencia con el diseño del robot en realidad y el del virtual se crea un contraste muy interesante.

Con esto listo dispusimos a unir el programa de conexión con la cámara de vuforia, tuvimos que recurrir a los objetos persistentes que brinda unity para que a la hora de cambiar las escenas no se pierda la información.

Con la posibilidad ya de controlar el robot y poder verlo de otra forma restaba crear la verdadera interacción, ¿qué hará el jugador con el robot? Para esta beta se planeó algo simple, el jugador debe de buscar los tres objetivos y una vez los toque con el robot debe accionar un botón RA y así regresar al mundo virtual, en primer lugar aprovechamos la oportunidad para enseñar demás modelos del juego virtual, siendo que los enemigos son los tres objetivos, por problemas técnicos tuvimos que agregar indicadores que hicieran notar si ya se tocaron todos los objetivos y por otros problemas debimos poner un botón que apareciera solo cuando el robot tocara uno de los objetivos, este botón no vuelve aparecer si ya se oprimió una vez, algo parecido pasa con el botón en realidad aumentada, si el los objetivos no se puede ver el botón ni se puede interactuar, este fue el que nos dio más problemas con la beta, ya que a veces cuando se movía rápido la cámara el programa daba por hecho que se oprimía y salta a la pantalla final, no se presentó mucho el error sin embargo se planteó que al ver el target tuviera una cuenta atrás para poder activar el botón, esta solución aún no sea aplicado.

Como conclusiones del proyecto respecto a los objetivos planteados logrando así que el robot tenga la capacidad de desplazarse en un espacio controlado mientras es manejado por un dispositivo móvil también vemos la creación de escenarios con los elementos y bases de datos necesarios para la detección al momento de accionar el videojuego también se elabora un armazón sencillo para que el usuario pueda ver el modelo 3D de manera virtual de nuestro protagonista Ashley además de que este cuenta con un sistema de chasis tipo oruga también se elaboran diferentes escenas e interfaces como los menús dando funcionalidad al proyecto además de la inclusión de canciones originales que buscarán generar mayor interés en los jugadores por el proyecto también se valida el funcionamiento del robot en diferentes eventos y la salida al colegio San Patricio aportando al crecimiento del proyecto y dando visibilidad a que más entidades quieran ser parte de este.

7. REFERENCIAS BIBLIOGRÁFICAS

- [1] D. R. G. S. DR Daniel Sanchez Guzman, Diseño y desarrollo de una aplicación para dispositivos móviles de realidad aumentada, Ciudad de Mexico: Instituto Politecnico Nacional, 2016.
- [2] Azoni-AR, Artist, *Paleta de color*. [Art]. Universidad Piloto de Colombia, 2024.
- [3] Azoni-AR, Artist, *Logotipo*. [Art]. Universidad Piloto de Colombia, 2024.
- [4] Azoni AR, *Azoni: El robot Protagonista*, Bogotá DC, 2024.
- [5] Azoni TERB, *Modelo Caneca .fbx*, Bogotá, 2023.
- [6] A. Núñez, «Business insider,» *Torrón Stock*, 24 Febrero 2023. [En línea]. Available: <https://www.businessinsider.es/agua-caduca-pasa-bebes-esa-vieja-botella-1192800>. [Último acceso: 1 Julio 2024].
- [7] Pongsak14, «shutterstock,» *shutterstock*, 18 Enero 2016. [En línea]. Available: <https://www.shutterstock.com/es/image-photo/rusty-nail-on-crack-wood-364300412>. [Último acceso: 1 Julio 2024].
- [8] A. López, «20minutos,» *Henneo*, 13 Octubre 2014. [En línea]. Available: <https://blogs.20minutos.es/yaestaellistoquetodolosabe/de-donde-surge-que-una-manzana-podrida-estropea-al-resto-de-manzanas/>. [Último acceso: 1 Julio 2024].
- [9] A. AR, Dirección, *Azoni AR Escenario*. [Película]. Bogotá Colombia: Azoni AR, 2024.
- [10] N. H. Vargas, Compositor, *Azoni AR musica ambiente*. [Grabación de sonido]. Azoni AR. 2024.
- [11] Unity, «Unity Documentation,» *Unity*, 2024. [En línea]. Available: <https://docs.unity3d.com/es/530/Manual/LightingOverview.html>. [Último acceso: 1 Julio 2024].
- [12] GamerGarage, Dirección, *NavMesh Unity Tutorial - PRIMEROS PASOS*. [Película]. GamerGarage, 2020.
- [13] S. Wakabayashi, Dirección, *Wonder egg priority*. [Película]. Japón: CloverWorks, 2021.

- [14] Universidad Piloto de Colombia, «UniPiloto,» UniPiloto, 2024. [En línea]. Available: <https://www.unipiloto.edu.co/nuestra-nueva-imagen/>. [Último acceso: 1 Julio 2024].
- [15] G. Nelson, «Pinterest,» Gillian Nelson, 2024. [En línea]. Available: <https://ar.pinterest.com/pin/502362533417425513/>. [Último acceso: 1 Julio 2024].
- [16] Vuforia Engine, «Vuforia Engine developer portal,» Vuforia, 2024. [En línea]. Available: <https://developer.vuforia.com/library/objects/best-practices-designing-and-developing-image-based-targets>. [Último acceso: 1 Julio 2024].
- [17] Wolfram, «Wolfram,» Wolfram, 2024. [En línea]. Available: <https://reference.wolfram.com/language/ref/ImageChannels.html>. [Último acceso: 1 Julio 2024].
- [18] K. o. Hearts, «Wikimedia,» Wikimedia, 14 Agosto 2010. [En línea]. Available: https://commons.wikimedia.org/wiki/File:Taj_Mahal,_Agra,_India_edit2.jpg. [Último acceso: 1 Julio 2024].
- [19] Vuforia Engine, «Vuforia Engine developer portal,» Vuforia, 2024. [En línea]. Available: <https://developer.vuforia.com/library/getting-started/pose-status-and-status-info-unity>. [Último acceso: 1 Julio 2024].
- [20] Vuforia Engine, «Vuforia Engine developer portal,» Vuforia, 2024. [En línea]. Available: <https://developer.vuforia.com/library/objects/image-targets-optimization-techniques>. [Último acceso: 1 Julio 2024].
- [21] GIMP, «GIMP.org,» GIMP-ES, 2024. [En línea]. Available: <http://www.gimp.org.es>. [Último acceso: 1 Julio 2024].
- [22] Leonardo Interactive Pty Ltd, «Leonardo IA,» Leonardo Interactive Pty Ltd, 2022. [En línea]. Available: <https://app.leonardo.ai>. [Último acceso: 1 Julio 2024].
- [23] DFROBOT, «DFROBOT,» DFROBOT, 2024. [En línea]. Available: <https://www.dfrobot.com/product-1477.html>. [Último acceso: 1 Julio 2024].
- [24] MecaBot, «MecaBot,» Generico, 2024. [En línea]. Available: <https://mecabot.com/producto/modulo-bluetooth-hc05/>. [Último acceso: 1 Julio 2024].

- [25] Vistronica, «Mercado Libre,» Generica, 2024. [En línea]. Available: https://articulo.mercadolibre.com.co/MCO-452726375-modulo-bluetooth-esclavo-hc-06-_JM. [Último acceso: 1 Julio 2024].
- [26] Colombianizate, «Colombianizate,» Generico, 2024. [En línea]. Available: <https://www.colombianizate.com.co/tienda/arduino/wifi-bluetooth-rf/modulo-wifi-8266/>. [Último acceso: 1 Julio 2024].
- [27] RED CIIMA, «RED CIIMA,» RED CIIMA, 2024. [En línea]. Available: <https://www.redrima.org>. [Último acceso: 1 Julio 2024].