Universidad
Rey Juan Carlos

ESCUELA DE INGENIERÍA DE FUENLABRADA

GRADO EN INGENÍERIA EN SISTEMAS AUDIOVISUALES Y MULTIMEDIA

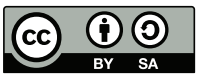
TRABAJO FIN DE GRADO

TÍTULO DEL TRABAJO CON LETRAS MAYÚSCULAS PARA SUSTANTIVOS Y ADJETIVOS

Autor : Juan José Arias Rojas

Tutor : Dr. Jesús María González Barahona

Curso académico 2024/2025



Dedicado a

todos aquellos que me animaron y apollaron

incluso en mis momentos de debilidad

II

Agradecimientos

Resumen

Summary

Índice general

Índice de figuras

Capítulo 1

Introducción

Capítulo 2

Tecnologías utilizadas

En esta sección se muestran las distintas tecnologías que han sido necesarias para el desarroyo de este proyecto tanto de forma directa como indirecta.

2.1. Tecnologías principales

Aquí se nombran y explican aquellas tecnologías que han tenido una implicación directa con el proyecto y que han tenido un papel crucial e imprescindible con las cuales, sin ellas, no se habria popido llegar a los resultados obtenidos.

2.1.1. HTML5

HTML5 es la quinta iteración del lenguaje de HTML (*marcado de hipertexto*), el cual es la estructura básica y principal de toda página web. Desarrollado conjuntamente por W3C (*World Wide Web Consortium*) y WHATWG (*Web Hypertext Application Technology Working Group*). HTML5 introduce mejoras sustanciales respecto a sus anteriores versiones, incluyendo nuevas etiquetas semánticas, soporte multimedia mejorado y compatibilidad ampliada con diversos navegadores y dispositivos, dándole mayor flexibilidad y dinamismo a la creación de páginas web.

Una de sus nuevas introducciones clave es la introducción de etiquetas semánticas como: `<header>`, `<article>`, `<section>` y `<footer>`, que facilitan una estructuración clara y accesible del contenido web. Estas etiquetas no solo ayudan a mejorar la organización de la

información, si no que también facilitan la búsqueda de información por parte de los motores de búsqueda y facilitan la interpretación por parte de los desarrolladores. Además, HTML5 incorpora nuevas interfaces de programación de aplicaciones (*API*), destacándose las funcionalidades de almacenamiento local mediante `localStorage` y `sessionStorage`, que permiten la gestión eficiente de datos directamente en el navegador, eliminando la necesidad de bases de datos externas. [3]

HTML5 también introduce mejoras significativas en la gestión de contenido multimedia, eliminando la necesidad de complementos externos. Para esto se implementaron las etiquetas `<audio>` y `<video>` las cuales permiten la incorporación directa de archivos de sonido y video en las páginas web, permitiendo mayor dinamismo en el desarrollo. Otro elemento que se añadió fue el elemento `<canvas>` el cual habilita la generación de gráficos y animaciones en tiempo real a través de JavaScript, permitiendo el desarrollo de aplicaciones interactivas y videojuegos en línea.

HTML5 se ha establecido firmemente como un estándar en el desarrollo de aplicaciones web progresivas (*PWA*) y en entornos multiplataforma. Su integración con tecnologías como CSS3 y JavaScript permite la creación de interfaces dinámicas y adaptativas, compatibles con una amplia gama de dispositivos, desde ordenadores hasta tabletas y teléfonos.

2.1.2. JavaScript

JavaScript es un lenguaje de programación ligero y multiplataforma utilizado principalmente para la creación de contenido dinámico e interactivo para páginas web. Su felixibilidad permite desarrollar elementos para mejorar la interacción del usuario en páginas web tanto del lado del servidor como del lado del cliente. En 1997, JavaScript fue estandarizado por *ECMA* como ECMAScript y poco después como un éstandar *ISO*.

JavaScript, creado por Brendan Eich de Netscape en 1995 bajo el nombre de *Mocha*, posteriormente se renombró a LiveScript y finalmente a JavaScript. En el año 2000, JavaScript se extendió al lado de los servidores con la introducción de tecnologías como *Node.js*, y posteriormente su popularidad creció con la llegada de *AJAX*. Y Con la llegada de ECMAScript 6 en 2015, se introdujeron características mas avanzadas y actualizaciones anuales, haciendo que hoy en día sea uno de los lenguajes mas importantes en el desarrollo web.

JavaScript es un lenguaje de programación de alto nivel, lo que significa que su leguaje esta

diseñado para ser lo mas 'humano' posible, permitiendo una rapida comprensión del codigo sin necesidad de un nivel alto de conocimientos. Es un lenguaje basado en eventos ya sean entradas de ratón o entradas de teclado, permitiendo la creación de interfaces de usuario interactivas. JavaScript puede integrarse en un HTML dentro de la etiqueta `<script>` de forma directa, escribiendo el codigo. También, se puede introducir codigos de JavaScript, los cuales estén en archivos distintos con la extensión correspondiente al lenguaje (.js), esto es posible vinculando dicho archivo en la cabecera del HTML.

2.1.3. WebXR

WebXR [4] es una tecnología desarrollada por W3C (*World Wide Web Consortium*) que permite crear experiencias inmersivas desde el navegador. Esta tecnología combina la Realidad Aumentada (*AR*) y la Realidad Virtual (*VR*) con la accesibilidad de un navegador web. Lo cual permite a los usuarios experimentar e interactuar con entornos tridimensionales a través de cualquier dispositivo con acceso a un navegador compatible.

A medida que la tecnología de WebXR ha ido evolucionando, se han desarrollado distintos tipos de experiencias para poder adaptarse a distintos contextos y necesidades. Hoy en día, se podria clasificar los distintos tipos en 3 categorias:

- **WebXR AR:** Este tipo de WebXR combina el mundo virtual y el mundo real, permitiendo que distintos elementos del mundo virtual puedan superponerse en el entorno físico a través de la cámara de un dispositivo compatible, pero sin que llegue a influir el mundo real en los elementos virtuales.

- **WebXR VR:** Este modo permite a los usuarios sumergirse en el entorno virtual creado, y con la ayuda de dispositivos suplementarios como auriculares, la experiencia es aún mayor. Esta tecnología permite a los usuarios interactuar y experimentar en primera persona distintos entornos virtuales, desde juegos y entretenimiento, hasta simulaciones virtuales y visualizaciones en 3D.

- **WebXR MR(Mixed Reality):** Esta tecnología combina el mundo real y el virtual de forma mas profunda que la tecnología AR. Permite a los usuarios poder interactuar con elementos tantos virtuales como físicos y que estos puedan interactuar entre si. Este tipo

de combinación proporciona un nivel mayor de interacción entre el usuario y y su entorno, dando mayor numero de posibilidades para la creación de contenido.

Estos distintos tipos de WebXR ofrecen distintos tipos de experiencias dependiendo del entorno virtual que se quiera diseñar.

2.1.4. WebLG

WebGl (*Web graphics Library*), se trata de una tecnología de bajo nivel multiplataforma usada para la renderización de gráficos tanto tridimensionales como bidimensionales dentro de cualquier navegador que sea compatible.

WebGl fue lanzado en 2011 por el grupo Khronos. Esta tecnología se fundamenta en OpenGL ES, la cual es una variante simplificada de OpenGL, diseñada para dispositivos moviles. WebGL ha sufrido numerables actualizaciones, lo cual ha permitido una evolución continua que ha ido mejorando tanto su funcionalidad como compatibilidad tanto en navegadores de escritorio como en navegadores en dispositivos móviles.

WebGL esta diseñado para trabajar directamente con la GPU (*Graphic Processing Unit*) del dispositivo, lo cual permite un mayor aprovechamiento de la computación para poder generar gráficos detallados y de alta calidad. Además, la tecnología de WebGL esta diseñada para poder integrarse de manera fluida con otros estandares de desarrollo web como HTML, CSS o DOM.

2.1.5. Three.js

Three.js [5] es una biblioteca de codigo abierto de JavaScript utilizada para la creación de gráficos 3D en los navegadores web. Three.js funciona sobre WebGL [2], la cual permite generar y visualizar animaciones y escenas 3D en el navegador sin necesidad de complementos. La tecnologia de WebGL tambiénpuede usarse junto con el elemento `<canva>` de HTML.

Esta biblioteca proporciona una API de alto nivel la cual permite al usuario la creación y manipulación de geometrías 3D como cubos, esfera o planos, así como la aplicación de texturas o la aplicación de tanto camaras como de efectos de iluminación en las escena, permitiendo que el desarrollo de dichas escenas sea mucho mas simplificado. También, Three.js permite la posibilidad de importar modelos 3D desde archivos distintos creados desde aplicaciones distintas.

La biblioteca de Three.js ofrece una alta variedad de herramientas que permiten la gestión y control de usuario, facilitando asi tanto la navegación como la interacción dentro de las escenas 3D. También, Three.js soporta animaciones suaves y dinámicas tanto para objetos en la escena como para camaras, lo cual permite la creación de efectos visuales mas dinámicos e impresionantes al igual que juegos interactivos.

2.1.6. A-frame

A-Frame [1] es un web framework diseñado para construir experiencias de realidad virtual. A-frame esta basado en HTML y en JavaScript, permitiendo realizar cualquier escena que uno pueda imaginar sin necesidad de conocimientos avanzados en gráficos 3D. Al estar basado en HTMl es posible realizar escenas simples directamente desde un archivo HTML simplemente importando la libreria de A-frame y añadiendo dentro de la etiqueta `<a-scene>` cualquier elemento que desee el usuario.

Uno de los ejemplos mas simples de A-frame es el siguiente:

```
<html>
  <head>
    <script src="https://aframe.io/releases/1.6.0/aframe.min.js"></
       script>
  </head>
  <body>
    <a-scene>
      <a-box position="-1 0.5 -3" rotation="0 45 0" color="#4CC3D9"
         ></a-box>
      <a-sphere position="0 1.25 -5" radius="1.25" color="#EF2D5E">
         </a-sphere>
      <a-cylinder position="1 0.75 -3" radius="0.5" height="1.5"
         color="#FFC65D"></a-cylinder>
      <a-plane position="0 0 -4" rotation="-90 0 0" width="4"
         height="4" color="#7BC8A4"></a-plane>
      <a-sky color="#ECECEC"></a-sky>
    </a-scene>
  </body>
```

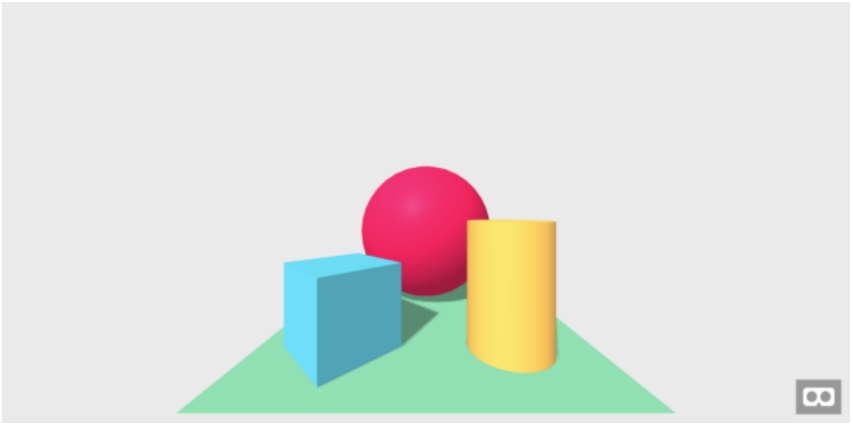


Figura 2.1: Escena basica de A-frame

```
</html>
```

Listing 2.1: Escena A-Frame básica

Este código de HTMl genera una de las escenas mas básicas que se pueden hacer en A-frame. La cual consiste de un plano y encima una serie de figuras geometricas como se aprecia en la figura 2.1

Una de las principales características de esta tecnología es su arquitectura basada en un modelo de entidad-componente (*Entity Component System*), donde cada objeto dentro de la escena es una entidad que luego es integrada con Three.js para crear la escena. Una entidad en A-frame es un objeto HTML que se crea añadiendo la etiqueta `<a-entity>` y el componente es la apariencia, comportamiento y funcionalidad que se le asigna mediante JavaScript.

Para hacer uso de un componente se requiere de una serie de pasos previos. Si el componente es externo es necesario importarlo en la cabecera del HTML del mismo modo que se importa A-frame y posteriormente añadirlo a la entidad en la que uno desee usarlo. Por otra parte, si el componente es de nuestra creación, primero hay que registrar y definir el componente. Esto se hace en un archivo externo de JavaScript mediante el método de `AFRAME.registerComponente`. A este método, del mimo modo que una función primero se le asigna un nombre, este sera el nombre del componente. Luego, este metodo consta de varias funciones internas las cuales se encargan de definir la apariencia y lógica de dicho componente. Alguna de estas funciones son la función `schema`, `init` o `tick`. La función `schema`, define las propiedades principales del componente, estas se pueden modificar dependiendo de la lógica del componente desde el HTML a la hora de llamarlo. La función `init` se ejecuta una única vez al inicio cuando se

inicializa el componente y la función `tick` se ejecuta a cada frame de la escena. Una vez el componente esta registrado y programado importamos el archivo que contiene el componente al HTML y lo insertamos a la escena en la entidad u objeto que deseemos.

Un ejemplo de la llamada de un componente dentro de un archivo de HTML se puede apreciar en 2.2

```
<html>
  <head>
    <script src="https://aframe.io/releases/1.6.0/aframe.min.js"></
       script>
    <script src = "componente1.js"></script>
    <script src = "componente2.js"></script>
  </head>
  <body>
    <a-scene>
      <a-entity componente1></a-entity>
      <a-box componente2></a-box>
    </a-scene>
  </body>
</html>
```

Listing 2.2: Ejemplo de entidad en A-frame

En este ejemplo primero se muestra como a una entidad vacia se le añade un compoennte llamado `componente1` y también se muestra como a una entidad de cubo se le añade el componente `componente2`.

2.2. Tecnologias auxiliares

En esta sección se detallan y describen las distintas tecnologías que aunque no hayan influido de manera directa en los resultados del proyecto, han tenido un papel crucial para el desarroyo de este.

2.2.1. Visual Studio Code

2.2.2. Github

2.2.3. Meta Quest 3

2.2.4. LaTeX

Capítulo 3

Desarrollo del proyecto

Capítulo 4

Resultados

Capítulo 5

Conclusiones

5.1. Aplicación de lo aprendido

5.2. Lecciones aprendidas

5.3. Trabajos futuros

Bibliografía

[1] A-Frame. A-frame - introduction, 2024.

[2] EncodeBiz. Parte 1: Webgl y su impacto en el desarrollo web moderno, 2024.

[3] E. Freeman and E. Robson. Head first html and css, 2018.

[4] Onirix. Webxr: Ejemplos y desarrollo en realidad extendida, 2024.

[5] Wikipedia contributors. Three.js, 2024.