



**Universidad
Rey Juan Carlos**

**GRADO EN INGENIERÍA EN TECNOLOGÍAS DE LA
TELECOMUNICACIÓN**

Curso Académico 2020/2021

Trabajo Fin de Grado

**VIRTUAL REALITY EDITOR FOR VIRTUAL
REALITY SCENES**

Autor : Julián Ángel Pérez Muñoz

Tutor : Dr. Jesús María González Barahona

Trabajo Fin de Grado

Editor en Realidad Virtual para Escenas de Realidad Virtual

Autor : Julián Ángel Pérez Muñoz

Tutor : Dr. Jesús María González Barahona

La defensa del presente Proyecto Fin de Carrera se realizó el día de
de 202X, siendo calificada por el siguiente tribunal:

Presidente:

Secretario:

Vocal:

y habiendo obtenido la siguiente calificación:

Calificación:

Fuenlabrada, a de de 202X

*Dedicado a
mi familia / mi pareja / mis amigos*

Agradecimientos

En primer lugar, gracias a mi familia por haberme apoyado en todas las decisiones que ido tomando a lo largo de este tiempo, su ejemplo de esfuerzo, constancia y dedicación me ha hecho crecer y llegar hasta quien soy hoy.

En segundo lugar, agradecer a mi pareja y amigos, por esos momentos de risas y de enfados, por compartir nuestros lamentos y nuestros éxitos y por estar siempre ahí cuando lo he necesitado.

Por último, agradecer tanto a los docentes como a los compañeros que han compartido conmigo este camino ya que también han sido parte fundamental de este.

Resumen

Este proyecto desarrolla la creación de un editor en realidad virtual para escenas de realidad virtual, su objetivo principal es permitir crear objetos, y poder modificar sus características, para incluirlos en la escena estando inmersos en la propia escena de realidad virtual que se está creando.

Para la creación de los diferentes elementos de las escenas, se usa como tecnología básica A-Frame, un sistema que permite construir o manejar escenas de realidad virtual y utilizar técnicas de desarrollo típicas de la construcción de aplicaciones web en el navegador. A su vez, está construido sobre APIs disponibles en navegadores modernos como WebXR y WebGL.

Gracias a estas tecnologías la aplicación creada en este proyecto funciona con cualquier navegador que las soporte, ya sea en un ordenador tradicional, un dispositivo móvil o un dispositivo de realidad virtual.

Todo lo relacionado con este proyecto está disponible en su respectivo repositorio en GitHub.

Summary

The project is about creating a virtual reality editor for virtual reality scenes. Its main objective is to allow you to create objects, and to be able to modify their characteristics, to include them in the scene while immersed in the virtual reality scene itself that is being created.

For the creation of the different elements of the scenes, A-Frame is used as basic technology, a system that allows you to build or manage virtual reality scenes and use development techniques typical of web application production in the browser. In turn, it is built on APIs available in modern browsers such as WebXR and WebGL.

Thanks to these technologies, the application created in this project works with any browser that supports them, whether on a traditional computer, a mobile device or a virtual reality device.

Everything related to this project is available in your respective repository in GitHub.

Índice general

1. Introducción	1
1.1. Contexto	1
1.2. Objetivo general	2
1.3. Objetivos específicos	2
1.4. Estructura de la memoria	3
2. Estado del arte y trabajos relacionados	5
2.1. WebGL	5
2.2. WebXR	6
2.3. JavaScript	6
2.4. Three.js	7
2.5. HTML5	8
2.6. A-Frame	10
2.7. GitHub	12
2.8. PyCharm	12
2.9. LaTeX	13
2.10. glTF	13
2.11. Trabajos relacionados	14
3. Desarrollo del proyecto	15
3.1. Sprint 0	16
3.1.1. Objetivo	16
3.1.2. Desarrollo	17
3.1.3. Resultado	20

3.2. Sprint 1	20
3.2.1. Objetivo	20
3.2.2. Desarrollo	20
3.2.3. Resultado	23
3.3. Sprint 2	23
3.3.1. Objetivo	24
3.3.2. Desarrollo	24
3.3.3. Resultados	26
3.4. Sprint 3	26
3.4.1. Objetivo	27
3.4.2. Desarrollo	27
3.4.3. Resultado	30
3.5. Sprint Final	30
3.5.1. Objetivo	30
3.5.2. Desarrollo	30
3.5.3. Resultado	33
4. Resultados	35
4.1. Manual de usuario	35
4.2. Construcción de escenas	38
4.3. Implementación	39
4.4. Escenarios creados con la aplicación	43
5. Conclusiones	45
5.1. Aplicación de lo aprendido	46
5.2. Lecciones aprendidas	46
5.3. Trabajos futuros y esfuerzo dedicado	47
Bibliografía	49

Índice de figuras

2.1. Escena creada con Three.js	8
2.2. Visualización gráfica del DOM	9
2.3. Escena básica A-Frame	11
3.1. Estructura metodología Scrum	16
3.2. Inicio minijuego	17
3.3. Situación siguiente sección	18
3.4. Parte final minijuego	18
3.5. Minijuego completado	19
3.6. Inicio primer ejercicio	21
3.7. Escena inicial con mejora en la interfaz	21
3.8. Figura creada en la escena	22
3.9. Modelos 3D en la escena	23
3.10. Modo grupo desactivado	24
3.11. Modo grupo activado	24
3.12. Menú modificación figura	25
3.13. Nueva interfaz de la aplicación	26
3.14. Escenario general	27
3.15. Figura seleccionada	28
3.16. Menú modificación figura	28
3.17. Manejador	28
3.18. Nuevas figuras del editor	29
3.19. Nuevo menú de propiedades	31
3.20. Podium sin luz	31

3.21. Podium con luz	31
3.22. Escenario con manejadores	32
3.23. Escenario sin manejadores	32
3.24. Menú de ambientes	32
3.25. Página de inicio	33
3.26. Página de instrucciones	33
4.1. Inicio página del editor	36
4.2. Menú modificación glTFs	36
4.3. Botones para acceder a las funcionalidades.	37
4.4. Página Básica	38
4.5. Casa	43
4.6. Parque	43
4.7. Ovnis	44
4.8. Faro	44

Capítulo 1

Introducción

El fin de este proyecto es la construcción de un editor de escenas en realidad virtual que sea accesible y usable tanto desde un navegador web como desde otras herramientas creadas para ello, como pueden ser, las gafas de realidad virtual.

Hace un tiempo, para la creación de escenarios en realidad virtual se necesitaban hardwares muy específicos, pero hoy en día existen tecnologías que nos permiten crear este tipo de aplicaciones de una manera más o menos sencilla. Además podemos disfrutar de estas escenas en una gran cantidad de dispositivos, incluido un dispositivo móvil.

Las escenas creadas en este proyecto se han realizado basándose en el framework de creación de experiencias en realidad virtual, A-Frame. Una de las principales características de este, y por el cual lo he elegido, es la facilidad con la que se pueden construir dichas escenas ya que se pueden crear como si de un programa HTML se tratara y además, sin instalar ningún software adicional.

1.1. Contexto

Como he comentado, hace un tiempo, el uso y la creación de aplicaciones de realidad aumentada, virtual o mixta sin las herramientas adecuadas se hacía impensable, pero gracias al aumento de popularidad que se ha producido hoy en día de estas aplicaciones, es posible la creación de diferentes softwares que permiten, a parte de la creación en sí de la aplicación, la posibilidad de que estas sean cada vez más eficientes, accesibles y usuales.

Gracias a este interés, las compañías más importantes del mundo como son Google, Apple,

Facebook o Microsoft se han centrado en la compra de empresas pequeñas, que se basan en la creación de este software como pueden ser Oculus VR comprada por Facebook.

La manipulación de modelos 3D y la creación de distintas escenas es uno de los campos más interesantes de esta tecnología. Aplicaciones como Blender, AutoCAD o Spoke by Mozilla se centran en estas ideas pero ninguna de ellas trabajan en la propia realidad virtual, por ello le quise incorporar una innovación a nuestro editor de escenas, la posibilidad de trabajar en la propia realidad virtual.

Otros muchos sectores se aprovechan de esta tecnología, desde la creación de simuladores, videojuegos y museos hasta en medicina.

Si nos centramos en mi aplicación, esta también puede ser usada en distintos entornos, tanto profesionales, como pueden ser el mundo de la arquitectura o el diseño, hasta personales, como simple ocio.

Como consecuencia de todo lo anterior, podemos observar que el poder de esta tecnología es cada vez mayor pero se trata aún de una tecnología en desarrollo. Esta fue la razón principal por la que me decidí a empezar con este proyecto y contribuir en este largo camino.

1.2. Objetivo general

El objetivo principal de este proyecto es la creación de un editor en realidad virtual para escenas de realidad virtual, que funcione en un modo de escritorio mediante el navegador web y en diferentes herramientas de realidad virtual. Busco realizar este editor basándome principalmente en la tecnología A-Frame de una manera sencilla y con una interfaz de usuario agradable e intuitiva.

1.3. Objetivos específicos

A continuación, paso a especificar los distintos objetivos que tenemos para realizar este proyecto a los que más adelante daré solución.

- El editor de escenas debe tener una página principal.
- El editor de escenas debe tener una página de instrucciones.

- El editor de escenas debe tener la propia página donde poder crear y editar figuras en la escena.
- El editor contará con un menú desde el cual podrá crear diferentes figuras, pueden ser figuras básicas o glTFs.
- Todas estas figuras podrán ser modificadas en orientación, rotación, colores, y en material.
- El editor deberá contar con un botón que le permita acceder a la funcionalidad Modo grupo, que posibilita unir figuras para crear una sola como conjunto de varias, manejadas por un manejador.
- El usuario podrá esconder los manejadores mediante un botón en la escena.
- En el editor nos encontraremos un botón para cambiar a distintos ambientes.
- El usuario podrá moverse por el escenario, tanto en la versión web como en la de realidad aumentada.
- El código de la aplicación y el acceso a los diferentes ejemplos estarán disponibles en la plataforma GitHub.

1.4. Estructura de la memoria

En esta sección voy a describir la estructura de la memoria para una mejor comprensión sobre la misma.

- Primer capítulo. Introducción. En este apartado se realiza una breve descripción del proyecto añadiéndole un contexto y presentando los objetivos generales y específicos del trabajo.
- Segundo capítulo. Estado del arte y trabajos relacionados. Donde se describe las tecnologías que hemos usado para la correcta realización del proyecto.
- Tercer capítulo. Desarrollo del proyecto. Ponemos de manifiesto los diferentes procesos que hemos seguido para la realización de este, exponiendo tanto los problemas que han ido surgiendo como las soluciones utilizadas.

- Cuarto capítulo. Resultado final. En este capítulo creo un manual de usuario para ampliar el modo de uso de la aplicación, también describo de una manera más técnica los componentes que hemos utilizado para la realización de la misma.
- Quinto capítulo. Conclusiones. Analizaremos los objetivos que teníamos marcados y cómo hemos llegado hasta ellos, exponiendo problemas y soluciones.

Capítulo 2

Estado del arte y trabajos relacionados

En este proyecto se han utilizado diversas tecnologías, tanto para la creación del mismo como para su correcto funcionamiento. Las tecnologías son las siguientes:

2.1. WebGL

WebGL [8] es una tecnología propuesta por Vladimir Vukicevic¹, su trabajo comienza con la creación de un prototipo de OpenGL [2] para el elemento Canvas de HTML, llamado Canvas 3D.

Estaba originalmente basado en OpenGL ES 2.0, especificación creada para el iPhone y iPad, a medida que esta se iba desarrollando, su objetivo fue la usabilidad en distintos sistemas operativos y dispositivos. Utiliza el elemento Canvas ya que es una evolución del Canvas 3D y se accede a este mediante el DOM.²

Esta API³ de gráficos 3D basada en OpenGL permite a los navegadores modernos renderizar escenas 3D de una manera estándar y eficiente. Esta idea abrió un universo de posibilidades en las web basadas en entornos 3D como son los videojuegos, la visualización científica o las imágenes médicas.

Los programas WebGL están escritos en JavaScript y en Shading language, un lenguaje similar a C, C++. Por último, esta tecnología fue diseñada y es mantenida por el grupo non-profit Khronos Group.

¹https://en.wikipedia.org/wiki/Vladimir_Vukicevi

²https://en.wikipedia.org/wiki/Document_Object_Model

³https://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicacione

WebGL es usado por mi aplicación a la hora de la creación de las figuras, mediante el framework A-Frame, a través de la tecnología Three.js.

2.2. WebXR

WebXR [4], antes conocido como WebVR. Con el objetivo de acceder a las principales capacidades de los dispositivos tanto de realidad aumentada (AR), de realidad virtual(VR), como de realidad mixta. El nombre VR carecía de sentido, ya que en cuanto a los dispositivos que englobaba, las siglas VR se quedaban cortas. Otra diferencia importante sobre esta evolución es el soporte de controladores de entradas basado en la API gamepad⁴.

Esta tecnología permite manejar el proceso de renderización de las vistas que simulan la experiencia 3D y proporciona los datos necesarios para actualizar las imágenes mostradas al usuario. Se debe tener claro que no es una tecnología de renderizado, WebGL te ayuda con esto.

Los principales casos de uso de esta API son desde la visualización de objetos/datos hasta experiencias artísticas.

Esta tecnología es muy importante en el editor ya que permite la conexión entre las gafas de realidad virtual y la aplicación.

2.3. JavaScript

JavaScript [9] es el lenguaje de programación que se complementa con el lenguaje que más adelante explicaremos, HTML. Esta tecnología se suele utilizar para crear comportamientos dinámicos a la página web.

JavaScript es un lenguaje que permite la construcción de objetos basada en prototipos⁵, esto ayuda al programador a crear objetos, no mediante el uso de instancias de clases sino mediante la clonación de objetos. La sintaxis de JavaScript, con la intención de no complicar mucho este, es muy parecida a otros lenguajes como Java y C++.

⁴https://developer.mozilla.org/en-US/docs/Web/API/Gamepad_API/Using_the_Gamepad_API

⁵https://es.wikipedia.org/wiki/Programaci%C3%B3n_basada_en_prototipos

Centrándonos en las funcionalidades que ofrece este lenguaje, JavaScript nos permite crear los comportamientos de los distintos elementos HTML mediante el acceso al DOM. A continuación se puede ver un pequeño ejemplo de cómo realizar un hola mundo en JavaScript:

```
<script>
  document.write("Hola Mundo");
</script>
```

Gracias a este lenguaje, se puede crear contenido de actualización dinámica con la creación de eventos, controlar multimedia o animar imágenes, por lo que HTML deja de ser un lenguaje estático.

JavaScript se utiliza principalmente del lado del cliente como parte del navegador web. Si hablamos del lado del servidor podemos hablar de Node.js⁶.

Gracias a la creación de diferentes eventos y modificación de atributos que nos permite esta tecnología, se han podido crear todas las funcionalidades de la aplicación.

2.4. Three.js

Three.js⁷ fue creada por Ricardo Cabello en 2010 y hoy en día está alojado en GitHub.

Esta tecnología es una biblioteca y una API escrita en JavaScript cuyo objetivo principal es crear y visualizar figuras en 3D. Esta tecnología usa WebGL y es capaz de mostrar las figuras en el navegador. Esta biblioteca tiene diversas características como son efectos, escenas, animaciones, materiales o sombreados. También soporta la realidad aumentada o virtual apoyándose en WebXR.

A continuación se puede ver un ejemplo de un escena creada con Three.js

⁶<https://nodejs.org/es/>

⁷<https://threejs.org>



Figura 2.1: Escena creada con Three.js

Esta tecnología no solo sirve para la simple representación de las figuras en la escena sino que también es capaz de realizar ecuaciones matemáticas y crear matrices en la escena para dicha representación.

Gracias a Three.js se pueden construir las figuras de las escenas del editor, ya que A-Frame utiliza Three.js para dicha creación.

2.5. HTML5

HTML [6] es uno de los pilares fundamentales de este proyecto ya que es una de las tecnologías más usadas para la creación del mismo.

HTML es un lenguaje de marcado utilizado en el desarrollo del contenido web. Suele estar acompañado por otras tecnologías que la complementan, en cuanto a apariencia, CSS, no muy utilizada en este proyecto y en cuanto a funcionalidad, JavaScript, muy utilizada en este trabajo.

Una de las características de HTML es la utilización de marcas para etiquetar los distintos elementos del documento para luego mostrarlo en la web. Algunas de las etiquetas son:

<head>, , <p>, .

Estas etiquetas permiten al elemento tener una gran versatilidad, estructura lógica y facilidad a la hora de entenderlo. Dentro de estas podemos encontrar los atributos⁸ que son utilizados para controlar el comportamiento de dicha etiqueta.

Un ejemplo básico de un documento HTML puede ser el siguiente:

⁸https://es.wikipedia.org/wiki/Atributo_HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>TFG Julián</title>
  </head>
  <body>
    <p>Hola mundo!</p>
  </body>
</html>
```

Cuando hablamos de HTML es importante destacar el DOM, es la estructura de todos los elementos del documento organizados en nodos, el conjunto de todos estos se denomina árbol de nodos. Gracias al DOM y al lenguaje JavaScript podremos acceder a los elementos para una libre utilización de ellos. Esta estructura es guardada en memoria por los navegadores y de este modo poder mostrar la página web al usuario.

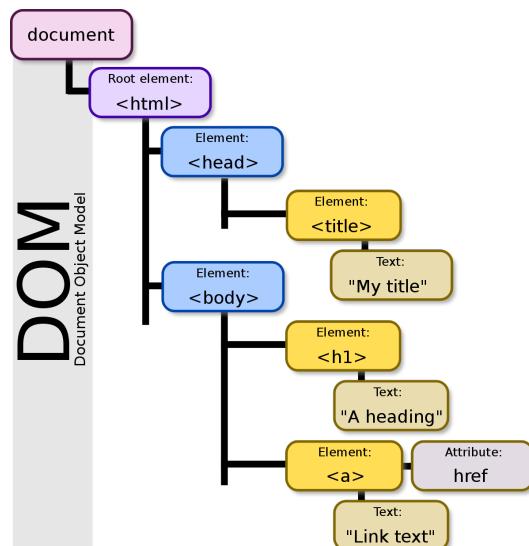


Figura 2.2: Visualización gráfica del DOM

Para terminar, destacar que ahora mismo HTML se encuentra en su versión HTML5, publicada en 2014, donde se añadieron varias nuevas funcionalidades como la inclusión de nuevas etiquetas o la compatibilidad con varias APIs con Canvas o WebGL.

HTML a parte de ser un lenguaje, es considerado como un estándar dedicado a la elaboración de páginas webs, como todo estándar sus normas están definidas en la web W3C⁹

En el editor de escenas, el framework, A-Frame, nos permite crear elementos mediante el lenguaje HTML.

2.6. A-Frame

Esta tecnología fue desarrollada por el equipo de Mozilla VR desde 2015. El objetivo principal de estos era permitir crear dichas escenas directamente en HTML sin conocer WebGL. A parte de esta, también usa otras tecnologías anteriormente explicadas como son WebXR y JavaScript.

A-Frame [1]es la tecnología más importante de este proyecto, ya que es la base fundamental de este. A-Frame es un framework creado para la construcción de experiencias en realidad virtual, basado en HTML, por lo que como hemos dicho antes, es un lenguaje bastante sencillo e intuitivo. Una de las características claves de A-Frame es el framework entidad-componente para Three.js.

Un ejemplo de creación de figuras en A-Frame mediante HTML puede ser el siguiente:

```
<html>
  <head>
    <script src="https://aframe.io/releases/1.2.0/aframe.min.js">
  </script>
  </head>
  <body>
    <a-scene>
      <a-box position="-1 0.5 -3"></a-box>
      <a-sphere radius="1.25"></a-sphere>
      <a-cylinder color="#FFC65D"></a-cylinder>
      <a-plane width="4" height="4"></a-plane>
      <a-sky color="#ECECEC"></a-sky>
    </a-scene>
  </body>
</html>
```

El resultado en la página web de este código, es el siguiente:

⁹<https://www.w3c.es>

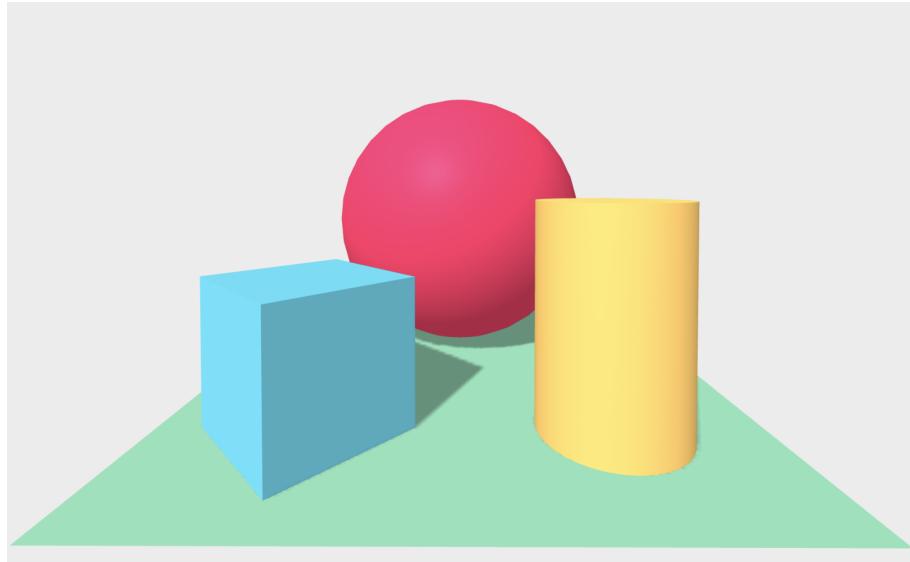


Figura 2.3: Escena básica A-Frame

Explicando un poco más sobre el código del escenario de ejemplo, destacaré:

- La inclusión de las dependencias de A-Frame en el elemento Head, esto nos permite usar todos los elementos de esta tecnología.
- La creación de las diferentes figuras mediante sus respectivas etiquetas.
- Modificación de ciertos aspectos de las figuras mediante los atributos.

Profundizando un poco más sobre la relación entidad-componente, una entidad es el objeto en sí creado mediante HTML y el componente es el comportamiento que le podemos asignar a dicha figura, estos se realizan en el JavaScript.

Estos componentes siguen una estructura estandarizada, algunos elementos de dicha estructura son: schema, el cual muestra las propiedades del componente e init, la parte del componente que se ejecuta al iniciar el componente. La manera correcta de incluir un componente a la escena es añadírselo al elemento como si de un atributo se tratara.

En mi aplicación, el uso de A-Frame ha sido fundamental, especialmente para la creación de figuras, la creación de la iluminación, la cámara del usuario y, para el comportamiento de estos.

2.7. GitHub

GitHub [7] es una plataforma de almacenamiento y administración de software, este sistema está basado en el control de versiones ¹⁰ y en Git ¹¹.

Gracias al control de versiones, el desarrollador puede administrar y llevar un registro de cualquier modificación sobre el proyecto. Este sistema permite al desarrollador trabajar de una forma segura mediante las ramas. Las ramas te permiten duplicar el código fuente, donde puedes hacer cambios sin afectar al proyecto "principal". Puedes fusionar las distintas ramas si lo deseas.

En cuanto a Git, se trata de un sistema de control de versiones, es decir, te ayuda a controlar lo anterior explicado pero mediante el terminal del ordenador.

GitHub es una interfaz gráfica de Git, el cual nos ofrece distintas funcionalidades como pueden ser, la creación de un usuario, la creación de distintos proyectos (repositorios) los cuales modificar desde la propia interfaz web, la capacidad de trabajar en repositorios creados por otros desarrolladores mediante un fork¹² o la modificación del mismo mediante un pull, creación de un foro en los que los desarrolladores pueden expresar dudas/problemas sobre el proyecto, etc.

Hoy en día según los desarrolladores el 87 % de estos utilizan GitHub.

Para el desarrollo de mi aplicación GitHub ha sido la plataforma donde he alojado el código del mismo. También para el despliegue de esta, he utilizado GitHub pages, una funcionalidad de la página que te permite desplegar tus aplicaciones.

2.8. PyCharm

Se trata de un entorno de desarrollo integrado (IDE) que se utiliza para la programación, es un software creado por la compañía JetBrains [3].

Este software es multiplataforma adaptado para Windows, macOS y Linux. Ha sido utilizado para la creación de este proyecto y por eso es una parte fundamental de este.

Tiene infinidad de ventajas entre las que están:

- Asistencia y análisis de codificación, con ayuda a la finalización del código, sintaxis y

¹⁰https://es.wikipedia.org/wiki/Control_de_versiones

¹¹<https://git-scm.com>

¹²[https://es.wikipedia.org/wiki/Bifurcaci%C3%B3n_\(desarrollo_de_software\)](https://es.wikipedia.org/wiki/Bifurcaci%C3%B3n_(desarrollo_de_software))

resaltado de errores.

- Navegación entre proyectos y ficheros.
- Acceso a la ejecución del código en el navegador mediante accesos directos.

PyCharm proporciona una API para que los desarrolladores puedan crear distintos plugins y de este modo se pueda crear código de manera más fácil y eficiente.

Ha sido la herramienta utilizada para la creación del código del editor.

2.9. LaTeX

LaTeX está basado en Tex¹³, programa destinado a la creación de documentos que contienen textos y fórmulas matemáticas. Este programa no es un editor de textos sino un procesador de macros.

LaTeX es un conjunto de macros para Text, cuyo objetivo principal es la alta composición tipográfica. Está orientando a la producción y creación de documentos científicos y se ha estandarizado para la creación de los mismos.

Esta tecnología nos presenta diversas ventajas entre las que se encuentran la facilidad de gestionar las referencias, bibliografías, etc. Es un sistema multiplataforma, puedes componer fórmulas con calidad de imprenta o la capacidad de exportar el documento a PDF. La creación de dichos documentos puede ser al principio, un poco complicada si no se conoce la herramienta.

Se trata de un software libre bajo la licencia LPPL¹⁴

Este sistema ha sido fundamental a la hora de crear este proyecto ya que esta memoria está creada utilizando LaTeX.

2.10. glTF

Este formato de archivo tiene como objetivo principal almacenar escenas y modelos en 3D y está basado en el formato de texto sencillo para el intercambio de datos, JSON. Popularmente se le describe como como el JPEG en 3D.

¹³<https://es.wikipedia.org/wiki/TeX>

¹⁴https://es.wikipedia.org/wiki/LaTeX_Project_Public_License

Este formato está creado para una eficiente distribución de objetos y escenas en 3D, disminuyendo el tiempo de procesamiento en ejecución. Este formato se puede usar en aplicaciones que usan la tecnología anteriormente explicada, WebGL.

Es importante explicar este formato, ya que la aplicación que he creado permite el manejo y modificación de este formato en la aplicación.

2.11. Trabajos relacionados

Cabe destacar algunos editores que han servido como inspiración para la creación de mi editor como son Blender, centrado en aspectos como la renderización o la creación de gráficos en 3D o Unity centrado en creación del motor gráfico en videojuegos.

Capítulo 3

Desarrollo del proyecto

Centrándonos en el desarrollo del proyecto debemos comenzar explicando la metodología SCRUM [5] ya que hemos seguido una metodología muy similar para la creación de este proyecto.

Está metodología, oficialmente surgió en 1995 presentada por Ken Schwaber y Jeff Sutherland mediante "SCRUM Development Process". Aunque esta fuera la fecha oficial, este modelo ya se había dejado ver en los años 80.

SCRUM es un metodología que está centrada en el desarrollo ágil¹ de software, aunque he realizado el proyecto de manera individual esta metodología se suele aplicar de manera colectiva con la participación en equipo con el objetivo de obtener el mejor resultado posible.

En SCRUM se pueden diferenciar distintos roles, SCRUM Master, Product Owner y el equipo. El rol de SCRUM Master en este caso está representado por el tutor del proyecto cuya función es facilitar y ayudar a gestionar el trabajo. En este caso el Product Owner no tiene un papel importante y respecto al equipo está representado por mí, el alumno que realiza este proyecto, y su función es el desarrollo y la entrega de las distintas tareas organizadas en cada sprint.

Durante los sprints el equipo, en este caso yo, me centraré en la realización de las tareas marcadas. Estos sprint se inician con un reunión SCRUM Master - equipo en la cual se fijaba unos objetivos y se marcaban los pasos a seguir para la consecución de dichos objetivos. Tras un tiempo trabajando en el sprint se emplaza otra nueva reunión en la cual se comparte la tarea terminada y los problemas que han ido surgiendo durante esta. En esta misma reunión el equipo

¹https://es.ryte.com/wiki/Desarrollo_gil_de_Software

prepara un versión funcional de la aplicación o tarea y se marcan los siguientes pasos a seguir.

Se puede ver más en detalle el proceso en la siguiente imagen:

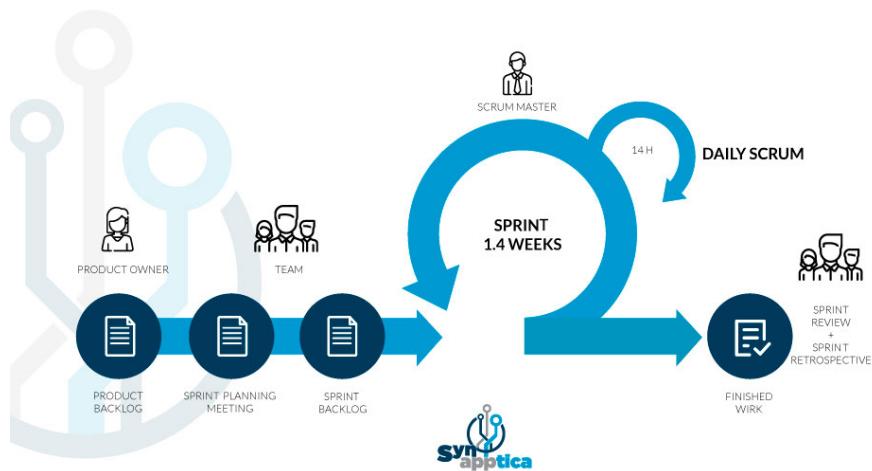


Figura 3.1: Estructura metodología Scrum

3.1. Sprint 0

En este primer sprint el SCRUM Master comenzó a introducir las principales tecnologías que iban a ser usadas en el proyecto, A-Frame, JavaScript y HTML. Por lo que en este sprint me dediqué a realizar una actividad que asentará la unión entre ambas tecnologías.

3.1.1. Objetivo

El objetivo principal de este Sprint es empezar a usar A-Frame para poder así introducirme en la tecnología y aprender cómo utilizarla. La creación de distintas entidades, eventos, componentes y físicas son los principales objetivos. En este sprint tenemos que tener en cuenta que una de las principales tecnologías utilizadas, JavaScript, era nueva para mí, por lo que también tuve que aprender sobre esta. Para consolidar todo lo anterior la tarea a realizar fue la creación de un minijuego en el que crear un componente y que este, reaccionara a un evento.

3.1.2. Desarrollo

Todo comienza haciéndose una idea sobre qué va a tratar el minijuego, qué componente vamos a crear y qué comportamiento le vamos a dar.

La base fundamental del minijuego será un recorrido en el cual debes adivinar una de las tres puertas que hay en las diferentes secciones para así, poder avanzar y llegar al objetivo final del juego.

Para el comportamiento de choque contra la pared, que tendrás que realizar para pasar a la siguiente sección, utilizaremos el componente Kinema.js que nos proporciona las físicas necesarias para esto.

El recorrido creado es muy sencillo y presenta las siguientes fases:

- Cuando entramos en el juego nos encontramos con la primera sección de tres puertas a adivinar, además, si giramos la cámara nos encontramos con las instrucciones del juego.



Figura 3.2: Inicio minijuego

- Tras esto nos encontramos con tres secciones en las cuales debemos adivinar la puerta para poder avanzar a la siguiente. La manera que tenemos de pasar a la siguiente sección es empujando la puerta, si es la puerta correcta, dicha puerta caerá y podrás pasar a la siguiente.

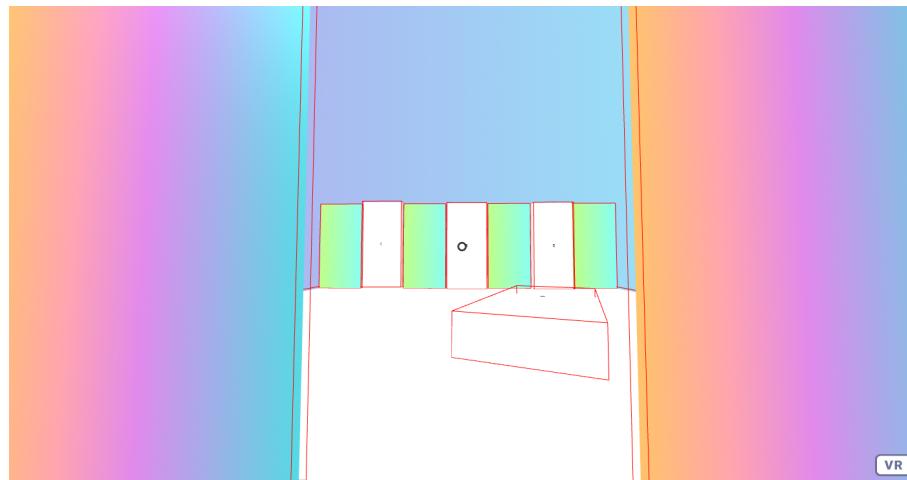


Figura 3.3: Situación siguiente sección

- Por último, tras adivinar todas las puertas, nos encontramos con el objetivo principal del juego, pulsar el botón de la victoria, al cual solo se puede llegar atravesando las puertas. Tras pulsar este botón, el texto que se encuentra en la parte superior te indicará que has ganado.

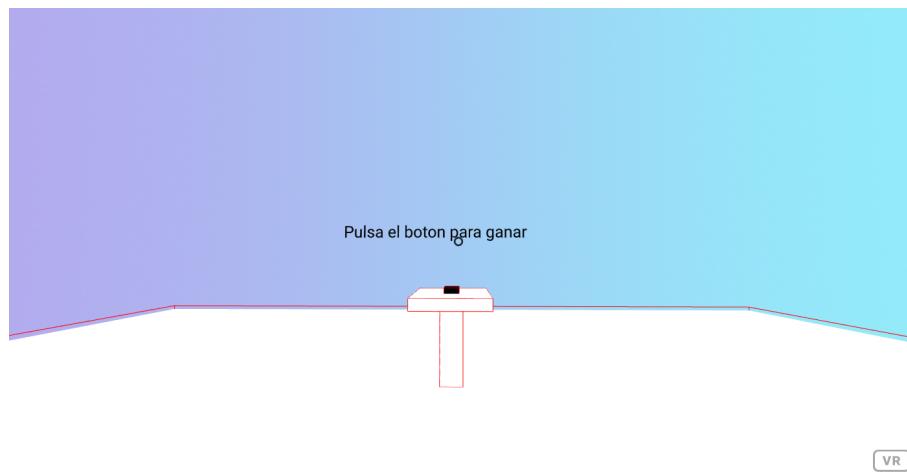


Figura 3.4: Parte final minijuego

En cuanto al código creado para este minijuego, cabe destacar lo siguiente:

```
<script src="//cdn.rawgit.com/donmccurdy/aframe-physics-system/
v4.0.1/dist/aframe-physics-system.min.js"></script>
<script src="//cdn.rawgit.com/donmccurdy/aframe-extras/
v5.0.0/dist/aframe-extras.min.js"></script>
<script src="components/kinema.js"></script>
```

Estas líneas de código localizadas en el elemento

```
<head>
```

son componentes que nos permiten utilizar distintas funcionalidades que aplicaremos en nuestro minijuego, como pueden ser la creación de distintas entidades o la utilización de físicas.

Otra línea de código interesante es:

```
<a-box dynamic-body><a-text></a-text></a-box>
<a-box static-body><a-text></a-text></a-box>
```

Esta parte de código trata de una parte de la creación de las puertas que tienes que elegir, estas, utilizan un componente que proporciona el comportamiento de una entidad fija o móvil.

`dynamic-body/static-body`

Por último, cabe destacar un sencillo componente creado por mí, cuya función es el cambio del texto al pulsar en una entidad. Este texto muestra al usuario que ha conseguido el objetivo del minijuego. Este era realmente el objetivo del sprint, la creación de una entidad que reaccionara a un evento.

La estructura de un componente es la siguiente:

```
AFRAME.registerComponent('nowinnertowinner', {
  schema: {
    },
  init: function () {
    },
  },
```



Figura 3.5: Minijuego completado

3.1.3. Resultado

En este primer sprint aprendí las posibilidades que nos puede dar a-Frame y a hacerme una idea de cómo plantear el resto del proyecto. De manera paralela fui aprendiendo a utilizar JavaScript ya que era una tecnología muy poco antes usada. Este minijuego se encuentra en el repositorio del proyecto y está accesible a cualquier persona, tanto a la interfaz del juego² como al código³. Tras esto, se mantuvo la correspondiente reunión con el SCRUM Master en la cual se pone en común lo aprendido y se plantea el siguiente sprint.

3.2. Sprint 1

A partir de este sprint se empezará a construir nuestra aplicación final por lo que podemos decir que este sprint es la base de nuestro proyecto.

3.2.1. Objetivo

El objetivo de este sprint es la creación de un menú con las cuatro formas básicas de A-Frame (plano, cubo, cilindro y esfera). Las cuales podremos seleccionar y aparecerán en un lugar determinado de la escena. Para la creación de estas acciones utilizaremos tanto elementos HTML como componentes creados en JavaScript.

3.2.2. Desarrollo

Algunos sprints los separaremos en tareas para así organizar mejor este y poder así trabajar lo más eficiente posible.

Tarea 1

En esta primera tarea crearé un menú con cuatro figuras básicas. El modo de interactuar con dichas figuras, es mediante el cursor clicando encima de ellas. De este modo las figuras aparecerán en el escenario. El resultado de esta tarea se puede visualizar en la figura 3.6

²<https://julianperezm.github.io/A-frame/FirstExercise/FirstExercise.html>

³<https://github.com/julianperezm/A-frame/tree/master/FirstExercise>

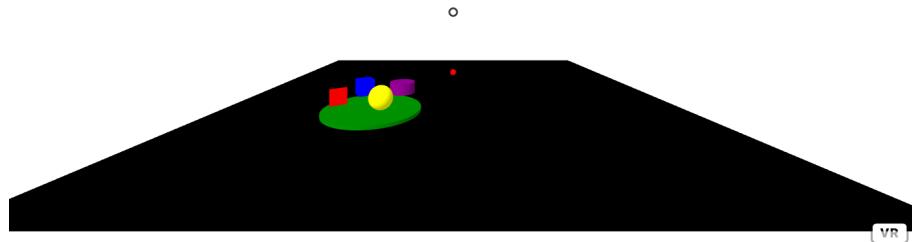


Figura 3.6: Inicio primer ejercicio

Tarea 2

En nuestro camino hacia el objetivo añadiré movimiento al menú para hacerlo más accesible al usuario, utilizaremos el atributo de animación para la creación de la tarea.

```
baseMenu.setAttribute('animation', 'property:rotation;to:0 360 0;  
loop:true;dur:10000')
```

Tarea 3

En esta tarea seguiré con la funcionalidad de crear una interfaz más amigable y pulsar una figura con el cursor de A-Frame no lo era. Por lo que creamos una interacción con la figura clicando directamente en ellas. A su vez también cambiamos el aspecto del escenario para que fuera más vistoso al usuario. Como se puede ver en la figura 3.7.

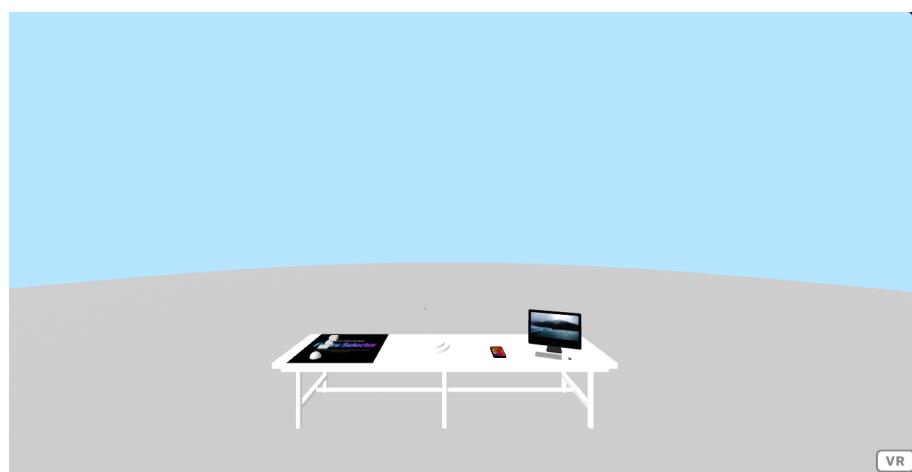


Figura 3.7: Escena inicial con mejora en la interfaz

Tarea 4

Para seguir adelante con la creación del editor, en esta tarea nos centraremos en la funcionalidad, la cual permitirá que las figuras puedan ser arrastradas a cualquier lugar de la escena y que cada una tenga una mini previsualización de distintos colores encima de ella.

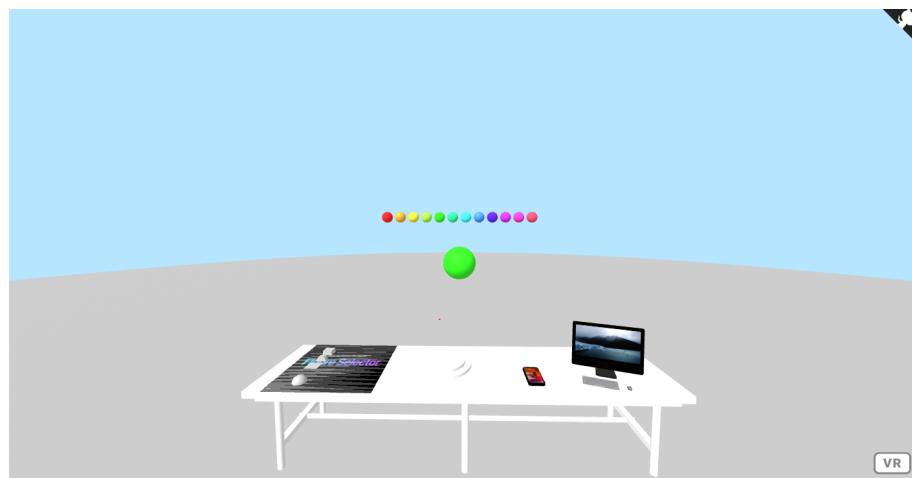


Figura 3.8: Figura creada en la escena

Tarea 5

Por último en esta tarea se empieza a jugar con la creación de modelos en 3D con el formato glTF explicado anteriormente. Insertamos en la escena un dispositivo móvil y un ordenador a modo de decoración, como se observa en la figura 3.9.

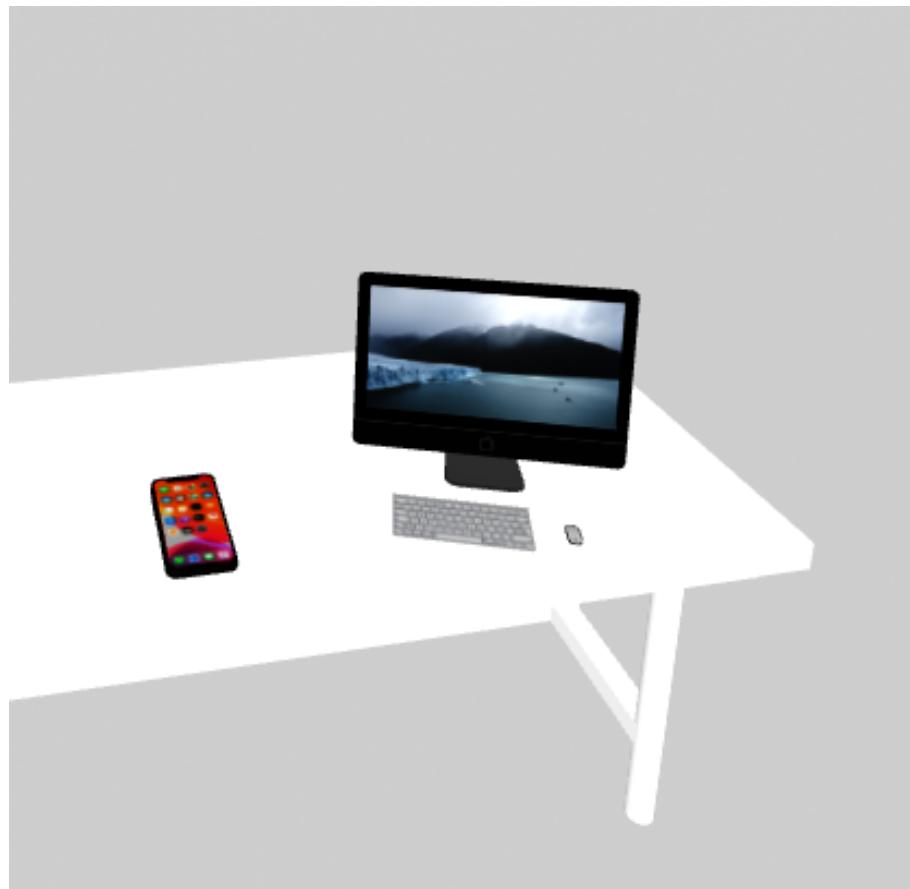


Figura 3.9: Modelos 3D en la escena

3.2.3. Resultado

En este sprint se dio el primer paso de lo que será nuestro editor, se comienza con la creación de algunas de las funcionalidades finales de la aplicación. Toda esta implementación se encuentra en el repositorio del proyecto en GitHub, tanto al código ⁴ HTML o JavaScript como a la escena en el navegador⁵

3.3. Sprint 2

En este sprint tras haber empezado este proyecto, me centraré en distintas funcionalidades y la mejora de la interfaz de la aplicación.

⁴<https://github.com/julianperezm/A-frame/tree/master/ThirdExercise>

⁵<https://julianperezm.github.io/A-frame/ThirdExercise/ThirdExercise.html>

3.3.1. Objetivo

Me centraré en tres objetivos principales,

- La modificación de una propiedad concreta de las figuras, el tamaño.
- La creación de un modo grupo de figuras, una de las funcionalidades principales de la aplicación.
- La mejora de la interfaz de la aplicación para que sea más usable en 3D.

3.3.2. Desarrollo

Como en los casos anteriores separaremos el desarrollo del sprint en diferentes tareas.

Tarea 1

En esta tarea me centraré en la creación de la funcionalidad que llamaré modo grupo. Esta funcionalidad consiste en la unión de diferentes figuras que serán manejadas por otra que llamaremos, manejador.

Esta funcionalidad es útil por si queremos crear una figura que se base en la unión de varias. El método por el que realizaremos esta funcionalidad es mediante la creación de distintos componentes. La manera de acceder a ella es pulsando la tecla q.

El resultado de esta tarea se puede observar en un panel creado en el escenario, cuando el modo grupo esté activado se mostrará como se puede observar en la figura 3.11, del mismo modo cuando esté activado figura 3.10.

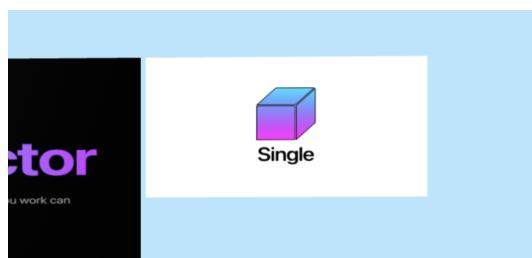


Figura 3.10: Modo grupo desactivado

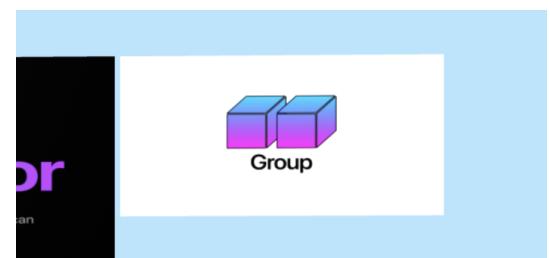


Figura 3.11: Modo grupo activado

Tarea 2

Tras la creación de la funcionalidad, me centraré en la modificación de una de las propiedades de la figura. La modificación se centra en el tamaño, recordemos que anteriormente añadimos el poder cambiar el color de la figura.

Esto lo realizaré de una manera similar, crear distintos componentes que permitan modificar la escala de la figura, con una particularidad, podremos modificar la escala en los distintos ejes para así poder tener más posibilidades a la hora de crear diferentes figuras.

Esta tarea se puede visualizar en la escena mediante diferentes botones que se pueden ver en la figura 3.12, además de esto, podrás ver las medidas de la figura creada.

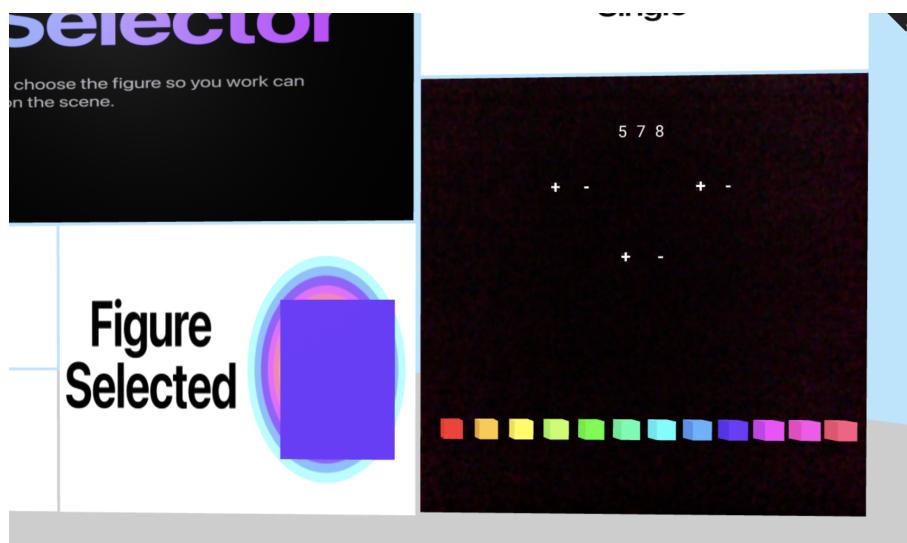


Figura 3.12: Menú modificación figura

Tarea 3

En esta tarea nos centraremos en la mejora de la interfaz de la aplicación para que sea más usable y más vistosa para el usuario.

A los paneles de información de la aplicación, en lugar de ser planos, se les dará un poco de profundidad. El menú de selección de figuras pasará a ser en 3D para que sea más intuitivo, para realizar este cambio se modificó el atributo de tamaño de todos los elementos del menú añadiéndoles profundidad.

El resultado se puede observar en la figura 3.13

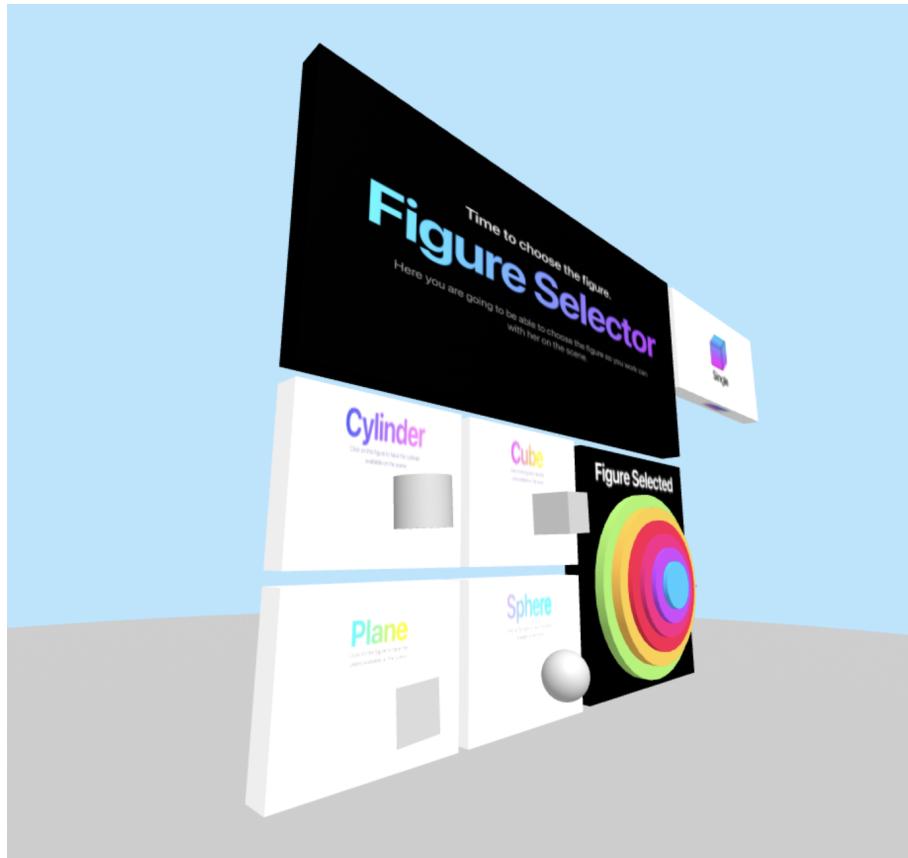


Figura 3.13: Nueva interfaz de la aplicación

3.3.3. Resultados

En este sprint hemos conseguido mejorar considerablemente nuestro editor, ya que hemos añadido una nueva funcionalidad, hemos añadido un método más de modificación de la figura y hemos mejorado el escenario para un uso más fácil de este. Este sprint está disponible en el repositorio del proyecto en GitHub, tanto el código⁶ como el escenario⁷.

3.4. Sprint 3

Tras el sprint anterior en este seguiré mejorando la interfaz de usuario de la aplicación además de añadirle nuevas funcionalidades como la creación de nuevas figuras y la modificación

⁶<https://github.com/julianperezm/A-frame/tree/master/FifthExerciseV1>

⁷<https://julianperezm.github.io/A-frame/FifthExerciseV1/FifthExercise.html>

de nuevos atributos de estas.

3.4.1. Objetivo

Para este sprint tenemos varios objetivos en los que centrarnos:

- Mejora de la interfaz, me centraré en distintos puntos como son las dimensiones del escenario y la visibilidad de distintos objetos.
- Añadiré la funcionalidad de poder añadir nuevas figuras con el formato glTF.
- Compatibilizar la aplicación con las gafas de realidad aumentada.

3.4.2. Desarrollo

Las tareas para este sprint son las siguientes:

Tarea 1

En esta tarea me centraré en la mejora de la interfaz de usuario. Rediseñaré el escenario, respecto del tamaño, lo haremos más pequeño, ya que anteriormente las figuras no tenían un tamaño acorde al escenario.



Figura 3.14: Escenario general

Las figuras que sean clicables se verán más claras que se pueden clicar.

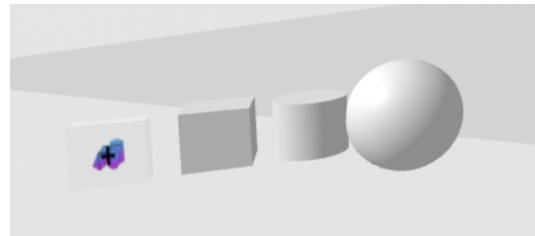


Figura 3.15: Figura seleccionada

Respecto al menú de creación de figuras y el modo grupo, haré que sigan al usuario para que se encuentre de una manera más accesible y escondida y de este modo, visualizar mejor el escenario. Respecto al menú de las figuras con ayuda de la creación de componentes, lo rediseñaremos para que se muestre de una manera más sencilla y vistosa y le añadiremos la posibilidad de poder esconderlo. También colocaré de una manera más adecuada el menú de modificación de la figura.



Figura 3.16: Menú modificación figura

Respecto al modo grupo cambiaré varias cosas, la primera, el manejador, ahora tiene una imagen especial para así poder diferenciarlo mejor.

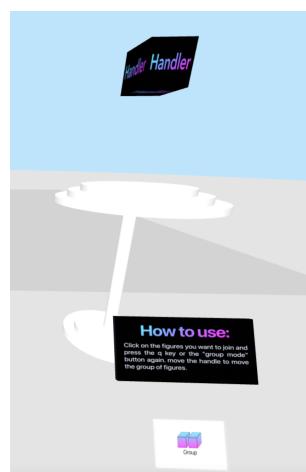


Figura 3.17: Manejador

Y respecto de las figuras seleccionadas, ahora tendrán una animación al ser clicadas.

Tarea 2

En esta tarea, me centraré en añadir distintas figuras en 3D.

Crearé un menú para estas figuras en las cuales podremos elegir entre distintas figuras las cuales están clasificadas por categorías. Para añadir estas figuras he utilizado el elemento

```
<a-asset-item>
```

que nos facilita A-Frame.

El resultado lo podremos observar en el escenario de la manera que nos muestra la figura 3.18



Figura 3.18: Nuevas figuras del editor

Tarea 3

En esta tarea añadiré la posibilidad de modificar más propiedades de las figuras.

Con la creación de nuevos componentes se añadió la capacidad de modificar la orientación de la figura en los tres ejes, la rugosidad, la metalización y la opacidad, a esta última se le añade una funcionalidad especial la cual puede hacer la figura transparente.

El resultado se puede observar en la figura 3.16

Tarea 4

Por último, me centraré en la compatibilidad de nuestro escenario con la gafas de realidad virtual, en concreto, las Oculus Quest.

3.4.3. Resultado

Como resultado de este sprint, puedo decir que estoy más cerca de la versión final. Solo nos quedaría refinar algunos aspectos. He conseguido crear nuevas funcionalidades, mejorar nuestra aplicación y hacerla más usable para el usuario. Este sprint está disponible en el repositorio del proyecto, tanto el código ⁸ como el escenario, ya sea la versión escritorio ⁹ o la versión de realidad virtual ¹⁰.

3.5. Sprint Final

En este sprint me centraré en el final del proyecto puliendo los últimos detalles y añadiendo las últimas funcionalidades de la aplicación. Acercándolo lo más posible a un editor usable y también lo más atractivo posible.

3.5.1. Objetivo

Los objetivos de este sprint serán:

- Seguir mejorando la interfaz de usuario haciéndolo más intuitivo.
- Añadir nuevos ambientes al editor.

3.5.2. Desarrollo

Seguiremos dividiendo las el desarrollo del sprint en tareas.

Tarea 1

En este tarea seguiré mejorando la interfaz de usuario para hacerla más intuitiva y sencilla para el usuario, añadiremos nuevos atributos, como es el porcentaje de progreso de las modificaciones de los materiales que se les pueden realizar a las figuras.

⁸<https://github.com/julianperezm/A-frame/tree/master/SeventhExercise>

⁹<https://julianperezm.github.io/A-frame/SeventhExercise/Desktop.html>

¹⁰<https://julianperezm.github.io/A-frame/SeventhExercise/Glasses.html>

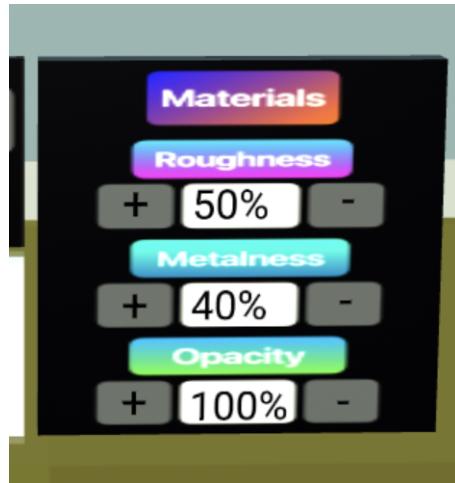


Figura 3.19: Nuevo menú de propiedades

También añadiré una luz en el pódium para que las modificaciones que le hacemos a la figura se puedan visualizar de una manera más clara.

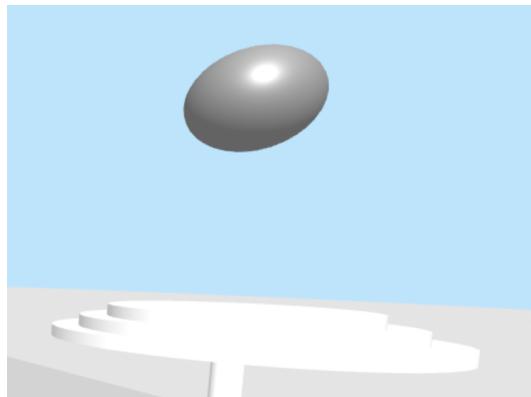


Figura 3.20: Podium sin luz

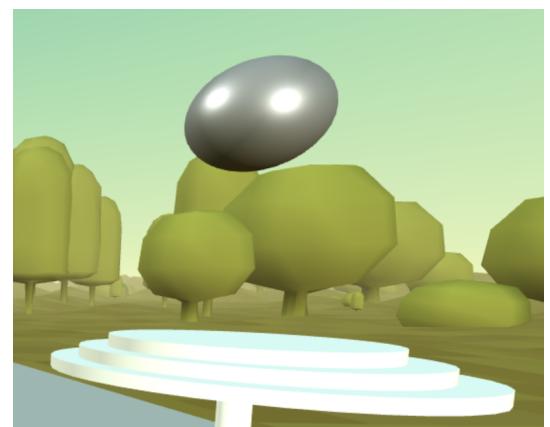


Figura 3.21: Podium con luz

Tarea 2

En esta tarea me centraré en mejorar la funcionalidad de modo grupo, añadiremos un botón para así poder eliminar el manejador y que el escenario quede mucho más limpio.

Además de esta mejora añadiremos una funcionalidad la cual te permita cambiar de ambiente al escenario.

La eliminación de los manejadores es posible añadiendo un componente que añada opacidad a dichas figuras, el escenario con manejadores quedaría como muestra la figura 3.22 y sin manejadores como la figura 3.23.



Figura 3.22: Escenario con manejadores



Figura 3.23: Escenario sin manejadores

Para los cambios de ambientes utilizaré un componente creado por la comunidad, se puede encontrar en su repositorio de GitHub¹¹.

El resultado de esta parte de la tarea se puede visualizar en la escena con un desplegable en el que puedes elegir que ambiente usar, como se muestra en la figura 3.24.

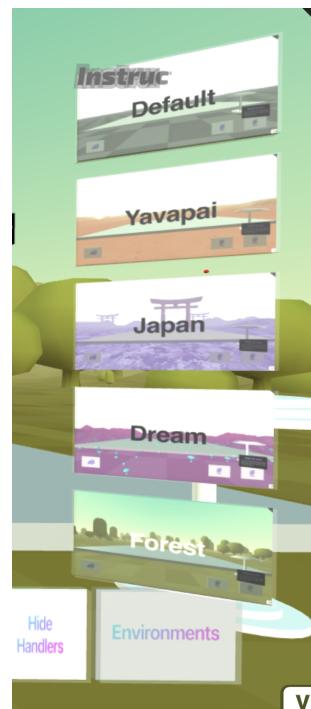


Figura 3.24: Menú de ambientes

Tarea 3

Esta última tarea tratará de terminar la aplicación añadiéndole una página principal,

¹¹<https://github.com/supermedium/aframe-environment-component>



Figura 3.25: Página de inicio

y otra página de instrucciones.

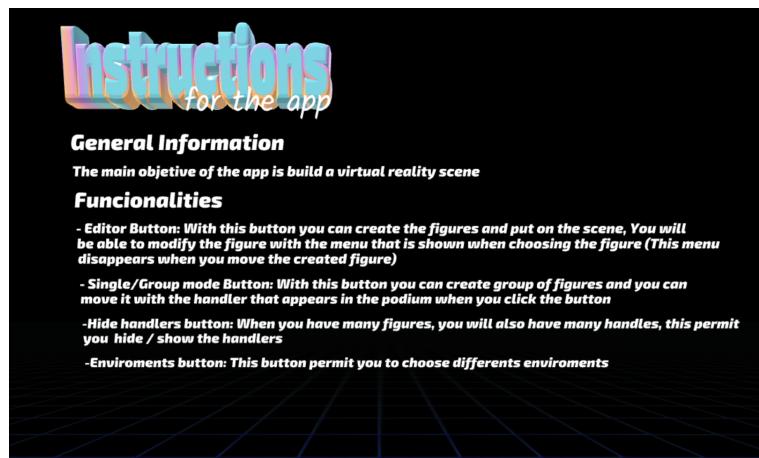


Figura 3.26: Página de instrucciones

Ambas son accesibles entre ellas además del escenario.

3.5.3. Resultado

En este último sprint he conseguido terminar el proyecto, este proyecto puede seguir desarrollándose y mejorando, puede ser el comienzo de algo grande. El código ¹² y el escenario, ya sea la versión escritorio ¹³ o la versión compatible con las gafas ¹⁴ están disponibles en el repositorio del proyecto en GitHub.

¹²<https://github.com/julianperezm/A-frame/tree/master/Home>

¹³<https://julianperezm.github.io/A-frame/Home/HomeDesktop.html>

¹⁴<https://julianperezm.github.io/A-frame/Home/HomeGlasses.html>

Capítulo 4

Resultados

En esta sección incluiré el manual de usuario de la aplicación donde se mostrará desde un punto de vista del usuario y de manera sencilla cómo utilizar la aplicación y la arquitectura en la que se basa el proyecto, donde explicaremos la estructura y los componentes usados para la creación de la aplicación.

4.1. Manual de usuario

Existen dos tipos de interacción del usuario con el escenario, según en que modo de escenario se encuentre, modo escritorio desde la web desde tu ordenador o inmerso en la propia realidad virtual.

Centrándonos en el modo escritorio, la forma de interactuar con los elementos de la escena se realiza clicando en ellos, para el movimiento de la cámara se utilizará el ratón y para el movimiento del propio usuario se hará uso de las teclas W, A, S, D o de los cursores del teclado.

Si por el contrario usamos la aplicación con las gafas de realidad virtual, la forma de interactuar con los elementos será con los mandos de estas, el movimiento de la cámara dependerá de donde mire el usuario y para el movimiento del usuario usaremos el joystick de uno de los mandos o con el propio movimiento del mismo.

Todo comienza en la escena principal, en la cual se puede acceder tanto al editor como a la escena donde puedes ver las instrucciones de la aplicación.

Si accedemos a la página de instrucciones el usuario se verá inmerso en una escena, podrá moverse por ella y a su vez leer las instrucciones del editor.

Si por el contrario accedemos a la escena de la aplicación, el usuario se encontrará frente a la zona de construcción y con varios botones en la parte inferior de la cámara como se muestra en la figura 4.1. El botón situado más a la izquierda hace referencia al selector de figuras. El siguiente botón siguiendo el orden hacia la derecha, es el que nos permite activar o desactivar el modo grupo. El botón que sigue hacia la derecha nos permite mostrar u ocultar los manejadores, por último nos encontramos con el botón que nos da acceso a los diferentes ambientes que se pueden añadir al escenario.



Figura 4.1: Inicio página del editor

Si el usuario pulsa el menú de selector de figuras, podrá elegir tanto las figuras básicas como varios modelos 3D, si elegimos uno de estos, aparecerá un menú en el cual, pulsando los botones “+” y “-”, podremos modificar dos atributos principales de las figuras, la orientación y el tamaño.

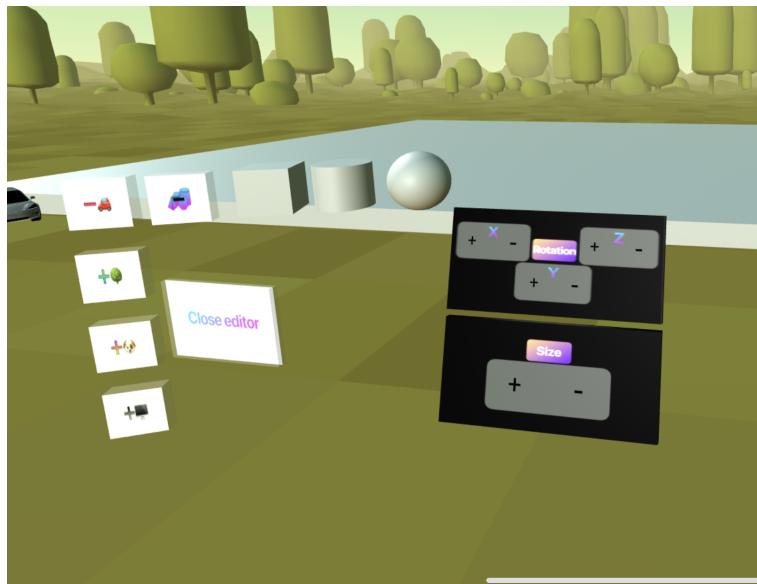


Figura 4.2: Menú modificación glTFs

Si el usuario decide pulsar una de las tres figuras básicas, el menú de modificación de la figura se amplia de manera notable, podremos modificarlas del mismo modo que el anteriormente, pulsando los botones + y -, el tamaño, el color, la orientación, la transparencia, opacidad, metálico y la rugosidad. En la figura 3.16 se puede visualizar dicho menú.

Una vez elegida la figura y las modificaciones correspondientes, la cogeremos del pódium y la arrastraremos al lugar deseado del escenario. En la figura 3.20 se muestra cómo se visualiza la figura en este.

Tras haber creado varias figuras, el usuario podrá crear otra, siendo esta la unión de varias, esto lo puede realizar gracias al modo grupo, a la cual se podrá acceder pulsando en el botón situado más a la izquierda que se muestra en la figura 4.3.

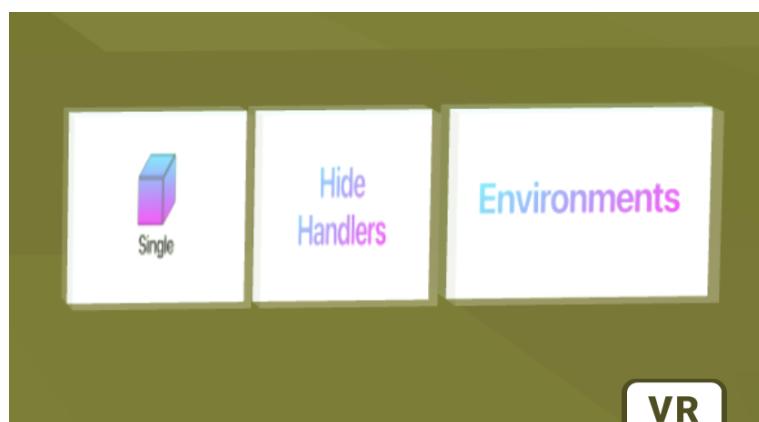


Figura 4.3: Botones para acceder a las funcionalidades.

Una vez el modo grupo esté activado, deberás pulsar las figuras que quieras unir, estas realizarán una animación haciendo ver que han sido seleccionadas y una vez ya las tengamos todas, volveremos a pulsar en el botón para confirmar el grupo. Esta figura se moverá mediante el manejador, generado en el pódium. Esta funcionalidad puede activarse mediante un atajo de teclado, la tecla Q.

Si se crean muchos grupos de figuras, el escenario estará formado tanto por las propias figuras como por los manejadores. El usuario tendrá el poder de ocultar los manejadores pulsando el botón 'Hide handler', como se observa en la figura 4.3.

La posibilidad de cambiar de ambientes en la escena es posible pulsando botón de menú de ambientes situado a la derecha de la pantalla. En él, nos aparecerán todos los ambientes que podemos elegir, para elegirlos solo tendremos que pulsarlos y el ambiente del escenario

cambiará.

Por último en la parte superior tanto izquierda como derecha el usuario se encontrará dos botones mediante los cuales poder acceder tanto a la página de inicio como a la de instrucciones.

4.2. Construcción de escenas

Gracias a cómo está construida la aplicación, podemos utilizar los componentes en cualquier otro escenario que no sea el definido para este proyecto.

Para poder empezar a usar la aplicación, lo único que debemos añadir a nuestro código HTML es lo siguiente:

```
<!-- Podium -->
<a-entity class="podium" podium></a-entity>
<!--User Camera-->
<a-entity>
<a-box id="showeditor" showeditor></a-box>
<a-entity id="menu" menu></a-entity>
<a-entity id="menuenv"></a-entity>
<a-entity id="editor"></a-entity>
<a-box id="showhandler" handlereditor></a-box>
<a-box id="deletehandler" deletehandler></a-box>
<a-box id="selectenviroments" selectenv></a-box>
<a-entity gotohome ></a-entity>
<a-entity gotoinstructiondesktop ></a-entity>
</a-entity>
```

Con este código, de ejemplo, ya podrás acceder a todas las funcionalidades de la aplicación como se muestra en la figura 4.4.

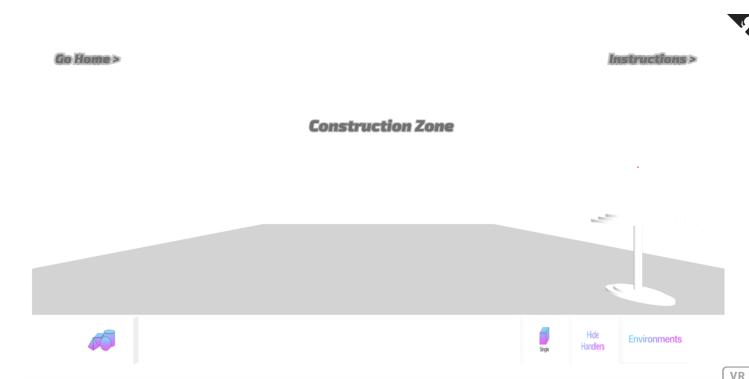


Figura 4.4: Página Básica

Por lo que observamos en el código, con dos elementos principales, como son el usuario y el podium, ya se tendría acceso a todas las funcionalidades de la aplicación.

Destacaré que mi aplicación no es un simple editor, sino que contiene diferentes componentes que pueden ser usados bien por separado, bien juntos, en cualquier otra aplicación de este campo u otro.

4.3. Implementación

En esta sección más técnica explicaré la arquitectura de la aplicación y los componentes creados para la construcción de este proyecto y explicaré la función que ejercen cada uno de estos.

Comenzando con la arquitectura de la aplicación, partiré del esqueleto principal de esta.

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <a-scene id="scn" overlay-visibility>
      <a-assets></a-assets>
      <!-- enviroment-->
      <a-entity id="env" environment="preset:forest"></a-entity>
      <!-- Podium -->
      <a-entity class="podium" podium></a-entity>
      <!--User Camera-->
      <a-entity>
        <a-box id="showeditor" showeditor></a-box>
        <a-entity id="menu" menu></a-entity>
        <a-entity id="menuenv"></a-entity>
        <a-entity id="editor"></a-entity>
        <a-box id="showhandler" handlereditor></a-box>
        <a-box id="deletehandler" deletehandler></a-box>
        <a-box id="selectenviroments" selectenv></a-box>
        <a-entity gotohome ></a-entity>
        <a-entity gotoinstructiondesktop ></a-entity>
      </a-entity>
    </a-scene>
  </body>
</html>
```

Como se puede ver en esta versión simplificada del código, este está formado por diferentes elementos que contienen los distintos componentes creados.

Lo primero que nos encontramos es el elemento head, en el cual se introduce todo lo necesario para que A-Frame y los distintos componentes, funcionen correctamente.

En el siguiente elemento, assets, se introducen todas las imágenes, modelos 3D y también un elemento llamado mixin que es un plantilla de creación de las figuras básicas para no repetir código en la creación de estas.

Tras el elemento assets, nos encontramos con el elemento que crea el ambiente de la escena y que será modificado dinámicamente mediante un componente que explicaré más adelante.

Tras este, nos encontramos con el elemento pódium en el cual generaré la figuras que vayamos creando.

Por último nos encontramos con el elemento usuario, el padre de los elementos de menú de interacción de la escena.

Como ya he comentado, a cada uno de estos elementos se le añadirán los componentes, a continuación pasaré a explicarlos.

Clasificaré los componentes según el elemento al que afecte.

- **Pódium.** Este componente se encarga de crear todas las figuras que componen el podium y su posición en la escena. Este es activado al generar el escenario y su función no es otra que tener una posición de referencia en la cual crear las figuras.

Los componentes que modifican el comportamiento del selector de figuras son:

- **Showeditor.** Es el encargado de generar las figuras disponibles para su creación y edición, a su vez genera el botón de acceso a los modelos 3D disponibles mediante la creación de otro componente. Este componente hace el uso de un evento encargado de escuchar si el elemento es pulsado, de este modo se activa el componente.
- **Showmorefigure** Componente generado a partir del anterior, encargado de crear cuatro botones con sus respectivos componentes, estos separan los modelos 3D en cuatro categorías distintas. Este componente reacciona al evento clicar, por lo que se activa al pulsar el botón.
- **Show(vehicles/plants/animals/gadgets)**. Este componente acompaña a los cuatro botones generados mediante el componente anterior, si pulsamos cada uno de estos botones,

te mostrará los diferentes modelos 3D. Por lo que este componente reacciona al mismo evento que los anteriores.

Los componentes que modifican los atributos de las figuras que vamos creando son:

- **Editentity.** La función de este componente es crear los elementos del menú modificación de la figura, cada uno con sus respectivos componentes, como se muestra en la figura 3.16. Este componente se activa al pulsar cualquiera de las 3 figuras básicas al abrir el menú de selector de figuras.
- **Changeattribute.** Este componente es creado por el anterior, y acompaña a las figuras de diferente color que se pueden ver en la parte del menú dedicada al cambio de color. El objetivo de este componente trata de extraer el atributo que contiene el color de las figuras del menú clicando en ellas y aplicárselo a la figura que he creado.
- **(Metalness/roughness-opacity)(up/down).** Estos componentes se encuentran en los botones + y - de la parte del menú que te permite modificar los atributos de metalizado, rugosidad y opacidad de la figura, este es creado al pulsar una de las tres figuras principales. Accediendo a dichos atributos de la figura y modificándolos ya sea aumentándolos o disminuyéndolos, su objetivo es modificar dichos atributos.
- **Size(up/down)(x/y/z).** Estos componentes tienen el mismo comportamiento que los anteriores y son creados del mismo modo, pero su objetivo es modificar el atributo de escala de la figura, aumentándolo o disminuyéndolo pulsando los botones + o -.
- **Rotation(up/down)(x/y/z).** Igual al anterior con la diferencia que este se centra en el atributo de la rotación.
- **Dotransparent.** La activación de este componente se genera al pulsar en el botón del menú dedicado para ello. El fin principal de este es modificar el atributo de transparencia activándolo cuando se pulsa dicho botón.

Los dos componentes siguientes no voy a desarrollarlos ya que tienen el mismo comportamiento que los anteriores pero dirigidos a los modelos 3D. No comparto componentes que realizan la misma función sobre figuras básicas debido a que la escala de los modelos en 3D actúa de manera diferente sobre las figuras básicas de A-Frame.

- **Editorglts**
- **Size(up/down)gltf(x/y/z)**

Sobre la funcionalidad modo grupo podemos distinguir los distintos componentes:

- **Handlereditor.** Este componente se genera al cargar el escenario y acompaña al botón que permite acceder a la funcionalidad del modo grupo. Este componente reacciona al evento clicar por lo que su activación y desactivación se realiza mediante la pulsación del botón. Cuando el componente esté activado y así lo mostrará la imagen del botón, creará un figura básica en el pódium que tomaré como manejador, a todas las figuras de la escena que sean clicadas en este momento se les añadirá un nuevo componente que permitirá unir dichas figuras.
- **Possibilityofgroup.** Este componente se genera al pulsar una figura con el modo grupo activado, se encarga de crear una figura idéntica a la que hay en el escenario y en la misma posición, pero esta vez siendo hija del manejador, de esta manera, cuando el manejador sea desplazado estas se moverán del mismo modo.
- **Deletehandler.** Componente generado al cargar el escenario acompañando en el botón que nos permite ocultar y mostrar los manejadores. Este componente se activa pulsando dicho botón y su objetivo es modificar el atributo encargado de la opacidad modificándolo para así poder mostrar los manejadores cuando este esté desactivado u ocultándolos cuando se encuentre activo.

Los componentes que se relacionan con el menú de ambientes son:

- **Selectenv.** Este componente es generado al cargar el escenario y acompaña al botón que permite acceder a los diferentes ambientes que añadir a la escena. Al pulsar el botón que contiene el componente crea distintos botones con previsualizaciones de la escena con los respectivos ambientes y con sus respectivos componentes.
- **Env(1/2/3/4/5).** Componente generado por el anterior cuya función es modificar el ambiente de la escena y añadir el ambiente elegido por el usuario según el botón pulsado.

Por último, los componentes que añaden comportamiento a los botones para ir a la pantalla de inicio y a la de instrucciones son:

- **Gotoeditor(glasses)**. Este componente se localiza en la parte superior de la escena y es creado al generar la escena. Su función es la de llevar al usuario a la escena de creación del editor. La activación de este componente se realiza pulsando el propio texto. Este componente solo es generado en la página principal y en la de instrucciones.
- **Gotohome(glasses)**. Mismo comportamiento que el anterior, pero en este caso solo se generará en la página de creación y en la de instrucciones.
- **Gotoinstructions(glasses)**. Mismo comportamiento que los anteriores, pero generado en la página principal y en la página de creación.

Para finalizar, cabe desatacar los componentes que nos ofrece A-frame y hacemos uso en nuestra aplicación:

- **Clickable**. Este componente nos aporta, al elemento que lo contiene, la cualidad de poder ser clicado para futuros eventos, este componente es utilizando en casi todos nuestros elementos de la aplicación.
- **Grabbable**. Este componente nos permite arrastrar a cualquier parte de la escena los elementos que lo contengan.

4.4. Escenarios creados con la aplicación

A continuación se pueden observar algunos ejemplos de escenarios creados con el editor.



Figura 4.5: Casa



Figura 4.6: Parque

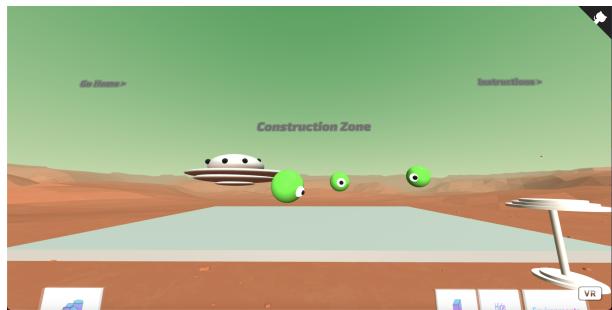


Figura 4.7: Ovnis



Figura 4.8: Faro

Capítulo 5

Conclusiones

En este punto de la memoria, visualizaré si se lograron los objetivos presentados en el capítulo 1, y el grado de cumplimiento de estos, también comentaremos los problemas surgidos durante la realización del mismo.

De manera general, se puede decir que he conseguido todos los objetivos propuestos al inicio del proyecto.

Respecto de los objetivos generales realizados,

- La creación de un editor de escena en realidad virtual.
- Este editor está basado en las tecnologías tanto de A-Frame como de WebGL.
- El funcionamiento tanto en un entorno de escritorio mediante el navegador web como en modo realidad virtual mediante las Oculus Quest.

Sobre los objetivos específicos conseguidos nos encontramos.

- La creación tanto de la página principal y página de usuario.
- En cuanto a las funcionalidades propuestas:
 - La creación de un menú desde el cual crear y modificar distintos atributos de las figuras.
 - La creación del modo grupo, gracias a él podremos construir una figura mediante la unión de varias.

- La posibilidad de ocultamiento de los manejadores del escenario.
- La oportunidad de modificar el ambiente del escenario.
- Añadidura de movimiento al usuario en ambas versiones.
- El uso de la plataforma GitHub para el alojo del proyecto, tanto de la aplicación como de la memoria como la Web. Esto se puede ver reflejado en los links adjuntos en el capítulo tres de esta memoria.

5.1. Aplicación de lo aprendido

Durante la realización del grado, he obtenido conocimiento sobre distintos temas, uno de ellos la programación, en distintas asignaturas, la cual me ha permitido desarrollar este proyecto, tanto a la hora de aplicar lo aprendido como la manera de aprender lo necesario para el desarrollo de este.

Algunas asignaturas de las que hablo son:

1. Servicio y Aplicaciones Telemáticas, esta asignatura me ayudó con el uso del navegador, tanto de la consola, como el tratamiento de errores, el conocimiento de la estructura del DOM en el cual poder añadir elementos y eventos y en menor medida el lenguaje HTML.
2. Desarrollo de Aplicaciones Telemáticas, en la cual aprendí el lenguaje HTML y en menor medida el lenguaje JavaScript.
3. Ingeniería en Sistemas de la Información, esta asignatura fue muy importante para el manejo de la plataforma GitHub y el tratamiento de control de versiones.

En resumen, todas las asignaturas relacionadas con la programación me han aportado distintos conocimientos, ya sea los distintos lenguajes de programación o aprender a resolver los distintos problemas que han ido surgiendo a lo largo de la creación del proyecto.

5.2. Lecciones aprendidas

Gracias a la creación de este Trabajo de Fin de Grado he adquirido diversos conocimientos y he aprendido muchas lecciones que pasaré a desarrollar a continuación.

1. Uno de los principales y más usados en este proyecto, A-Frame. Dentro de este framework incluimos:
 - Creación de escenas.
 - Creación de componentes, los cuales reaccionan con los elementos de la escena.
 - Construcción de distintos elementos de este framework como puede ser, la cámara, la iluminación, el ambiente, las figuras etc.
2. El lenguaje JavaScript, ya que este no se enseña en profundidad en la carrera y este trabajo me ha permitido aumentar el conocimientos sobre este. La creación de eventos y la modificación en tiempo real de los elementos de la escena es una de las funciones de este lenguaje que se incluyen en el proyecto.
3. El uso de los entornos de realidad virtual ya sea tanto en el navegador como con las gafas de realidad virtual.
4. El aumento del conocimiento de las herramientas de desarrollo de este proyecto, en las que se incluyen:
 - LaTeX para la creación de textos técnicos como este.
 - PyCharm para el desarrollo del código de la aplicación.
 - GitHub, como ya he contado anteriormente para el alojamiento del código y el tratamiento de control de versiones.

5.3. Trabajos futuros y esfuerzo dedicado

Como se suele decir en un proyecto de este tipo, ningún proyecto se termina, siempre se pueden añadir nuevas funcionalidades que amplíen y mejoren este, por lo que a continuación enumeraré una serie de ideas para poder incluir en un futuro.

- La posibilidad de poder guardar el escenario creado, ya sea una figura o todo el escenario, de este modo poder añadir versatilidad al proyecto y podernos llevar las figuras creadas a otros tipos de escenarios.

- La posibilidad de poder modificar los atributos de las figuras, directamente en estas, y no mediante un menú. En el caso del navegador, con el uso del ratón y en el caso de la realidad virtual, directamente con los mandos o con las manos.
- Poder modificar las figuras de una manera más específica, por ejemplo, poder eliminar una parte de la figura.
- Añadir más posibles ambientes al escenario.
- Añadir más modelos 3D al escenario.
- Añadir más físicas a las figuras del escenario, como puede ser, gravedad.

Sobre el tiempo dedicado a este proyecto, han sido unos 10 meses. He ido realizando el trabajo mientras cursaba mis prácticas en empresa. Un gran porcentaje de este, fue hecho los fines de semana y entre semana tras salir del trabajo.

Bibliografía

[1] A-Frame – Make WebVR.

[https://aframe.io/.](https://aframe.io/)

[2] The Industry's Foundation for High Performance Graphics.

[https://www.opengl.org//.](https://www.opengl.org/)

[3] Essential Tools for Software Developers and Teams.

[https://www.jetbrains.com/.](https://www.jetbrains.com/)

[4] Immersive-web/webxr.

[https://github.com/immersive-web/webxr/blob/master/explainer.mdwhat-is-webxr.](https://github.com/immersive-web/webxr/blob/master/explainer.mdwhat-is-webxr)

[5] Qué es SCRUM.

[https://proyectosagiles.org/que-es-scrum.](https://proyectosagiles.org/que-es-scrum)

[6] C. M. Bill Kennedy. *HTML XHTML: The Definitive Guide*. 2000.

[7] P. B. Brent Bell. *Introducing GitHub*. 2014.

[8] D. Cantor. *WebGL Beginner's Guide*. 2012.

[9] M. Haverbeke. *Eloquent JavaScript: A Modern Introduction to Programming*. 2011.