

# **Memoria técnica del proyecto VALERIE**

Proyecto de innovación docente PIBD Básico de VALERIE (código nº 22-147) del Plan de formación e innovación docente 2022-2023, Plan de innovación y Buenas Prácticas Docentes.

Versión 1.0  
Mayo 2023.



Memoria técnica del proyecto VALERIE v1.0 por Juan José Ramos Muñoz tiene una Licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional

# Resumen

Este documento presenta los resultados del proyecto de innovación docente Valerie, sobre la elaboración de material de realidad virtual para exponer contenidos docentes. Su estructura tiene como objetivo describir el desarrollo del proyecto y proporcionar detalles para que otros grupos docentes puedan usar las conclusiones y resultados obtenidos en este proyecto, y aplicarlo en los suyos propios. Como resultado, cada capítulo mostrará cómo se llevó a cabo cada fase del proyecto, así como los consejos y buenas prácticas aprendidas durante su realización.

# Índice de contenidos

<b>Resumen</b> .....	<b>3</b>
<b>Introducción</b> .....	<b>5</b>
<b>Tecnologías candidatas</b> .....	<b>6</b>
Tecnologías de realidad virtual y realidad aumentada.....	6
Herramientas de desarrollo para WebXR.....	7
Motores de videojuegos con soporte para realidad virtual.....	7
Frameworks basado en javascript para el desarrollo de aplicaciones de realidad virtual y en realidad aumentada.....	8
<b>Diseño</b> .....	<b>10</b>
Diseño de boceto.....	10
Desarrollo de bocetos con inteligencia artificial.....	12
Modelado 3D de los bocetos.....	14
<b>Implementación de las experiencias</b> .....	<b>19</b>
Uso de la plataforma PlayCanvas.....	19
Programación de interactividad.....	21
Interacción VR/AR mediante webXR.....	22
<b>Conclusiones</b> .....	<b>22</b>
<b>Agradecimientos</b> .....	<b>23</b>
<b>Referencias</b> .....	<b>23</b>

# Introducción

Valerie es un proyecto de innovación docente de un grupo de profesorado perteneciente al departamento de Teoría de la señal, Telemática y Comunicaciones de la Universidad de Granada, contando además cuenta con profesorado de distintas especialidades, para formar un equipo multidisciplinar y complementario. En el presente proyecto se pretende desarrollar experiencias de realidad virtual y realidad aumentada que permitan al estudiante tener una perspectiva diferente de los conceptos fundamentales de las materias impartidas.

Las asignaturas técnicas con contenido de ingeniería pueden presentar conceptos difíciles de asimilar por los estudiantes. Esto es especialmente frecuente en el caso de conceptos muy abstractos.

La presentación de estos contenidos de una forma que permita visualizar mediante ejemplos el fundamento presentado, puede mejorar la atención y el interés de los estudiantes por la materia, así como la comprensión de dicho contenido. Para ello, se utilizarán tecnologías asequibles de realidad virtual.

Como efecto secundario, el hecho de trabajar en equipo por parte del profesorado para realizar contenido de una forma innovadora también puede fortalecer la coordinación entre las distintas asignaturas, y quizás mitigar el síndrome del quemado.

Una de las restricciones principales que tiene este proyecto es el que se utiliza en tecnología inmersiva de realidad aumentada (Augmented Reality, AR) y realidad virtual (VR), pero de forma que sean asequibles para el estudiantado. Dado que es complejo poder tener unas gafas de realidad virtual autónomas, como el caso de *Oculus Quest* o *HTC Pro*, en este proyecto se intenta abordar el desarrollo de aplicaciones de VR y AR para dispositivos más modestos. Este es el caso de los dispositivos móviles con una carcasa tipo *Google Cardboard*. Como ejemplo, estas carcasas suelen tener un coste inferior a 20 euros.

Por ello se han estudiado las distintas alternativas para poder ofrecer este tipo de servicio, y se ha elegido una de las herramientas de las que ofrecen la funcionalidad necesaria, teniendo en cuenta los requisitos antes esbozados.

Otro de los puntos más contributivos del proyecto es la experiencia obtenida en el desarrollo de material con formato VR y AR, en un equipo multidisciplinar, pero no experto en todas las facetas del proceso. Eso ha hecho que se hayan tenido que identificar herramientas y procedimientos alternativos, que facilitaran a personas no expertas parte de la implementación. Por eso, en las secciones de diseño e implementación, se describe cómo se ha llevado a cabo el desarrollo, así como se ofrecen consejos para quien quiera realizar experiencias similares.

# Tecnologías candidatas

Para diseñar una solución que satisficiera los requisitos identificados en la introducción, hubo que revisar qué tecnologías candidatas estaban disponibles, y qué herramientas de desarrollo se podrían utilizar.

## Tecnologías de realidad virtual y realidad aumentada

Uno de los aspectos fundamentales a considerar era la necesidad de incluir un servicio que no requiriera que el usuario tuviera que instalar demasiadas aplicaciones. Para lograr esto, se eligió la tecnología estándar webXR (World Wide Web Consortium, 2023), que permite utilizar el navegador web como plataforma para ejecutar este tipo de aplicaciones. Esta tecnología estándar de realidad extendida (eXtended Reality, XR) permite ejecutar aplicaciones de realidad virtual y aumentada, sacando el máximo partido de la plataforma soportada en el equipo del usuario. Por ejemplo, un estudiante que posea un casco de realidad virtual propias y esté conectado a su ordenador podrá disfrutar de la experiencia desarrollada, solo con conectarse a la página web en el navegador de su dispositivo. La aplicación se ejecutará de forma compatible con el dispositivo encontrado gracias a la tecnología webXR.

Si bien no todos los navegadores web soportan toda la funcionalidad de WebXR, uno de los más populares, Chrome de Google, sí que tiene un soporte casi íntegro. Este navegador es gratuito y se puede encontrar para casi todo tipo de dispositivos móviles, incluso para equipos fijos o de sobremesa. Se espera no obstante que haya un número creciente de navegadores web que soporten la extensión de webXR en el futuro.

Si bien es cierto que WebXR se puede implementar simplemente creando una página web utilizando JavaScript, esta aproximación puede resultar incómoda para desarrollar rápidamente una aplicación completa que ofrezca experiencias virtuales lo suficientemente interesantes. Además, esto requeriría que el desarrollador tenga un profundo conocimiento y formación en JavaScript y otras tecnologías relacionadas.

Por esta razón, es necesario utilizar otras herramientas que dispongan de editores, preferiblemente visuales, y que utilicen un lenguaje ampliamente conocido. En este sentido, los motores de videojuegos y algunos frameworks han sido desarrollados para permitir la visualización de experiencias en 3D desde el navegador.

Estos motores y frameworks ofrecen un entorno de desarrollo más completo, con herramientas y funciones específicas para la creación de experiencias virtuales más sofisticadas. Permiten crear gráficos en 3D, gestionar la física, manejar interacciones y otros aspectos necesarios para ofrecer una experiencia inmersiva de alta calidad.

# Herramientas de desarrollo para WebXR

A la hora de decidir qué herramienta utilizar para desarrollar experiencias de realidad virtual, realidad aumentada u otro tipo de experiencias audiovisuales, existen varias opciones que pueden facilitar el proceso. Entre ellas, destacan los motores de videojuegos y sus entornos de desarrollo correspondientes, los cuales permiten desarrollar de manera amigable mediante una interfaz gráfica, y el uso de editores para agregar código y comportamientos a los objetos. Estos motores brindan la posibilidad de desarrollar de forma homogénea experiencias interactivas y videojuegos.

Es importante mencionar que el uso de motores de videojuegos se ha vuelto cada vez más accesible debido a su popularización, y muchos de ellos están disponibles bajo licencias libres o gratuitas. Sin embargo, en el caso de las licencias gratuitas, es posible que algunos motores estén diseñados principalmente para el ámbito comercial, y no sean adecuados para su aplicación en el ámbito educacional.

Es recomendable investigar y evaluar las licencias de los motores de videojuegos antes de tomar una decisión, especialmente si se va a utilizar en un contexto educativo. Es importante asegurarse de que la licencia sea compatible y permita el uso educativo sin restricciones legales.

En el ámbito del proyecto se pretende además utilizar herramientas que sean asequibles y puedan ser utilizadas por profesores con distinta especialización en programación. Por eso se intenta identificar herramientas que sean lo suficientemente intuitivas como para que pueda ser utilizada por personal que no sea experto en programación o arte. Aunque siempre será necesario algo de programación, este tipo de herramientas elegidas intentan minimizar esta parte del proceso.

## Motores de videojuegos con soporte para realidad virtual

De entre las herramientas comerciales disponibles con soporte para realidad virtual y realidad aumentada, destaca el motor CryEngine (Crytek GmbH, 2023), creado por Crytek, que destaca por su alta fidelidad visual y su aplicabilidad a la realidad virtual. Sin embargo, es importante tener en cuenta que este motor no cuenta con soporte simple para dispositivos móviles. Además, su licencia no permite su uso en videojuegos serios, como los que se puedan desarrollar para docencia, lo cual podría ser una limitación para el tipo de experiencias que se desean desarrollar.

El siguiente motor destacable por su uso en el desarrollo de experiencias con realidad virtual es Unreal Engine (Epic Games, 2023). En realidad tiene una licencia que nos permitiría desarrollar nuestra aplicación sin tener que pagar poner ningún coste a no ser que tuviera que ser utilizada para la actividad comercial y sobrepasara 1 millón de dólares en beneficios. Sin embargo, este motor no tiene un soporte adecuado para experiencias tipo Google Cardboard, y requiere una formación avanzada y un equipo de desarrollo avanzado para poder desarrollar videojuegos o experiencias virtuales.

Otro de los motores más ampliamente utilizados en el mundo de los videojuegos es Unity 3D (Unity Technologies, 2023). Este motor tiene varias licencias que permiten

hacer un desarrollo de gratuito, aunque dependen de los ingresos de la empresa que lo use. Se programa mediante el lenguaje C#, aunque también tiene entornos visuales. Unity 3D tiene otra ventaja y es que tiene un soporte y documentación mediante tutoriales de resolución de problemas técnicos muy avanzados. Sin embargo, la dependencia con el cambio recurrente de condiciones en las licencias, puede hacer que el desarrollo se paralice.

Otro de los motores que pueden utilizarse a la hora de desarrollar videojuegos y obtener que interactiva este Godot Engine (Godot Engine team, 2023). Godot Engine es su motor de código abierto con una licencia muy permisiva y muy ligero. El desarrollo en esta plataforma es bastante sencillo y está basado en un lenguaje similar a Python. La licencia permisiva de este motor permite y la posibilidad de modificar su código fuente permiten que se pueda introducir extensiones que sean necesarias para desarrollos futuros.

Este motor ha sido elegido para poder desarrollar las primeras experiencias virtuales. A partir de la versión 4.0 del motor, tiene soporte para web XR de forma nativa. Sin embargo, tras meses de desarrollo probando documentación oficial del motor, se ha comprobado que su soporte para realidad virtual y realidad aumentada es limitado. Aunque se espera que en un futuro pueda ser un buen candidato para poder desarrollar este tipo de experiencias, dada la facilidad de programación y el soporte que cada vez va incluyendo, se decidió adoptar otro motor diferente.

## Frameworks basado en javascript para el desarrollo de aplicaciones de realidad virtual y en realidad aumentada.

Además de los motores anteriormente descritos, que facilitan el desarrollo de experiencias completas interactiva audiovisuales o de videojuegos, existe una serie de frameworks que están orientados a su ejecución en el navegador web, y que pueden implementar y soportar extensiones como web XR.

De entre estos frameworks destaca A-Frame que es un frameworking incluye un editor de experiencia virtuales, y que se programa con un lenguaje de marca similar al de HTML5 (Supermedium, 2023). La principal ventaja de este motor es que es muy ligero y que tiene un editor sencillo, así como una documentación suficiente. La desventaja de utilizar este framework es que la programación de los comportamientos es más compleja que en el caso de los motores anteriores. Además, el renderizado final de la parte visual es muy simple, comparado con otros motores.

Otro de los frameworks destacables a la hora de desarrollar aplicaciones virtuales es *threejs* (Three.js team, 2023). Este Framework permite renderizar con una calidad muy alta distintos tipos de contenido. Esto permitiría el renderizado de aplicaciones muy realistas. Incluye además un editor que permite crear las escenas que van a renderizarse. Sin embargo, como en el caso anterior, tiene una limitación en cuanto a cómo programar los comportamientos de las experiencias virtuales



Otro framework destacable a la hora de desarrollar aplicaciones o experiencias inmersivas, podemos destacar ar.js (ar.js team, 2023). Este framework está orientado a desarrollar experiencias de la unidad aumentada basada también en el navegador utilizando Javascript. Sin embargo la programación del comportamiento sigue siendo más compleja, comparada con los motores de videojuego anteriormente descrito. Tampoco posee un editor amigable.

Por último, una de las herramientas que puede permitir el desarrollo de experiencias virtuales interactivas usando webXR, es el motor de videojuego *PlayCanvas*. Este motor está basado en Javascript que puede ejecutarse desde el navegador web, mediante su propio editor. Puede programarse a partir de un entorno visual y el desarrollo de scripts en Javascript. La ventaja principal de este motor es que integra algunos de los frameworks anteriormente comentados, y que tiene un soporte muy avanzado para multiplataforma de web XR. De esta manera este motor tiene las ventajas que habíamos reseñado anteriormente, incluyendo un soporte suficientemente amplio para los requisitos del proyecto.

Si bien playcanvas.com es un producto que puede tener una suscripción mensual, tiene una versión de suscripción gratuita si se cumplen ciertas condiciones. La primera condición es que los proyectos se publiquen de forma abierta en su plataforma. Para el presente proyecto esta restricción no es un problema. Otra de las ventajas que tiene PlayCanvas es que permite alojar los resultados de los juegos en su propia página. De esta manera no hace falta que el grupo de desarrollo mantenga una infraestructura propia para poder ejecutar los videojuegos. La otra ventaja fundamental es que soporta web XR de forma nativa. Por último tiene una documentación suficientemente extensa para poder desarrollar los ejemplos que se diseñados. Para ello cuenta con asistentes para poder generar plantillas de distintos tipos de experiencia, como puede ser de realidad virtual o Realidad Aumentada. Aunque el editor sea parte de él servicio que ofrece playcanvas.com, el motor que hace que funcione los videojuegos de este tipo es abierto, con lo que en un futuro podría extenderse para ofrecer nuevas funcionalidades

Desde el punto de vista de equipos que no sean expertos en programación, cabe destacar que el editor visual de PlayCanvas permitiría, sin requerir profundos conocimientos, realizar experiencias basadas en la presentación de animaciones o de modelos 3D. Este tipo de experiencia serían muy fáciles de generar a partir de las plantillas que ofrece el propio entorno, y los ejemplos de los que se dispone el sitio web. Para aplicaciones más complejas que requieran cierta interacción sin que haga falta que haya algún experto en programación que utilice JavaScript y que estudie la bibliotecas que ofrece play punto com.

Otra de las ventajas de esta plataforma, es que desde el propio navegador web se puede acceder al editor como de forma concurrente con distintos usuarios al mismo tiempo punto esto permite que de forma colaborativa se puedan desarrollar los proyectos de una forma más ágil además esta plataforma cuenta con un sistema de control de versiones, Lo que permite mantener un histórico de las modificaciones que se han realizado y editar distintas versiones de las implementaciones llevadas a cabo.

# Diseño

La aproximación seguida en este proyecto para hacer el diseño de las experiencias virtuales se describe a continuación. En primer lugar, los distintos equipos de trabajo que abordan conceptos similares en la materia de su asignatura, intentaron identificar cuáles son los conceptos más importantes y comunes a desarrollar en una experiencia. Este tipo de conceptos suelen ser conceptos fundamentales, pero que con suficiente abstracción como para que al estudiantado le cueste asimilarlo. Especialmente en los primeros cursos. De esta manera, se intentó enfocar el esfuerzo de desarrollo en los contenidos fundamentales para superar con éxito las distintas asignaturas.

El siguiente paso fue exponer cuáles son los tipos de recursos y explicaciones que suelen emplear cada uno de los/as profesores/as involucrados, para intentar identificar alguna metáfora que sea visual y que pueda permitir a los estudiantes y comprender de forma más efectiva los conceptos correspondientes.

El siguiente paso en el diseño de esta experiencia consistió en reunirse con el equipo artístico del proyecto para que pudiera reflejar la idea que me ponían en los grupos. Después de estas reuniones, los/as artistas propusieron distintos bocetos y diseños de personajes y entornos para intentar ajustarse a la descripción realizada por el profesorado.

Una parte importante y compleja a la hora de definir las distintas experiencias virtuales era intentar seguir una coherencia en la historia que se iba a presentar. La idea es que facilitara la comprensión global de todas las experiencias, aunque no fue fácil abordarlo.

Por otro lado, mantener la coherencia entre las distintas experiencias, es decir, utilizando los mismos símiles, utilizando el mismo tipo de personaje, o escenarios, permitieron reutilizar los recursos desarrollados que se generaron para las distintas experiencias. Esto es de especial importancia en el caso de que haya recursos limitados a la hora de buscar diseñadores o artistas para complementar la parte visual del proyecto.

## Diseño de boceto

En el proyecto Valerie, los artistas propusieron bocetos de la estética de la experiencia virtuales (Fig. 1). Después de estos bocetos, se desarrollaron unas hojas de personaje para el personaje principal (Fig. 2).



Fig 1. Boceto de personaje y escenario.

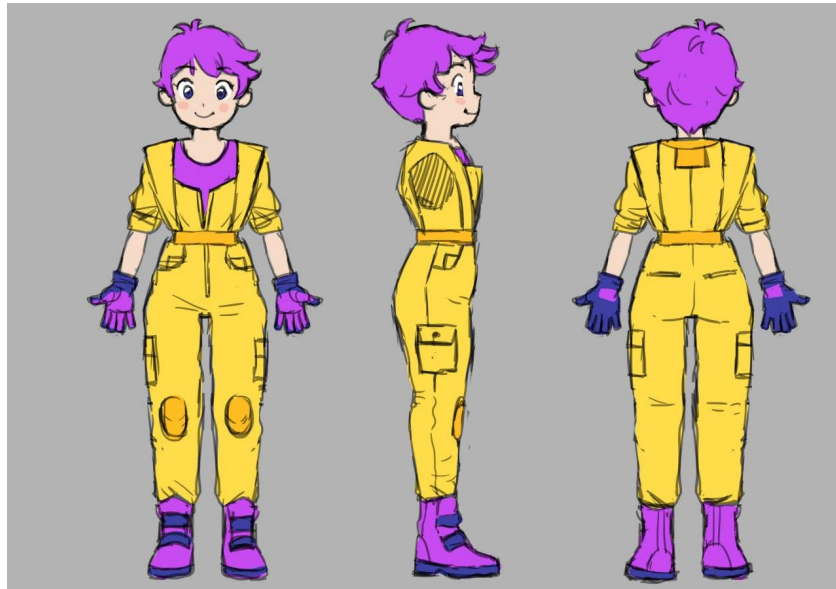


Fig 2. Hoja de referencia de personaje. CC BY-NC-SA 4.0 por Elena García Bravo.

A partir de este boceto se desarrolló un modelo 3D utilizando Blender y otro tipo de herramientas que son aunque requieren una especialización y un conocimiento avanzado en el modelado y animación 3D. Un boceto inicial de este modelado se puede apreciar en la figura 3.

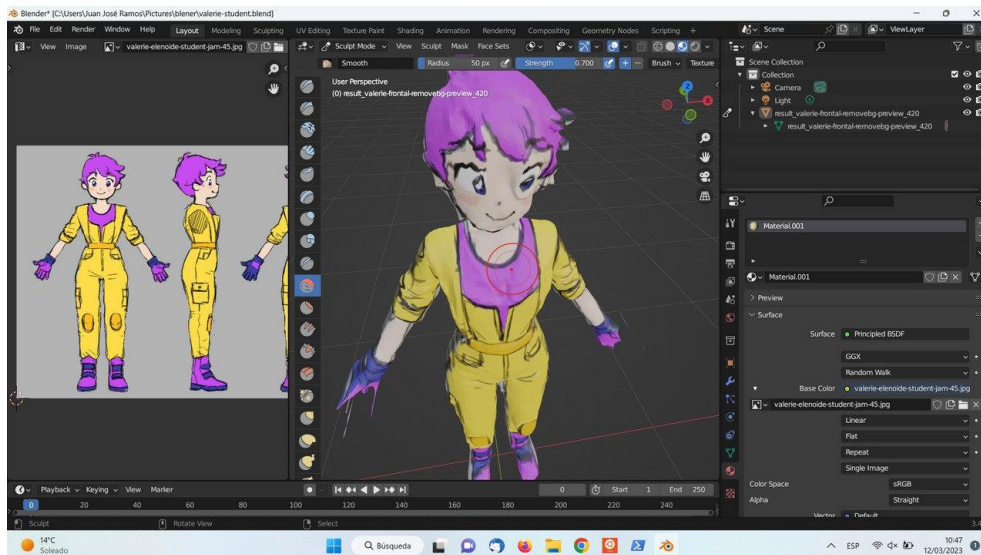


Fig 3. Boceto 3D basado en hoja de referencia.

No obstante una de las alternativas que están disponibles para los distintos grupos que quieran desarrollar experiencias virtuales utilizando modelos 3D y animaciones, que se utilizan algunos de los repositorios que ya existen y que tienen licencia abierta. Por ejemplo sketchfab (Sketchfab, inc.), entre otros.

Afortunadamente se contaba en el equipo de desarrollo con profesorado y estudiantado del ámbito artístico. Por eso, se ha intentado aprovechar esa fortaleza para poder diseñar contenido propio y personalizado para el proyecto.

Sin embargo, debido a la carga de trabajo que supone el desarrollo de distintos tipos de personajes en tan poco espacio de tiempo, para su modelado 3D y su animación se decidió explorar las distintas opciones que pueden ofrecer actualmente en distintas herramientas de inteligencia artificial. Esta aproximación permitió el desarrollo de distintos personajes de una forma ágil, aprovechando las nuevas tendencias en la inteligencia artificial generativa.

## Desarrollo de bocetos con inteligencia artificial

Utilizando la herramienta Automatic1111 (AUTOMATIC1111, 2023), el framework de estable difusión, la extensión de controlnet, y la boceto de referencia generado por la artista, se procedió al desarrollo de distintos personajes.

Para poder generar distintos personajes que se basaran en la estética diseñada por la artista, hizo falta utilizar una técnica llamada LoRA (Hu, 2021) por la cual se puede refinar parte de un modelo de *stable diffusion* basándose en las imágenes de referencia que se le proporcionen durante el entrenamiento. De esta manera, cuando se generan nuevas imágenes desde el sistema de *stable diffusion*, se pueden obtener imágenes similares a la original pero con distintas variaciones.

Una de las técnicas utilizadas para que los modelos generados fueran mucho más similares a la imagen de referencia original fue utilizar la técnica de controlnet (Zang & Agrawala, n.d., 2023) para poder indicar al modelo generativo que tenía que

seguir una estética similar de referencia a la que se le proporcionaba con la imagen del boceto. Concretamente, la implementación actual de controlnet permite utilizar como restricción una imagen de referencia. Además introduciendo una condición adicional en la que se proporcionaban la poses que debía tener la imagen generada para utilizarlo como hoja de referencia, controlnet proporcionaba la posibilidad de utilizar como criterio restricción que se ajustará a la poses que se proporcionaban en un formato similar a la de openposenet (ver Fig 4).

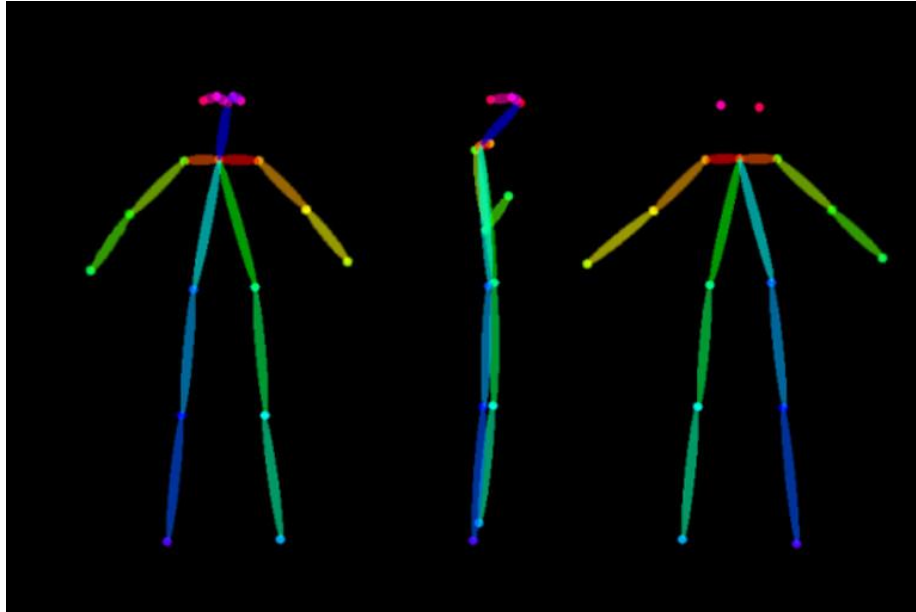


Fig 4. Poses de referencia para añadir restricciones en controlnet, y generar imágenes de referencia con las poses indicadas.

De esta manera después de la ejecución del fallo e intentos con la descripción de los distintos personajes, se podían obtener distintas hojas de referencia que pudieran ser viables para implementarlo en el videojuego, como se aprecia en la figura 5.

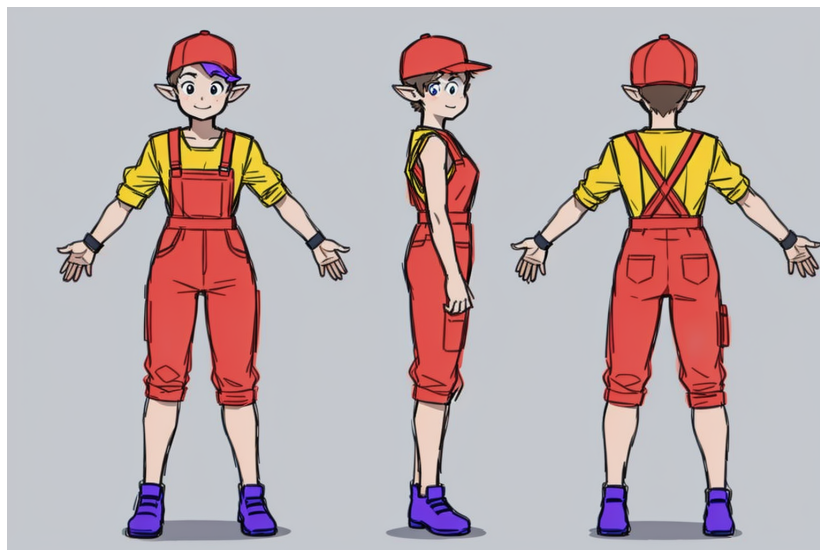


Fig. 5. Hoja de referencia de personaje generado con stable diffusion 1.5 y controlnet.

Nótese que la generación de este tipo de imágenes contiene un componente aleatorio, por lo cual no se podía obtener directamente en los personajes que se estaban solicitando. Por ello, sigue siendo necesaria la intervención de los/as artistas para poder seleccionar con el ese diseño era válido o coherente con el diseño estético original. Esta restricción de mantener una coherencia estética le da un valor adicional para que los usuarios lo perciban con una mejor calidad.

## Modelado 3D de los bocetos

Hay que tener en cuenta que a la hora de realizar una experiencia con realidad virtual o Realidad Aumentada los componentes que se visualizarán son modelos 3D. Sin embargo el paso de un boceto 2D a un boceto 3D no es trivial y necesita de una parte de modelado en el que se requieren reconocimiento avanzado o especializado en modelado y animación 3D.

La ventaja además de utilizar estos modelos 3D frente a los 2D es que en el último caso, hay que dibujar casi todas las animaciones, a no ser que se utilicen técnicas basadas en un esqueleto 2D o animación por partes. Pero en el caso de 3D sí que es más fácil definir una especie de esqueleto que se utiliza tanto en videojuegos como en animación convencional digital, y que permite modificar la postura de este modelo de forma más sencilla.

Una vez más ha intentado explorar las posibilidades que ofrecen las nuevas tecnologías relacionadas con inteligencia artificial, con el objeto de poder aplicarla para que agilice el desarrollo de este tipo de proyecto.

Una de las alternativas que se ha explorado ha sido utilizar el modelo generativo de modelos 3D basado en una imagen estática. Por ejemplo ICON (Xiu et al., 2022) es una de las propuestas científicas que permiten a partir de una foto con una pose de una persona. Genera una malla 3D que prediga cuál es el volumen que tiene ese personaje. El resultado es una 3D, aunque directamente no puede ser utilizada para incluirla al entorno del videojuego. La malla que genera es un poco burda, pero sirve como boceto inicial para poder seguir refinándolo en un entorno de modelado 3D, como se puede ver en la Fig. 6. Este modelado 3D se llevará a cabo en Blender, un software para modelado y animación 3D abierto con licencia gratuita .



Fig 6. Vista en programa de edición 3D Blender de una malla generada por ICON, a partir de un boceto 2D.

A la hora de integrar un modelo 3D en un videojuego una experiencia virtual como la que deseamos desarrollar, hay que tener en cuenta la capacidad de cómputo y memoria que dispone el dispositivo en el que se va a ejecutar. Por eso es deseable optimizar los recursos que se vayan a utilizar.

Para realizar esta optimización, un experto en el modelado 3D podría realizarlo con la herramienta adecuada como Blender. Sin embargo, se han explorado mecanismos que permitan hacer esto de forma ágil, y por personal menos experto. Se han probado optimizadores de topologías de mallas 3D. Concretamente, Instant Meshes (Jakob et al., 2015). Esta herramienta permite dar un objeto 3D de entrada, generando otro que sea equivalente, pero con una topología más estructurada y con menos vértices. Es decir, con una disposición de las caras que conforman la malla más ordenada. El resultado de aplicar este algoritmo, reduciendo el número de vértices y caras del modelo original puede verse en la Fig 7.

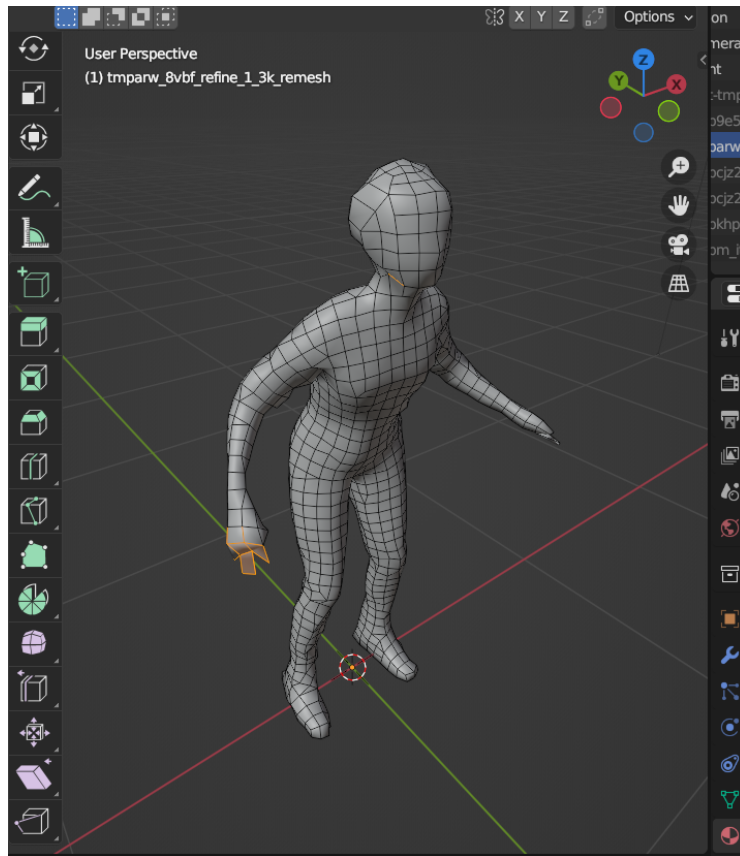


Fig 7. Malla resultante de aplicar la herramienta Instant Meshes.

Otro de los pasos en el modelado 3D es el texturizado del modelo. Para ello hace falta desplegar el mapa de UV del modelo. Este proceso básicamente consiste en definir la correspondencia de cada píxel de una imagen a cada punto de una malla 3D. En este proyecto se realizó de forma muy básica basado en la hoja de referencia de personaje y utilizando una proyección del mapa UV de textura desde las distintas vistas principal, trasera y lateral. Esta aproximación ofrece un texturizado que no es óptimo pero que es suficiente para la aplicación que nos ocupa, como se puede apreciar en la Figura 8.



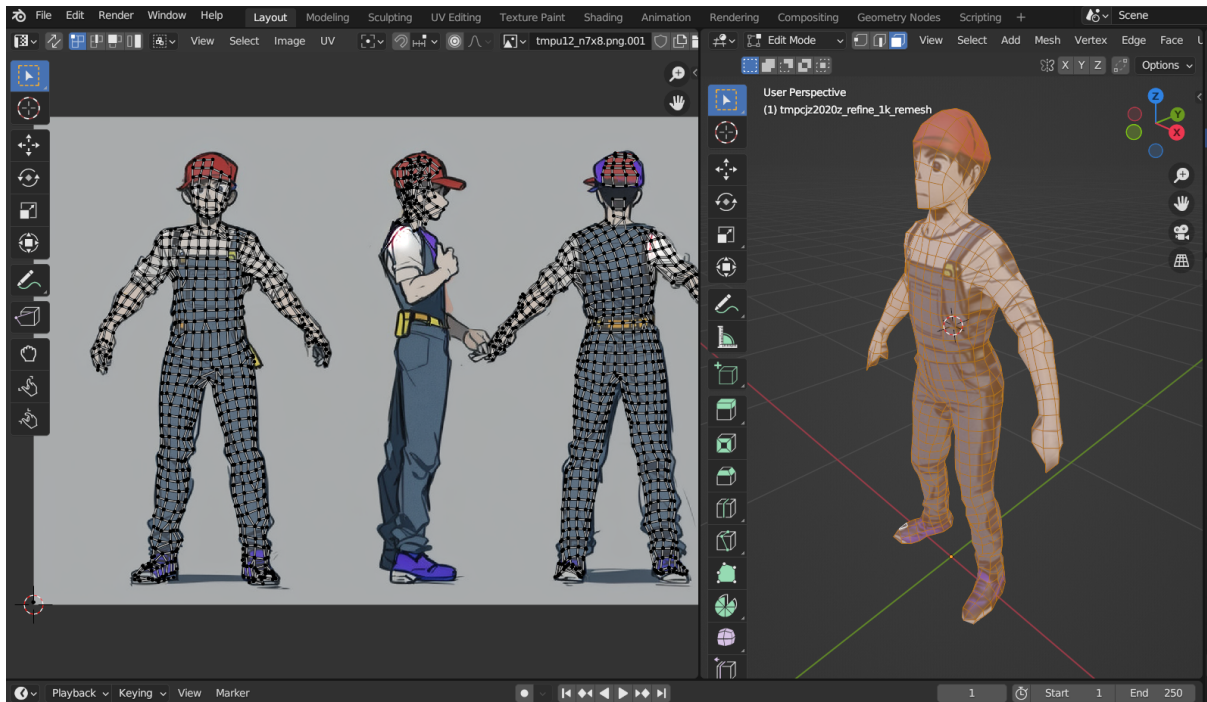


Fig 8. Ejemplo de despliegue de mapa de UV para el texturizado.

El siguiente paso para poder realizar la animación de un modelo 3D es el de generación de una *tramoya* (*rig*). Este paso consiste en asignar a un esqueleto básico que zona de influencia tiene cada hueso sobre la malla del modelo a animar. De esta manera se define la influencia que tiene a cada hueso en este objeto y cuando se animen dicho hueso eso se traducirá en la animación del personaje.

Este proceso de creación del esqueleto y asignación de los influencias que tiene sobre las distintas partes del objeto puede ser tediosa. Es por ello que se ha utilizado otra herramienta gratuita llamada [mixamo.com](https://mixamo.com), que permite generar dicha asignación de hueso aparte de un objeto de forma automática, requiriendo una intervención muy corta del usuario para definir algunas de las partes del cuerpo. Otra de las ventajas de este tipo de de esta herramienta es que es en línea, y que también cuenta con una galería de animaciones predefinidas que pueden aplicarse sobre cualquier muñeco.

Debido a las peculiaridades de este tipo de experiencia en lo que puede haber una animaciones de las que un personaje explica algún concepto al usuario de la experiencia, se ha procedido a integrar animaciones propias que no aparecían en la garantía de animaciones de la herramienta anterior. Para ello se pueden utilizar alguna de las herramientas que se han analizado para hacer animación basada en la captura de los movimientos de un personaje real. Este tipo de aplicaciones permiten a partir de un vídeo generar un esqueleto compatible con la herramienta que se ha descrito anteriormente, y que se puede utilizar para animar un personaje que ya tiene un esqueleto asignado.

Alguna de las herramientas que se han evaluado son [deepmotion.com](https://deepmotion.com), [Ples.com](https://ples.com) y [rokokoo.com](https://rokokoo.com). DeepMotion permite hacer una captura de movimiento a partir de un vídeo que tiene en cuenta no solamente las distintas partes del cuerpo sino también

dedos y gestos de la cara. Sin embargo es un servicio de suscripción mensual. Plask.ai es otra de las herramientas que permiten hacer una captura de ciertos minutos de animación. Pero por las pruebas realizadas, se observó que no era suficientemente precisa, con lo que generaban animaciones de menor calidad.

Por último se evaluó la herramienta rokokoo.com para video que permite hacer las mismas capturas a partir de un vídeo en el que se viera una persona realizando alguna acción. La ventaja de esta herramienta reside en que rokoko es en realidad una empresa que comercializa un traje para hacer captura de movimiento (MOCAP) con bastante más precisión, pero que ha incorporado la posibilidad de utilizar vídeos 2D para la capturas de movimiento que sean menos exigentes. Esto supone un punto de entrada al mundo de la captura de movimiento y la herramienta. Esta herramienta permite hacer una captura con menos errores de captura, y permite también exportar el resultado a formato que sean compatibles con los motores de animación que se han visto anteriormente.

Una vez que se tiene la captura de movimiento he portado a un esqueleto que es compatible con las herramientas que se utilizan para el modelo de animación 3D, es necesario asignar esta animación al modelo que se había utilizado con su tramoya o rig. Dado que no son exactamente iguales los dos esqueletos que se utilizan para el muñeco y el que genera la aplicación de rokoko, hace falta llevarlo a cabo un procedimiento que se llama *retargeting*. Este procedimiento lo que permite es copiar los movimientos de cada hueso de un esqueleto, y aplicarlo en cualquier otro personaje que tenga un esqueleto similar. Para ello se realiza un proceso de renombrado del nombre de cada uno de los huesos para ver cuál es la equivalencia entre uno y otro esqueleto. Esto es un proceso automático en la mayoría de los casos. Afortunadamente rokoko tiene una extensión gratuita que se puede utilizar en Blender para hacer de forma automática y esta asignación aunque requiere de cierta intervención de usuario.

La utilización de esta herramienta junto a la extensión de retargeting ha permitido que se puedan capturar los movimientos de los usuarios reales o los participantes en este proyecto a los muñecos que se han diseñado para el mundo 3D y la experiencia 3D, como se puede apreciar en la Fig 9.

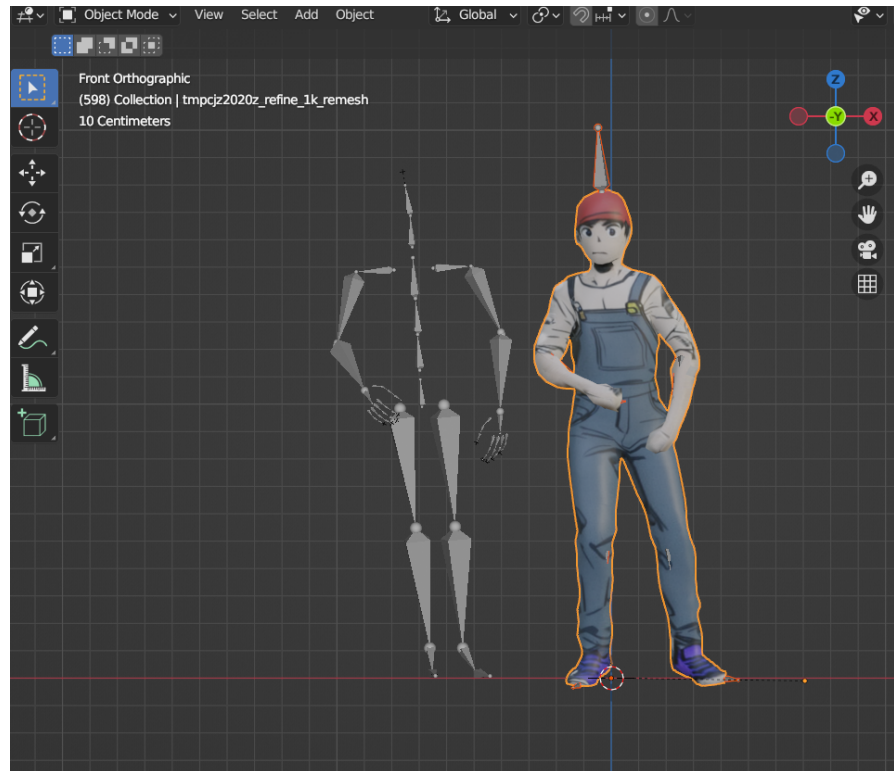


Fig. 9. Personaje con esqueleto para animación, con los movimientos capturados en el esqueleto de Rokoko (izquierda).

## Implementación de las experiencias

Para la implementación en nuestro caso se siguió la idea de crear distintos grupos de trabajo que hicieron una interpretación conjunta. Una de las ventajas de utilizar la herramienta PlayCanvas es que permite tener un editor compartido entre distintos usuarios, de forma que acceden concurrentemente al mismo proyecto, y pueden programar al mismo tiempo. En todos los casos hizo falta seguir los distintos pasos que se ha mencionado anteriormente en la parte de diseño personalizado a la experiencia que se pretende desarrollar en cada uno de la materia.

En este estadio del proyecto se han llevado a cabo algunas implementaciones con una interactividad bastante reducida. En esta sección se describen los procedimientos comunes a las distintas experiencias virtuales que pueden servir para el proceso y el desarrollo de distintas experiencias adicional.

La parte común que puede interesar a distintos tipos de experiencias virtuales se describen a continuación .

### Uso de la plataforma PlayCanvas

Para ello se puede utilizar desde la herramienta de Play canvas en la que hay que registrarse de forma gratuita.

Para cargar un modelo 3D con sus animaciones se puede subir cada modelo en un formato compatible como son GLB, GLTF y FBX . Estos son formatos estándares que permiten compartir modelos y animaciones entre distintos tipos de software de modelado y animación 3D, incluyendo a los motores de videojuegos. Estos ficheros se pueden generar desde la herramienta elegida en este proyecto, Blender.

Hay que tener en cuenta que en PlayCanvas, cuando se sube un fichero FBX o GLB se crean una serie de componentes que son básicos en este motor. En el caso de la experiencia que nosotros estamos utilizando es necesario utilizar el componente plantilla. Esta plantilla o *template* contiene los componentes que son necesarios para mostrar un muñeco y en su caso animarlo. En la Fig. 10 se muestra el entorno de desarrollo con un escenario de prueba.

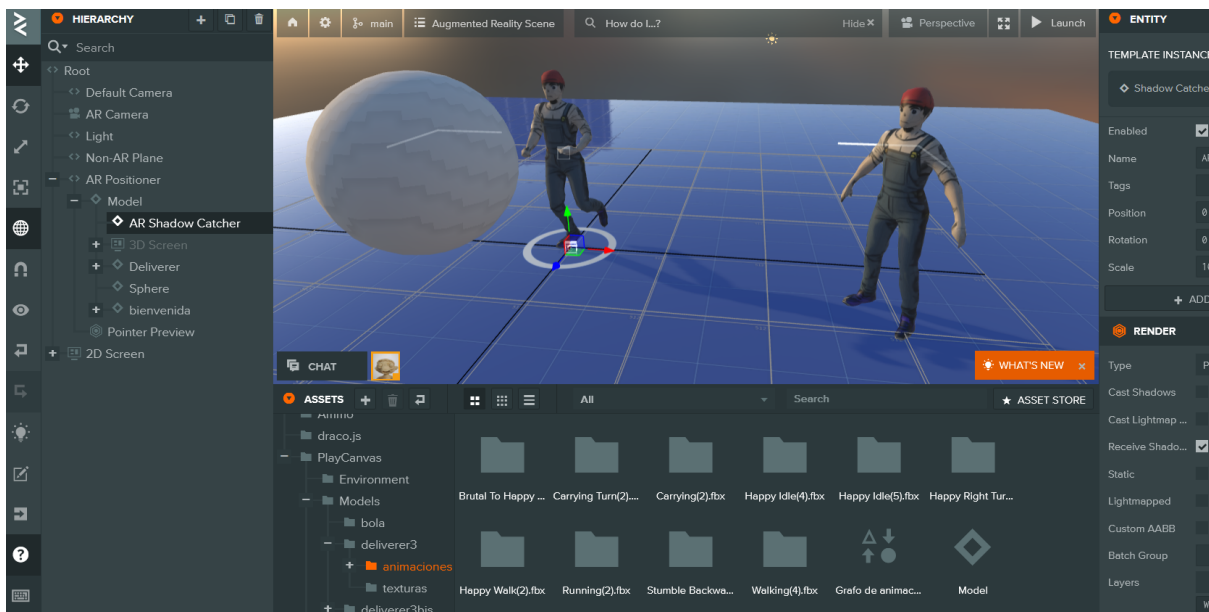


Fig 10. Entorno de desarrollo online de PlayCanvas.

Para poder añadir animaciones al personaje una vez integrado en el escenario, es necesario añadir un componente a la entidad que se ha introducido en que representa al personaje. Este componente es necesario se llama *Anim* y necesita a su vez que esté compuesto por un grafo de animaciones. Dicho grafo de animaciones consta de distintos nodos que representan a una animación concreta, y transiciones para pasar de una animación a otra. Por ello una vez que se ha creado un grafo y se ha asignado al componente de animación, es necesario arrastrar o asignar la cada animación a cada uno de los nodos que se hayan creado (ver Fig 11).

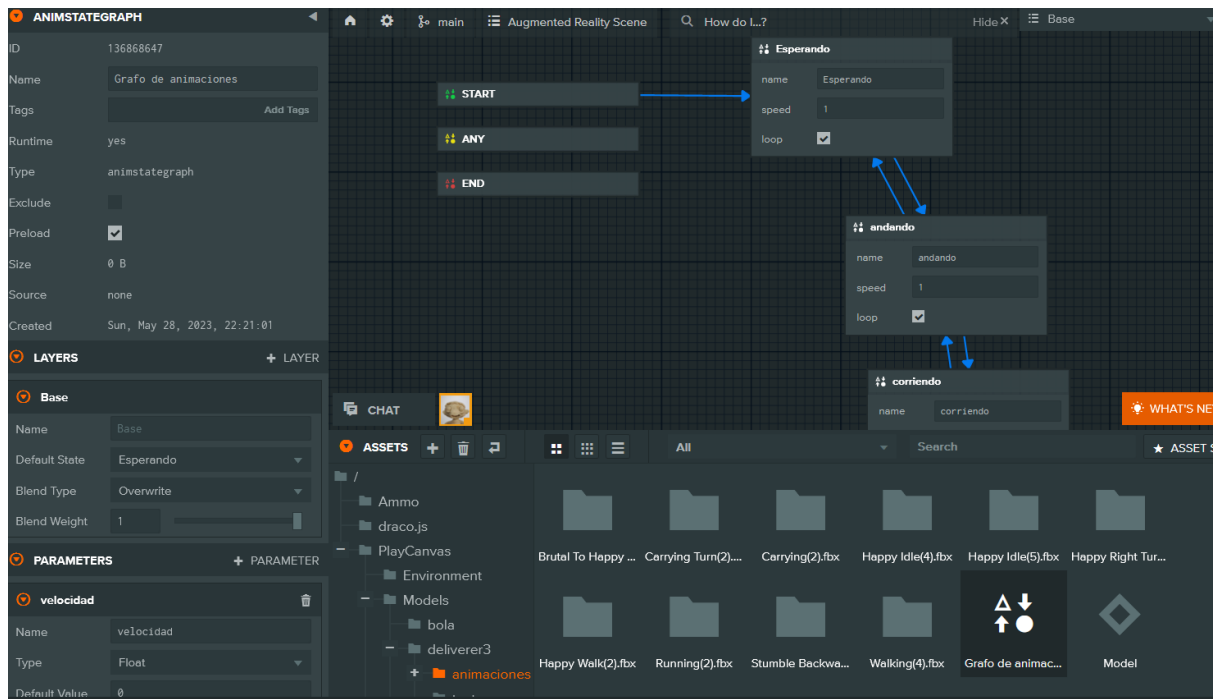


Fig 11. Grafo de animaciones para los personajes.

Para experiencias sencillas de realidad aumentada o realidad virtual, es suficiente con tener una animación para toda la sesión que se asigna al personaje. Por ello solo hace falta mantener un grafo de animación con un estado y el que a su vez contiene la animación que necesitamos.

## Programación de interactividad

Para conseguir una experiencia interactiva más rica en la que el usuario pueda interactuar con el entorno moverse, o por ejemplo, responder a cuestionarios que le propongan al personaje virtual. Para asignar scripts que permitan definir el comportamiento del personaje o del tipo de interacción, es necesario añadir un componente de script a la entidad que sea necesaria.

Hay que tener en cuenta que un script en PlayCanvas tiene una estructura definida.

Inicialmente, se crea en javascript un objeto con el nombre del script. A continuación, se pueden definir qué atributos tiene el script, de forma que pueda parametrizarse según sea necesario.

Luego que hay que insertar una sección de inicialización que se ejecuta cada vez que se cree el objeto que incluye el script.

Después del anterior, existe un una función de actualización que se ejecutaría en cada uno de los fotogramas y es donde se puede realizar la interacción que haya definido en la experiencia.

## Interacción VR/AR mediante webXR

En el estudio de funcionamiento de PlayCanvas se encontró la posibilidad de utilizar varias facilidades que aporta el motor para llevar a cabo la interacción en futura experiencia desarrollada en este motor.

Concretamente, para el entorno de VR con experiencias virtuales cabe la posibilidad de utilizar la técnica de *gazing*. Ya que se va a utilizar una carcasa como Google Cardboard portando un móvil, no es adecuado usar un mando adicional, para facilitar que mayor número de estudiantes pueda usarlo. El *gazing* consiste en definir un punto en el visor del usuario de forma que cuando ese punto se encuentra encima de algún objeto interaccionable, se pueda seleccionar. Para ello, hay que mantener la vista en ese punto durante el tiempo que defina en la experiencia. De esta forma no sería necesaria ningún tipo de control o mando adicional para poder disfrutar de experiencias que sean interactivas.

En el caso de realidad aumentada, se puede utilizar el uso de botones, ya que el motor permite soportar la detección de pulsaciones sobre objetos que aparezcan en el escenario.

## Conclusiones

En este proyecto se han desarrollado distintas experiencias de realidad virtual y realidad aumentada para explicar conceptos de ingeniería, de forma que sea fácilmente asimilable de una forma visual por el alumnado de distintas asignaturas.

Para ello se ha contado con un equipo multidisciplinar que ha tenido distintos roles durante el desarrollo del proyecto. A pesar de contar con profesionales entre el ámbito artístico como de programación etc, se ha intentado identificar posibles herramientas que permitan la automatización de alguno de estos procesos para poder elaborar experiencias similares a la desarrollada en este proyecto sin necesitar una especialización tan profunda en el diseño en arte o en programación.

Para el desarrollo de las experiencias se ha utilizado una herramienta gratuita llamada PlayCanvas que permite un desarrollo de experiencia interactiva y audiovisuales. Este motor ha resultado igual, y en algunos casos mejor, que los que motores se han utilizado de referencia para el desarrollo de videojuegos en otros entornos.

Como resultado, se ha obtenido una serie de experiencias tanto realidad virtual como en realidad aumentada que pueden ejecutarse a través del navegador web de un móvil o cualquier otro dispositivo. Como utilizan una tecnología estándar (webXR) se puede beneficiar a un número mayor de usuario que no tenga un dispositivo específico en realidad virtual o realidad aumentada. Además, en este documento se ha resumido el aprendizaje fruto del desarrollo del proyecto, para que otros equipos puedan aprovechar las herramientas y la plataforma utilizadas.

# Agradecimientos

Este proyecto ha sido parcialmente financiado por el Programa de Innovación y Buenas Prácticas Docentes de la UGR.

# Referencias

ar.js team. (2023). *ar.js*. AR.js - Augmented Reality on the Web.

<https://ar-js-org.github.io/AR.js-Docs/>

AUTOMATIC1111. (2023). *AUTOMATIC1111/stable-diffusion-webui: Stable Diffusion web UI*.

GitHub. Retrieved 2023, from

<https://github.com/AUTOMATIC1111/stable-diffusion-webui>

Crytek GmbH. (2023). *CryEngine*. Página principal de CryEngine. Retrieved mayo 1, 2023,

from <https://www.cryengine.com/>

Epic Games. (2023). *Unreal Engine*. Página principal de Unreal Engine.

<https://www.unrealengine.com>

Godot Engine team. (2023). *Godot Engine*. Página principal de Godot Engine.

<https://godotengine.org>

Hu, E. J. (2021, June 17). [2106.09685] *LoRA: Low-Rank Adaptation of Large Language*

*Models*. arXiv. Retrieved May 31, 2023, from <https://arxiv.org/abs/2106.09685>

Jakob, W., Tarini, M., Panozzo, D., & Sorkine-Hornung, O. (2015). Instant Field-Aligned

Meshes. *ACM Transactions on Graphics*. <https://github.com/wjakob/instant-meshes>

Playcanvas.com. (2023). *PlayCanvas WebGL Game Engine*. Página principal de

PlayCanvas. <https://playcanvas.com/>

Sketchfab, inc. (n.d.). *Sketchfab*. Sketchfab - The best 3D viewer on the web. Retrieved

May, 2023, from <https://sketchfab.com/>

Supermedium. (2023). *A-Frame*. Página principal de A-Frame. <https://aframe.io/>

Three.js team. (2023). *three.js*. Página principal de three.js. <https://threejs.org/>

Unity Technologies. (2023). *Unity*. Página principal de Unity. <https://unity.com/es>

World Wide Web Consortium. (2023). *WebXR Device API* (B. Jones, M. Goregaokar, & R. Cabanier, Eds.) [W3C Candidate Recommendation Draft]. WebXR Device API. W3C Candidate Recommendation Draft. <https://www.w3.org/TR/webxr/>

Xiu, Y., Yang, J., Tzionas, D., & Black, M. J. (2022). *ICON: Implicit Clothed humans Obtained from Normals*. *ICON: Implicit Clothed humans Obtained from Normals*. Retrieved May, 2023, from <https://icon.is.tue.mpg.de>

Zang, L., & Agrawala, M. (n.d.). Adding Conditional Control to Text-to-Image Diffusion Models. *arXiv*. <https://arxiv.org/abs/2302.05543>