

**Universidad San Jorge**

**Escuela de Arquitectura y Tecnología**

**Grado en Diseño y Desarrollo de  
Videojuegos**

**Proyecto Final**

**Waterpolo Taquigol**

**Autor del proyecto: Jorge Aguirregomezcorta Aina**

**Director del proyecto: Asier Quesada Corta**

**Zaragoza, 15 de junio de 2021**





Este trabajo constituye parte de mi candidatura para la obtención del título de Graduado en Ingeniería Informática por la Universidad San Jorge y no ha sido entregado previamente (o simultáneamente) para la obtención de cualquier otro título.

Este documento es el resultado de mi propio trabajo, excepto donde de otra manera esté indicado y referido.

Doy mi consentimiento para que se archive este trabajo en la biblioteca universitaria de Universidad San Jorge, donde se puede facilitar su consulta.

Firma

Fecha **15 de junio de 2021**

A handwritten signature in black ink, consisting of a large, stylized 'A' followed by a series of loops and a final flourish.



## **Dedicatoria y Agradecimiento**

Me gustaría dedicarle este proyecto a la familia de waterpolo del Centro Natación Helios. A los compañeros y entrenadores con los que entreno todos los días, con los que he sufrido y disfrutado de los mejores momentos de mi vida.

Doy las gracias también a Asier Quesada por aguantarme durante el transcurso del desarrollo y por hacerme dar todo lo que tengo para crear el mejor juego posible.

Finalmente, quiero darle las gracias a Jorge Echeverría por introducirme al mundo de la informática y los videojuegos, y por guiarme a lo largo de mi aventura universitaria.



## Tabla de contenido

<b>Resumen .....</b>	<b>1</b>
<b>Abstract.....</b>	<b>1</b>
<b>1. Introducción .....</b>	<b>3</b>
<b>2. Contexto .....</b>	<b>5</b>
<b>2.1. Waterpolo.....</b>	<b>5</b>
<b>2.2. Juegos móviles .....</b>	<b>7</b>
<b>3. Estado del Arte .....</b>	<b>11</b>
<b>3.1. Juegos de deportes .....</b>	<b>11</b>
<b>3.1. Herramientas ya implementadas .....</b>	<b>13</b>
3.1.1. <i>Shaders .....</i>	13
3.1.2. <i>Assets.....</i>	13
3.1.3. <i>Rigging y Animaciones.....</i>	13
3.1.4. <i>Efectos de sonido .....</i>	14
3.1.5. <i>Edición de videos .....</i>	14
<b>3.2. Unity.....</b>	<b>15</b>
<b>4. Objetivos .....</b>	<b>17</b>
<b>4.1. Desarrollo de un juego móvil .....</b>	<b>17</b>
<b>4.2. Control y seguimiento del desarrollo .....</b>	<b>17</b>
<b>5. Metodología.....</b>	<b>19</b>
<b>6. Desarrollo .....</b>	<b>21</b>
<b>6.1. Concept .....</b>	<b>21</b>
6.1.1. <i>Shader de agua.....</i>	21
6.1.2. <i>Modelaje del escenario .....</i>	22
6.1.3. <i>Implementación de una skybox.....</i>	26
6.1.4. <i>Flotación del balón .....</i>	27
<b>6.2. Segunda iteración – Prototype (v0.1).....</b>	<b>28</b>
6.2.1. <i>Lanzamiento del balón.....</i>	28
6.2.2. <i>Físicas del balón en movimiento .....</i>	30
6.2.3. <i>Rotación del balón.....</i>	31
<b>6.3. Tercera iteración – Alpha (v1.1) .....</b>	<b>32</b>
6.3.1. <i>Obtener assets de personajes .....</i>	32
6.3.2. <i>Añadir assets al fondo de la escena.....</i>	33
6.3.3. <i>Crear animaciones de movimiento.....</i>	34
<b>6.4. Cuarta iteración – Alpha (v1.2).....</b>	<b>36</b>
6.4.1. <i>Creación e implementación del algoritmo de lanzamiento y reposición.....</i>	36
6.4.2. <i>Diseño de la interfaz de usuario de la pantalla de juego.....</i>	39
<b>6.5. Quinta iteración – Beta (v2.1) .....</b>	<b>40</b>
6.5.1. <i>Creación e implementación del algoritmo de movimiento del portero.....</i>	40
6.5.2. <i>Creación e implementación del algoritmo de aparición de defensores .....</i>	42
6.5.3. <i>Creación e implementación del algoritmo de movimiento de defensores .....</i>	43
<b>6.6. Sexta iteración – Master Candidate .....</b>	<b>45</b>
6.6.1. <i>Diseño e implementación de los menús .....</i>	46
6.6.2. <i>Búsqueda y creación de efectos de sonido .....</i>	49



---

6.6.3.	<i>Edición e implementación de efectos de sonido .....</i>	<i>50</i>
6.6.4.	<i>Creación del tutorial .....</i>	<i>50</i>
<b>7.</b>	<b>Control y seguimiento del desarrollo .....</b>	<b>53</b>
7.1.	<b>Concept .....</b>	<b>54</b>
7.2.	<b>Prototype (v0.1).....</b>	<b>55</b>
7.3.	<b>Alpha (v1.1).....</b>	<b>55</b>
7.4.	<b>Alpha (v1.2).....</b>	<b>56</b>
7.5.	<b>Beta (v2.1) .....</b>	<b>56</b>
7.6.	<b>Master Candidate .....</b>	<b>57</b>
<b>8.</b>	<b>Estudio Económico .....</b>	<b>59</b>
8.1.	<b>Método económico .....</b>	<b>59</b>
<b>9.</b>	<b>Resultados .....</b>	<b>63</b>
9.1.	<b>Análisis del artefacto.....</b>	<b>63</b>
9.2.	<b>Desviaciones del proyecto.....</b>	<b>65</b>
<b>10.</b>	<b>Conclusiones.....</b>	<b>67</b>
10.1.	<b>Cumplimiento de objetivos .....</b>	<b>67</b>
10.1.1.	<i>Creación de un juego móvil .....</i>	<i>67</i>
10.1.2.	<i>Control del proceso de trabajo.....</i>	<i>69</i>
10.2.	<b>Futuras mejoras .....</b>	<b>70</b>
<b>11.</b>	<b>Bibliografía .....</b>	<b>71</b>

---



## **Resumen**

El mundo de los videojuegos cuenta con una gran cantidad de géneros que captan la atención de múltiples generaciones de jugadores. Uno de estos géneros es el de deportes, que incluye títulos de fútbol, baloncesto, tenis y muchos más. Son muchos los practicantes de estos deportes que disfrutan de juegos basados en sus disciplinas. Sin embargo, no existe ningún juego que los waterpolistas puedan jugar de la manera que los futbolistas juegan al FIFA.

El proyecto de Waterpolo Taquigol surge de dicha necesidad, proporcionar a los jugadores de waterpolo un juego con el que se puedan sentir identificados. Es decir, un juego de su deporte. Los objetivos del proyecto son el desarrollo de un juego móvil Android de waterpolo y el control y seguimiento de dicho desarrollo usando una metodología SCRUM.

El resultado del proyecto incluye la creación de un juego móvil de waterpolo usando tecnologías como Unity, Maya, y Blender. La aplicación generada cumple con los objetivos propuestos y proporciona a los waterpolistas un juego con el que se pueden sentir identificados.

## **Abstract**

The world of videogames features a lot of different genres that capture the attention of numerous generations of gamers. One of such is sports, which includes titles ranging from football and basketball to tennis and golf. There are a lot of athletes that enjoy playing games related to their sports. However, in the case of water polo players there are no videogames they can play the same way football players play FIFA.

Waterpolo Taquigol is created to cover this very necessity, allowing water polo players to play a game they can identify with. The objectives of the project are the development of a water polo mobile game for Android and the tracking of such development process using a SCRUM methodology.

The result of the project includes the creation of a mobile water polo game using platforms such as Unity, Maya, and Blender. The application created as a consequence of the project fulfills the proposed objectives and allows water polo players to finally play a game they can relate to.



## **1. Introducción**

Los videojuegos de deportes han sido parte del entretenimiento de los jugadores desde el primer día que surgió un videojuego. Siendo 'Tennis for Two', el primer juego de la historia, un simulador de tenis que corría en una computadora analógica y mostraba su contenido a través de un osciloscopio. Muchos fueron los títulos que trataron de copiar este último y crearon la era de juegos de pala y pelota, con el mítico 'Pong' siendo uno de ellos. Pasaron los años y con los primeros arcades aparecieron juegos de boxeo, béisbol y atletismo, siendo seguidos no mucho más tarde por títulos que introducían el fútbol y la copa mundial a las pantallas de los salones. Las décadas continúan avanzando y con la introducción a las videoconsolas no hay un solo deporte del que no se pueda disfrutar desde la comodidad que viene al estar sentado en un sofá.

Esta clase de juegos son realmente populares entre el público más joven y se encuentran entre las franquicias más famosas de todos los tiempos, pero sin embargo no hay un juego para cada deporte. Este dato es completamente entendible, las desarrolladoras se centran en los deportes más populares y los exprimen para conseguir vender millones de copias anuales. ¿Y qué pasa con los deportes minoritarios? Simplemente no tienen juegos, y si los tienen son alguno casi imposible de encontrar que forma parte de las leyendas que deportistas comentan en los vestuarios al acabar de entrenar. Ya se hablaba hace más de 10 años de un juego de waterpolo para una consola portátil que existía en Japón, pero nadie que ha pasado por el deporte aragonés puede decir como era ni como se jugaba. Estas son la clase de situaciones que hacen soñar a los que practican un deporte así, tener un 'FIFA' pero de lo que practican diariamente.

El objetivo del proyecto nace de esa misma necesidad, darle a un grupo de deportistas la oportunidad de jugar a algo que refleje su día a día, de la misma forma que hacen muchos otros. El juego se desarrolla para la plataforma Android, debido al gran número de dispositivos móviles que hacen uso del sistema operativo y la consideración de que la gran mayoría de gente tiene el teléfono móvil a mano en todo momento, lo cual permite llegar al proyecto a las pantallas del público sin ninguna dificultad. Otra de las necesidades a partir de las que surge el proyecto son los largos tiempos de espera que viven los deportistas que practican el deporte. Al no haber una gran cantidad de competición debido al número de clubs que hay en España, es muy común que para poder competir haya que viajar a otras ciudades o comunidades autónomas, muchas veces con horas de viaje entre ellas. Para ello, un pasatiempo móvil en el bolsillo ayudaría de alguna manera a minimizar el estrés generado por el viaje, y que mejor que visualizar lo que se va a practicar unas horas después.

La estructura del proyecto tomará la siguiente forma: Para empezar, el capítulo 2 introduce el contexto, donde se explican conceptos básicos sobre el deporte a simular, los juegos móviles y la motivación para unir ambos en un solo proyecto. En el capítulo 3, que presenta el estado del

---

arte, se recogen y discuten las diferentes anteriores realizaciones del campo de estudio del proyecto, como son productos similares, competidores, estrategias y demás. En el capítulo 4 se indican los objetivos planteados para la realización del proyecto y la correlación entre ellos. En el 5, se explica la metodología del proyecto. Es decir, cómo se ha realizado el trabajo, y como el flujo de alumnado-tutor funciona en las diferentes reuniones que se han dado a lo largo del desarrollo. A continuación, en el capítulo 6, se indica como se ha desarrollado la solución del proyecto, con sus consecuentes problemas, ideas y resoluciones. El capítulo 7 muestra el control de seguimiento del desarrollo, donde se analiza versión a versión el progreso del proyecto. Con la implementación estudiada, en el capítulo 8 se procede a la visualización estudio económico, en el que se refleja tanto el presupuesto necesario como el beneficio económico que puede derivar del proyecto. En el capítulo 9 se discuten los resultados obtenidos tras la finalización de la solución creada para el proyecto. Finalmente, el capítulo 10 presenta las conclusiones generales del proyecto, las cuales reflejan el grado de cumplimiento de los objetivos planteados y las futuras ampliaciones del proyecto.



## 2. Contexto

Para poder llegar a entender el proyecto, es preciso conocer o entender hasta cierto punto en que consiste el deporte del waterpolo, aunque sea de manera superficial. Seguida de la explicación se estudiará también el mercado de los juegos móviles y en concreto los que caen dentro de la categoría de simulación deportiva.

### 2.1. Waterpolo

El waterpolo es un deporte acuático en el que se enfrentan dos equipos compuestos por trece jugadores, de todos ellos solo puede haber siete por cada equipo en el agua al mismo tiempo. El objetivo del juego es sencillo, introducir el balón en la portería rival y recibir menos goles que el adversario.



*Figura 1. Origen del waterpolo*

Nadie sabe exactamente cómo surgió el deporte. Unos [1] dicen que su origen nació para entretener al público mientras éstos disfrutaban de competiciones de natación en las orillas de ríos o del mar. En estos encuentros el deporte consistía en dos equipos que montaban sobre barriles de madera usando remos e intentando introducir un balón en unas porterías que flotaban sobre el agua. La imagen que se obtenía al ver el deporte era similar al del polo tradicional, lo cual sirvió para darle el nombre de water-polo o polo de agua. La Figura 1 presenta lo que expertos consideran el origen del waterpolo.

El deporte evolucionó a lo largo del siglo XIX para llegar a asemejarse a una especie de rugby acuático, en donde sin ninguna regla y haciendo demostración de fuerza bruta, los jugadores trataban de introducir el balón sobre unas balsas colocadas en los bordes del campo de juego. No fue hasta finales de siglo cuando el *London Swimming Club* creó una serie de reglas para poder practicar rugby dentro del agua. El deporte fue poco a poco evolucionando hasta que en 1911 la FINA, Federación Internacional de Natación, hizo obligatorias una serie de reglas que

---

[1] (González, 2011)



serían mostradas a todo el mundo en los Juegos Olímpicos de 1920, y que serían desarrolladas año tras año hasta llegar a ser lo que conocemos actualmente. En la Figura 2 [2] se muestra una situación de un partido de waterpolo moderno.

Sin entrar en tecnicismos se describirán a continuación algunas de reglas, situaciones de partido y otras piezas de información importantes para apreciar el estudio que hay detrás de la implementación del juego, las cuales se desarrollarán en el apartado 6:

- Existen 6 posiciones distintas dentro del agua sin contar al portero. Estas posiciones forman una media luna en torno a la portería y cuentan con diferentes ángulos de lanzamiento y roles que adoptan los jugadores.
- Solo el portero puede coger el balón con dos manos. El resto de los jugadores solo pueden tocar el balón con una mano en todo momento, ya sea para lanzar o para bloquear.
- A lo largo de un partido ocurren cientos de faltas, que se consideran leves. Tras estas faltas los jugadores en posesión del balón pueden lanzar a portería, dando oportunidad a meter gol en la portería rival.



*Figura 2. Partido de waterpolo moderno*

Finalmente, para concluir con la introducción al deporte del waterpolo, se procede a explicar el juego detrás de la idea e implementación del videojuego. Para entrenar los lanzamientos, que son muy importantes en el deporte, los jugadores pueden jugar a algo que se llama taquigol. En el taquigol, un pequeño número de jugadores realiza lanzamientos en orden a una distancia

---

[2] (Ruiz, 2015)

mínima de la portería con el objetivo de lograr un número acordado de puntos. Estos puntos se obtienen o pierden de la siguiente manera:

- Gol es un punto más.
- Palo-gol es dos puntos más.
- Fuera directa es un punto menos.
- Una parada normal del portero ni suma ni resta.
- Una parada del portero con las dos manos es un punto menos.

Este sencillo juego de entrenamiento se practica en todas las piscinas donde haya un balón amarillo, independientemente de su lugar y de su tamaño. Es uno de los ejercicios más sencillos y divertidos que se pueden realizar dentro del agua, y como tal, sirve de inspiración e imagen para el desarrollo e implementación del proyecto móvil. Fácil de reconocer, es una manera de transportar la vida dentro de las piscinas a la pantalla de todo aquel que desee probarlo.

## **2.2. Juegos móviles**

La industria de los videojuegos [3] es un mercado en continua evolución del cual se estima que obtuvo más de 160 billones de dólares en ingresos el año pasado. Casi la mitad de dichos ingresos provienen de juegos móviles, uno de los sectores que más rápido crecen año tras año. Considerando tal crecimiento, no es ninguna sorpresa que solo el año pasado inversores llegarán a meter casi 10 billones de dólares en la industria, de los cuales se estima que en 2023 tendrán un valor de 250 billones. El éxito de la industria suele ser atribuido a la evolución del desarrollo móvil y los smartphones, pero los primeros juegos móviles datan de hace mucho más tiempo.

El primer juego para móvil surgió en 1994 para el Hagenuk MT-2000, que llevaba preinstalada una versión del Tetris. Ya en 1997 salió el clásico Snake de Nokia, marcando el inicio de lo que se convertiría en una exitosa y lucrativa industria.

Desde el lanzamiento de los primeros pioneros, miles han seguido su camino, llegando a más de 900.000 juegos activos en la App Store y casi 600.000 en la Play Store. Cuando Snake fue introducido en 1997, se convirtió en un fenómeno a nivel mundial gracias a su simple forma de jugar, su adictivo tema y el hecho de que era apto para todos los públicos.

Siendo los primeros juegos portátiles con un gameplay sencillo y sesiones cortas, Tetris y Snake son los predecesores del género que se conoce como híper-casual, y que tan popular es hoy en día en las tiendas de juegos.

En la Figura 3 [4] se muestra el diseño original de Snake junto al Nokia 3310.

---

[3] (PubNative, 2020)

[4] (Brown, 2020)





*Figura 3. Diseño del Snake original en Nokia 3310*

Entrando en el siglo XXI muchos han sido los juegos que habiendo seguido los pasos de Tetris y Snake se han llegado a tener una popularidad de nivel mundial, entre los que se encuentra Space Invaders. Una de las metas durante estos inicios fue la salida al mercado del Nokia N-Gage, que permitía jugar a una gran cantidad de títulos, incluidos los primeros en 3D.

No fue hasta 2008, un año después de la salida del primer iPhone, que la AppStore fue mostrada al público, y con ella, una nueva era de juegos móviles. Ese mismo año solo había 500 aplicaciones, incluyendo clásicos como Texas Hold'em y Super Monkey Ball. La introducción del iPhone en 2007 hizo posible el uso de controles táctiles, y con ello Angry Birds, siendo el primer juego realmente popular en hacer uso de ellos y convertirse en una franquicia en muy poco tiempo, lo cual colocó el listón de los juegos del año 2010 en adelante. Con el comienzo de la nueva década muchas compañías trataron de usar aplicaciones móviles como extensiones de pantalla para sus juegos, siendo claros ejemplos Ubisoft y Rockstar, aunque la iniciativa no llegó a despegar. En la Figura 4 se muestra la línea temporal de juegos móviles.



*Figura 4. Línea temporal de juegos móviles*

En 2012, uno de los juegos con mayores ingresos de la historia fue introducido a la AppStore, Candy Crush. Seguido por otro gran éxito, Clash of Clans. Conforme aumenta el número de juegos



lanzados en las tiendas, nueva tecnología es usada para diversificar los productos y resaltar sobre la competencia. Un ejemplo claro es el uso de realidad aumentada que Pokémon Go utiliza como mecánica principal. Hoy en día existen incontables géneros que uno puede encontrar en tiendas de aplicaciones, con diferentes formas de abordar la clasificación de éstos. El sorprendente desarrollo de juegos ha creado una audiencia que no solo se basa en gamers, sino gente fuera de este grupo que se siente atraída hacia juegos hiper-casuales.

En 2021 existen más de 6 millones de aplicaciones disponibles al público. De las cuales la mayoría se pueden obtener a través de Google Play. El rápido crecimiento de las tiendas de aplicaciones es equivalente al incremento global de juegos móviles en el mercado. A pesar de ello, debido a este crecimiento, son muchas las aplicaciones que no logran sobrevivir durante mucho tiempo. Encontrar la mejor estrategia de monetización puede ser un gran problema para una gran cantidad de aplicaciones que, en muchos casos, nunca llegan a lograrlo. Casi un 94% de aplicaciones adquiere sus ingresos a través de publicidad y el número aumenta a diario. App Annie, una aplicación de analítica del mercado de aplicaciones, reporta que dicha cifra aumentó un 6% en 2019 entre los mejores 1000 juegos tanto de la App Store como de la Play Store.

La mayoría de los juegos en estas tiendas son gratis, o tienen un modelo de negocio freemium, que consiste en darle al cliente un servicio gratuito que puede ser mejorado a través de pago para obtener mayores beneficios. Conseguir monetizar estos juegos [5] implica vender productos dentro de la aplicación o mostrar anuncios cada cierto periodo de tiempo. Y al igual que la industria, la tecnología para anunciar dentro de las aplicaciones evoluciona rápidamente.

La Figura 5 muestra un ejemplo de monetización en juegos móviles a través de la publicidad in-game.



*Figura 5. Ejemplo de monetización a través de publicidad*

[5] (Admixer, 2020)



---

### **3. Estado del Arte**

Para llegar a entender el desarrollo del juego móvil es fundamental conocer los antecedentes técnicos y tecnológicos que existen en el campo de los videojuegos. Estos incluyen tres puntos esenciales enfocados desde la siguiente manera. La primera es el estudio del mercado de los juegos móviles de deportes. La segunda trata sobre la implementación de diferentes elementos en el videojuego a través de herramientas ya creadas, siendo los más importantes el desarrollo gráfico de elementos 3D, sonido y demás. Por último, se hace hincapié en el motor utilizado, patrones de diseño usados y otras herramientas técnicas.

#### **3.1. Juegos de deportes**

A día de hoy el género de los juegos de deportes [6] está dominado por grandes empresas, como EA Sports y 2K Sports. La práctica que usan es la de obtener licencias de ligas nacionales para producir sus videojuegos. Estas incluyen ligas de fútbol de todo el mundo, la NBA, NHL, NFL y muchas otras competiciones mundialmente conocidas. Año tras año sale al mercado una nueva versión de estos títulos y, a pesar de que muchas franquicias tienen décadas de antigüedad, los juegos son bien recibidos por el público y obtienen millones de ventas en las primeras semanas nada más haber sido hechos públicos.

Con el mercado tan dominado por grandes corporaciones el género se hace difícil de abordar para muchos desarrolladores. A pesar de tal dominio, muchos jugadores prefieren pasar el tiempo con juegos más simples y baratos en páginas que ofrecen juegos en línea, como es el caso de Minijuegos [7]. Esta clase de productos no solo triunfan en la web. En la Play Store, la mayoría de los juegos de la categoría de deportes son desarrollados por compañías más pequeñas, y no de EA o 2K como se podría pensar.

Este hecho tiene que ver con el éxito de los juegos híper-casuales [8]. En la pantalla del móvil los jugadores no buscan complejos simuladores, sino que los productos sencillos, fáciles de usar y, sobre todo gratis se encuentran en la cabeza de carrera, recibiendo puntuaciones y descargas superiores a las de grandes desarrolladoras.

La Figura 6 [9] presenta algunos de los juegos más populares de la Play Store en la categoría de deportes.

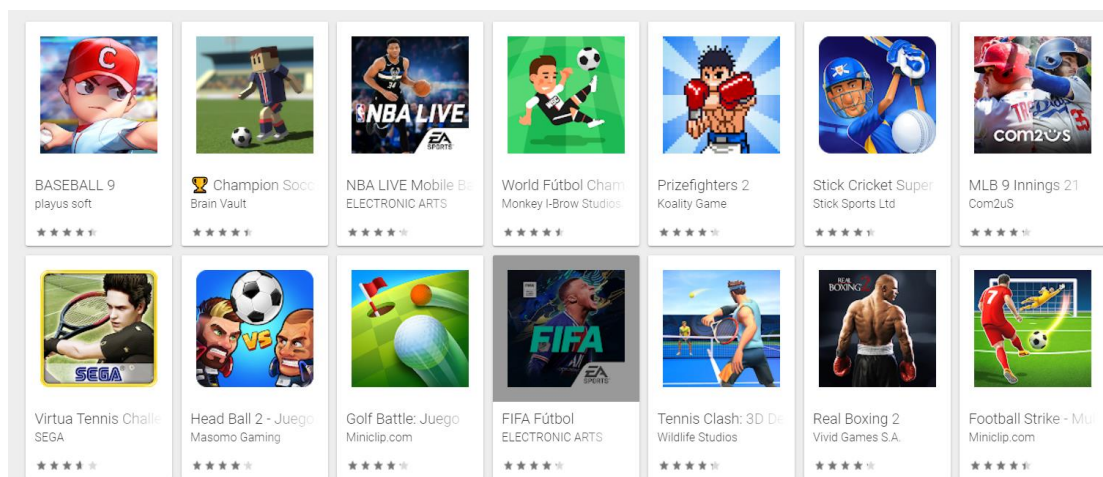
---

[6] (Wikipedia, 2021)

[7] (Mini Juegos, 2021)

[8] (Wikipedia, 2021)

[9] (Google, 2021)



*Figura 6. Juegos de deportes en la Play Store*

Entre estos títulos se encuentran una gran cantidad de simuladores de faltas. Un género de deporte donde el jugador hace lanzamientos a portería o canasta desde posiciones sin movimiento. Estos juegos tienen una gran popularidad ya que unen simplicidad con deportes, una mezcla sencilla pero muy eficaz de cara a obtener descargas.

Entre ellos, se encuentra uno en específico llamado New Star Soccer. Este título, que lleva 7 años en el mercado, empezó como un sencillo simulador de fútbol donde el jugador tiene que escoger que parte del balón quiere golpear para obtener diferentes disparos y así marcar gol. Con una simple premisa ahora tiene más de 2 millones de descargas, y no es el único caso de un pequeño juego siendo realmente exitoso. Los juegos de deporte generan descargas, gustan a quienes los juegan e impulsan un género tan antiguo como el primero.

Sin embargo, a pesar de ser un mercado donde los juegos casuales triunfan, sigue estando compuesto por decenas de miles de títulos que compiten entre todos. Para conseguir el éxito, es preciso adoptar estrategias que hagan que un producto pueda diferenciarse del resto de juegos. Una de ellas, muy recomendada por algunos profesionales [10], es la de atacar un nicho específico del mercado. Conocer de primera mano lo que un posible cliente quiere o espera de un producto hace que el desarrollador tenga una ventaja que no poseería en otros casos. Seguir esta estrategia puede diferenciar un producto de muchos en un mercado con más de 500.000 aplicaciones similares.

Esta forma de afrontar el mercado se usa en el proyecto desde antes del comienzo de la implementación, centrando el objetivo de este en torno al mercado de juegos de waterpolo, el cual a día de hoy no está nada explotado.

---

[10] (García, 2020)

### **3.1. Herramientas ya implementadas**

Para construir un videojuego existen múltiples campos que el desarrollador debe comprender, y para los que los profesionales se especializan. Siendo algunos de ellos los modeladores, artistas conceptuales, animadores, compositores y mucho más. Abordar el desarrollo de algunas de estos roles sin dominarlos previamente requiere el uso de herramientas ya creadas y optimizadas. Waterpolo Taquigol no es diferente y hace uso de programas, frameworks y tutoriales para conseguir implementar dichas tareas, entre las que se encuentran:

#### *3.1.1. Shaders*

Una de las características que primero se aprecian en el proyecto y la primera implementación que se realiza en el desarrollo es un shader que emula un cuerpo de agua. Para lograrlo se hace uso del canal de renderizado universal que Unity ofrece a sus clientes. Este canal permite renderizar de forma abierta y flexible los gráficos que se usan en un videojuego, pudiendo escalarlos dependiendo de la plataforma para lograr ese gran rendimiento tanto en pc como en móvil.

Más adelante, en el capítulo 6 se explica detalladamente como se implementa el shader y las decisiones que se toman en sus últimas versiones. Lo que es importante de comprender en este punto es que toda la implementación sigue un tutorial creado por Brackeys [11] donde explica los pasos necesarios para lograr un efecto similar.

#### *3.1.2. Assets*

Aunque todos los assets del escenario del juego se hacen a mano, lo mismo no ocurre en 3 casos concretos. Estos son el diseño de los personajes, los árboles y las nubes. Al igual que el shader, el diseño de personajes requiere de un punto de explicación que se encuentra más adelante en la memoria, pero la base del personaje se obtiene de un producto ya desarrollado. Lo mismo ocurre en los otros dos casos comentados, y el creador de ellos es Synty Studios [12], un publicador de assets en 3D de los cuales la universidad ha adquirido dos paquetes. Uno de ellos es Polygon nature, mientras que el otro es Polygon Sci-Fi Space. Dichos paquetes contienen una inmensa cantidad de modelos que se pueden obtener y que encajan perfectamente en el proyecto.

#### *3.1.3. Rigging y Animaciones*

Rigging [13] es una técnica usada en la animación de esqueletos para poder representar un modelo 3D haciendo uso de una serie de huesos digitales interconectados. Es decir, crear

---

[11] (Brackeys, SIMPLE CARTOON WATER in Unity, 2019)

[12] (Synty Studios, 2021)

[13] (Petty, 2021)



esqueletos para personajes. Al ser una técnica usada por todos, o sino la mayoría de los estudios para desarrollar animaciones, existen una gran cantidad de programas para conseguir implementarla.

Uno de ellos es Maya, un programa especializado en los gráficos 3D desarrollado por Autodesk. Gracias a Maya y a su gran cantidad de documentación, que a veces toma la forma de videos, se consigue completar una tarea que de cualquier otra manera podría resultar mucho más complicada. Además, ser un gran motor de gráficos permite realizar otras labores del proceso dentro de Maya, pudiendo utilizar el resultado del rigging para construir las animaciones.

En el caso de las animaciones, también se siguen guías de desarrollo ya establecidas para obtener resultados de forma rápida y eficaz. Éstas también vienen en formato de video tutorial y permiten visualizar y entender conceptos fácilmente. Finalmente, para exportar las animaciones creadas se utiliza un formato usado tanto por Maya como por Unity, obteniendo el resultado del trabajo en el motor sin ninguna complicación. En la Figura 7 se muestra el logo de Autodesk Maya.



*Figura 7. Logo de Autodesk Maya*

#### *3.1.4. Efectos de sonido*

Uno de los elementos más importantes de cara a la interacción con el usuario son los elementos de sonido. De forma similar a la creación de assets en 3D, la mayoría de los elementos de sonido del proyecto se crean de manera autónoma, haciendo uso de grabadoras y otras herramientas. Pero en algunos casos se hace uso de assets ya creados. Elementos de este estilo son obtenidos de la librería de Epidemic Sound [14], la cual tiene un convenio con la Universidad San Jorge donde los estudiantes pueden hacer uso de ella en sus proyectos.

#### *3.1.5. Edición de videos*

Finalmente, para acabar la explicación de las herramientas ya creadas que se usan en el desarrollo se introduce a Blender. Blender es un programa informático especializado en el modelaje, la animación y la edición de video, razón por la cual se utiliza en el desarrollo de Waterpolo Taquigol. Una de las últimas tareas del proceso requiere la edición de videos y Blender ofrece todas las necesidades de forma fácil de usar.

---

[14] (Epidemic Sound, 2021)



### 3.2. Unity

Unity es una plataforma de desarrollo para crear juegos, experiencias, modelos y diseños en 2D y 3D. Es sencillo de usar y posee una gran comunidad y desarrolladores que mejoran el producto a diario. Además, es el motor que se utiliza en la escuela de ingeniería para aprender sobre el diseño de videojuegos en 3D, lo cual lo hace ideal para el desarrollo de este trabajo.

Como se comenta anteriormente, la funcionalidad que Unity ofrece para optimizar gráficos en múltiples plataformas hace que el juego aumente su calidad visual de forma considerable. El campo gráfico es muy importante para que el cliente pueda interactuar con el producto, y Unity consigue acercar el juego al jugador gracias a su laborioso trabajo en los últimos años.

En cuanto a lo que el rendimiento se refiere, Unity ofrece una herramienta llamada Profiler [15] que ayuda a leer información sobre la aplicación en tiempo real. Esta característica de Unity puede ser usada también en el desarrollo móvil conectando un cable USB al dispositivo deseado. El Profiler reúne y muestra la información sobre el rendimiento de la aplicación en áreas definidas, como son la CPU, memoria, renderizado y audio. Lo cual lo convierte en una herramienta muy útil a la hora de identificar áreas en las cuales la aplicación puede ser mejorada.

Por último, otra de las herramientas de las que se hace uso durante el desarrollo es la consulta en los foros de Unity, los cuales están llenos de respuestas para poder solucionar hasta el mayor de los problemas. Lo que lo convierte en una de las mayores características que el motor posee. En la Figura 8 se muestra el logo de Unity.



*Figura 8. Logo de Unity*

---

[15] (Unity, Profiler overview, 2021)





## **4. Objetivos**

Para introducir el apartado en el cual se detallan los objetivos del proyecto es preciso reiterar, que tal y como se comenta en la introducción, la misión del proyecto es la de proporcionar a jugadores de waterpolo un videojuego dirigido especialmente para ellos.

El proyecto tiene como objetivos principales:

1. El desarrollo de un videojuego móvil para Android
2. Control y seguimiento del desarrollo del videojuego

A continuación, se detallan más a fondo las características y razonamiento detrás de los objetivos y los subobjetivos que componen cada uno de ellos.

### **4.1. Desarrollo de un juego móvil**

La creación de un juego para móvil requiere comprender sus diferencias con un juego en otra plataforma más conocida, como pueden ser las consolas o el PC. Los juegos móviles, como se explica en el apartado 2, están en constante desarrollo y compiten con cientos de miles en plataformas donde solo se anuncian unos pocos. Para lograr una posible y futura distribución el juego tiene que poseer una calidad mínima y unas características que lo hagan capaz de rivalizar o sobresalir en un mercado muy competitivo. De esa manera el producto final debe tener la mejor forma posible, y conseguirlo es uno de los objetivos principales del proyecto.

Los subobjetivos que componen el desarrollo del juego son los siguientes:

- Desarrollo de la idea del juego.
- Creación de un escenario 3D.
- Diseño y creación de personajes con animaciones.
- Diseño y creación de elementos interactivables con el jugador.
- Diseño de la interfaz de usuario.
- Implementación y desarrollo de elementos sonoros.
- Desarrollo de algoritmos de inteligencia artificial para los personajes no jugables.
- Desarrollo de algoritmos de rejugabilidad y gestión de curvas de dificultad.
- Desarrollo de mecánicas sencillas, entendibles y satisfactorias para el jugador.

### **4.2. Control y seguimiento del desarrollo**

Muchas veces obviado y con su importancia subestimada, se encuentra el control y seguimiento del desarrollo en los proyectos. Esta clase de herramientas ayudan a saber qué se está haciendo en todo momento y el ritmo al que va el desarrollo, obteniendo una mayor organización y rendimiento, de esa forma elevando la calidad del trabajo final.

Usando un diagrama burn-up, se procede a estudiar y controlar el trabajo que se realiza en el proyecto. Para ello, se crean fases de trabajo asociadas a diferentes versiones del proyecto donde se estipula su carga dentro de éste. Este diagrama ayuda a la visualización del progreso y a medir

---

la diferencia entre el trabajo observado y el trabajo esperado, de esa forma obteniendo la desviación del proceso de trabajo.

El objetivo de esta fase de trabajo es mantener este proceso de trabajo por encima de una desviación no optimista del 7%. Toda la información relacionada con este objetivo se profundizará en el apartado de la metodología y se discutirá a medida que se desarrolle la implementación de la demo en el apartado 6.

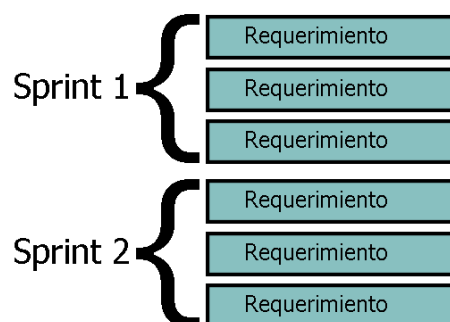


## 5. Metodología

La metodología hace referencia al conjunto de procedimientos que se siguen para alcanzar los objetivos de una tarea, proyecto o cualquier investigación de origen científica. Es la herramienta que hace que el desarrollo del proyecto siga siempre un curso estable y controlado.

Uno de los puntos más importantes de la primera reunión entre el tutor y el alumno es la elección de la metodología a usar durante el desarrollo del proyecto. Tras estudiar la naturaleza de este, se elige utilizar una de carácter ágil, y entre las más utilizadas se acaba haciendo uso de SCRUM. Con SCRUM se divide el proyecto en 6 diferentes fases de desarrollo, cada una con diferentes subobjetivos que deben ser analizados, desarrollados y testeados. Dichas fases tienen una duración de entre 3 y 4 semanas, y concluyen en pequeñas reuniones donde se revisa el progreso, se analizan las razones detrás de las tomas de decisiones y se deciden futuras implementaciones y correcciones para futuros ciclos de trabajo.

Una de las herramientas más importantes que posee SCRUM es el uso de la pila de producto. La pila de producto es una lista ordenada que contiene todas las mejoras necesarias para incrementar la calidad de un producto. La manera más común de expresar características de un producto en la pila de un proyecto ágil es a través de historias de usuario. Estas historias son cortas y simples descripciones que contienen la funcionalidad deseada desde el punto de vista del cliente. Una vez que la pila de producto está lo suficientemente completa se puede elegir que requerimientos se deben incluir en un sprint. Estos requerimientos pueden variar a lo largo del sprint, o incluso dar lugar a nuevos requerimientos para futuras iteraciones. La Figura 9 muestra la relación sprint-requerimiento.



*Figura 9. Relación sprint-requerimiento*

Para medir el progreso del trabajo se utiliza una herramienta visual conocida como gráfico burn-up [16]. Esta herramienta resulta muy útil trabajando con SCRUM, ya que ayuda a tomar decisiones realistas acerca del trabajo pendiente y las expectativas de este. Para construir el gráfico se elige primero el número de iteraciones que va a haber y luego se construye la pila de

---

[16] (Beas, 2015)

---

producto. A cada una de las tareas se le asigna luego un punto de historia, o *storypoint* en inglés. Estos puntos sirven para estimar la duración y peso de cada tarea. En el caso del proyecto, cada *storypoint* es una hora de trabajo. Con los cálculos de cada iteración calculados se puede apreciar cuantas horas de trabajo hacen falta para cada sprint, y así calcular una desviación positiva y negativa del flujo de trabajo del proyecto. Tal y como se comenta en el anterior capítulo, uno de los objetivos del proyecto es mantener la desviación del gráfico burn-up por encima del 7%.

## 6. Desarrollo

El capítulo de desarrollo explica cómo se ha implementado la solución del proyecto. Para poder transmitir al lector de mejor manera el curso de este, se divide el capítulo en subapartados correspondientes a las diferentes iteraciones que ha habido. En ellos se discute sobre los objetivos o tareas propuestas en las reuniones y como se enfocan, analizan, y desarrollan, de esa manera pudiendo comprender el flujo de implementación del proyecto de la mejor forma posible.

### 6.1. Concept

La primera fase de desarrollo del proyecto comienza con una reunión de estudio entre el alumno y el tutor, Asier Quesada. En esta reunión se habla sobre la idea del juego y la forma en la que el alumno propone abordar la implementación de este. La idea principal trata sobre la creación de un juego para móvil en el que los jugadores puedan hacer lanzamientos de falta en una piscina, simulando una situación de partido de waterpolo. La implementación se propone realizarla a través del motor de Unity, que es la herramienta que mejor conoce el alumno. Además, tras haber cursado asignaturas de modelaje en 3D, el alumno propone al tutor realizar el modelaje de todos los assets requeridos para el funcionamiento del juego, de esa manera obteniendo un resultado uniforme y personal.

Tras la discusión sobre las diferentes facetas del juego se concluye la reunión eligiendo los subobjetivos que se deben implementar a lo largo de la primera fase de desarrollo, que se centra mayormente en el modelado de objetos para crear así el concepto del juego, y poder estudiar si se continúa con la idea propuesta o se empieza un nuevo proyecto.

Entre los subobjetivos de la primera versión del juego se encuentran:

- Shader de agua: 2 *storypoints*
- Modelaje del escenario: 20 *storypoints*
- Físicas de flotación del balón: 4 *storypoints*

Para concluir la reunión se añade el valor de *storypoints* totales al gráfico burn-up, para así poder comprobar en la siguiente reunión el cumplimiento de estos y el progreso realizado.

#### 6.1.1. Shader de agua

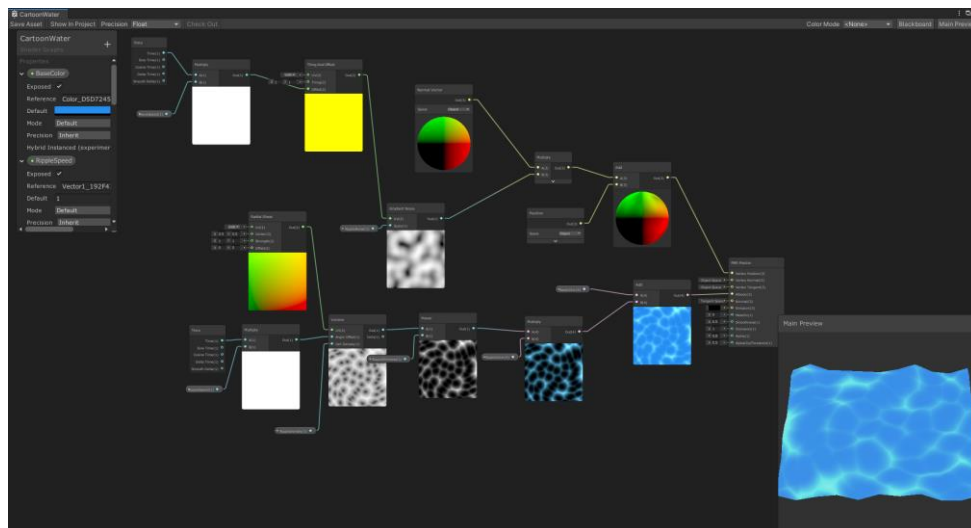
La primera tarea del proyecto consiste en la creación de un plano que simule las texturas de una piscina. Para ello se estudian diferentes tecnologías capaces de obtener dicho resultado, y se acaba haciendo uso del canal de renderizado universal a través de un tutorial del canal de YouTube de Brackeys [17].

---

[17] (Brackeys, GameDev Tutorials Brackeys, 2020)

El primer paso del tutorial consiste en implementar el canal de renderizado universal de Unity [18] a través de una de las páginas de su documentación. Este canal de renderizado proporciona a los desarrolladores flujos de trabajo sencillos de usar para implementar de forma rápida gráficos optimizados para cualquier plataforma, desde móviles hasta consolas y PCs. Para implementar este canal de renderizado basta con inicializar un nuevo proyecto y seleccionar la plantilla que incluye esta característica.

Una vez se implementa el canal de renderizado, el siguiente paso es crear un shader para poder asignárselo al material que se usará en un objeto para mostrar el resultado final por pantalla. El shader que se crea es de tipo PBR, es decir, con un renderizado basado en físicas para simular el flujo de luz como en el mundo real. Y para generarlo basta con crear un flujo de nodos para definir las diferentes características que tiene el shader. Algunas de estas propiedades son el color, el ruido que genera las ondas del agua y los multiplicadores que hacen que el ruido se modifique a partir del tiempo. La Figura 10 muestra el flujo de nodos del shader de agua.



*Figura 10. Flujo de nodos del shader de agua*

Con el shader generado el paso final del tutorial consiste en crear un plano al cual se le asigna el material que contiene el shader, de esa forma implementando exitosamente la primera tarea del proyecto.

### *6.1.2. Modelaje del escenario*

Uno de los aspectos más importantes de un juego es el escenario, ya que transporta al jugador al lugar en el espacio donde sucede toda la acción que el juego puede brindar. En el caso de un juego de waterpolo este lugar es la piscina, que debe contener todos los elementos necesarios para hacer creer al jugador que se trata de un lugar de verdad, y no solamente una simulación.

---

[18] (Unity, Universal Render Pipeline overview, 2021)



Para modelar los distintos assets del juego se hace uso de Maya, un programa informático ideal para desarrollar gráficos y animaciones en 3D.

### Piscina

El primer paso para modelar un objeto es conocerlo de antemano, y por esa misma razón se decide modelar la piscina de 33m del Centro Natación Helios, donde se han jugado partidos de waterpolo en Zaragoza desde hace décadas, y que forma parte de la historia del deporte en la ciudad. Sin embargo, debido a reformas, la piscina no presenta actualmente las cualidades que la hicieron tal especial en su momento, por lo que se hace uso de fotografías antiguas para reconstruirla. Las Figuras 11 y 12 muestran los cambios realizados en la piscina a lo largo de los años, entre los cuales se encuentra la demolición de las gradas.



*Figura 11. Piscina de 33 décadas atrás*

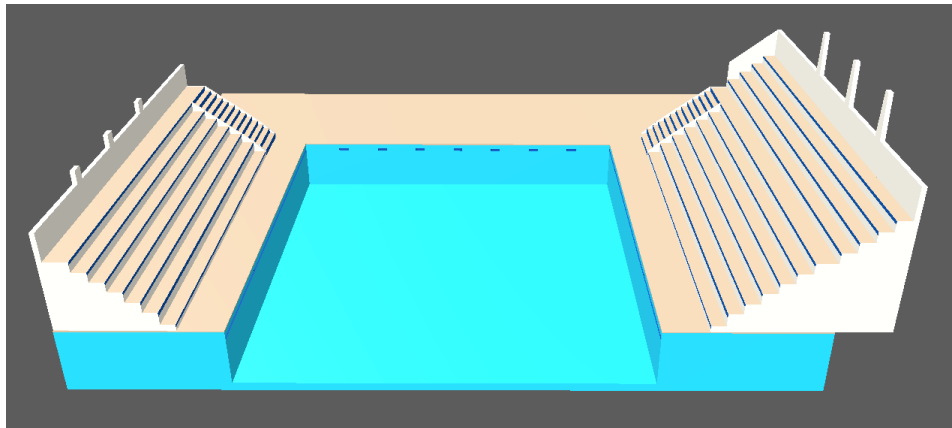


*Figura 12. Piscina de 33 actualmente*





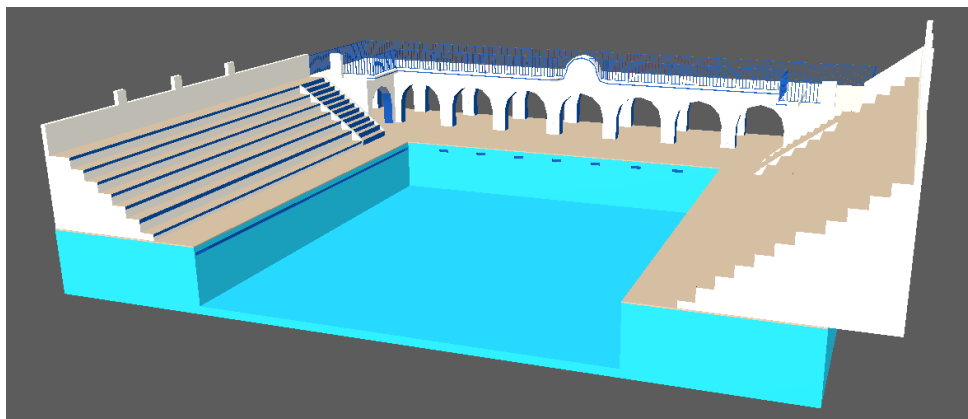
Para capturar la imagen de las gradas se analizan multitud de fotografías y se habla con socios del club que estaban presentes cuando todavía se jugaban partidos, construyendo así la piscina del juego de la manera más fiel posible. Usando Maya se decide realizar un modelaje low-poly, o con pocos polígonos, para hacer así que la calidad del escenario no choque con el shader que se utiliza para el agua, y tras unas horas analizando archivos y modelando caras se obtiene el escenario deseado para el juego. Además de ello, se crea una pequeña textura atlas para poder darle color homogéneo a toda la escena. La Figura 13 muestra el modelado de la piscina dentro de la aplicación de Maya.



*Figura 13. Objeto de la piscina en Maya*

### Arcos

El siguiente objeto que se debe modelar para el escenario son los arcos que se encuentran a la entrada de la piscina. Estos arcos fueron construidos a los pocos años de acabar la piscina, y forman parte de la identidad de la misma. Sin embargo, de forma similar a las gradas, los arcos sufrieron una serie de reformas cuando el resto de los elementos antiguos fueron demolidos. Por lo que para modelarlos se analizan fotografías antiguas para combinarlas con la apariencia actual y obtener una versión de los arcos que destaque de la misma forma que las gradas. La Figura 14 presenta el modelo de los arcos junto a la piscina.



*Figura 14. Modelo de la piscina con arcos en Maya*





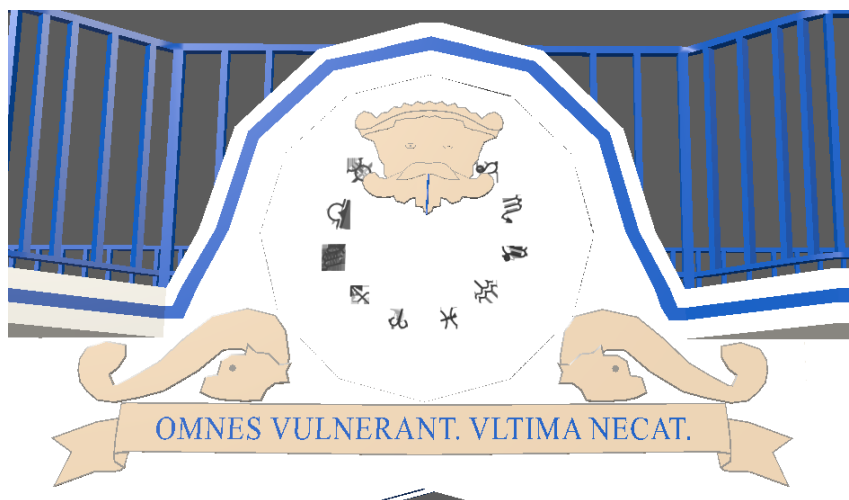
### Reloj de Sol

El último objeto que se requiere modelar para representar la piscina de 33 del Centro Natación Helios es su elemento más característico, el reloj de sol. Este reloj fue construido a la vez que los arcos y destaca sobre el resto de los elementos por ser el que más alto se encuentra. El reloj representa al dios del Sol y cuenta con una escritura que dice "Omnes Vulnerant. Ultima Necat", que en latín se traduciría como: "Todas hieren, la última mata", una referencia a las horas del reloj. La Figura 15 muestra el reloj de sol en la actualidad.



*Figura 15. Reloj de sol en la actualidad*

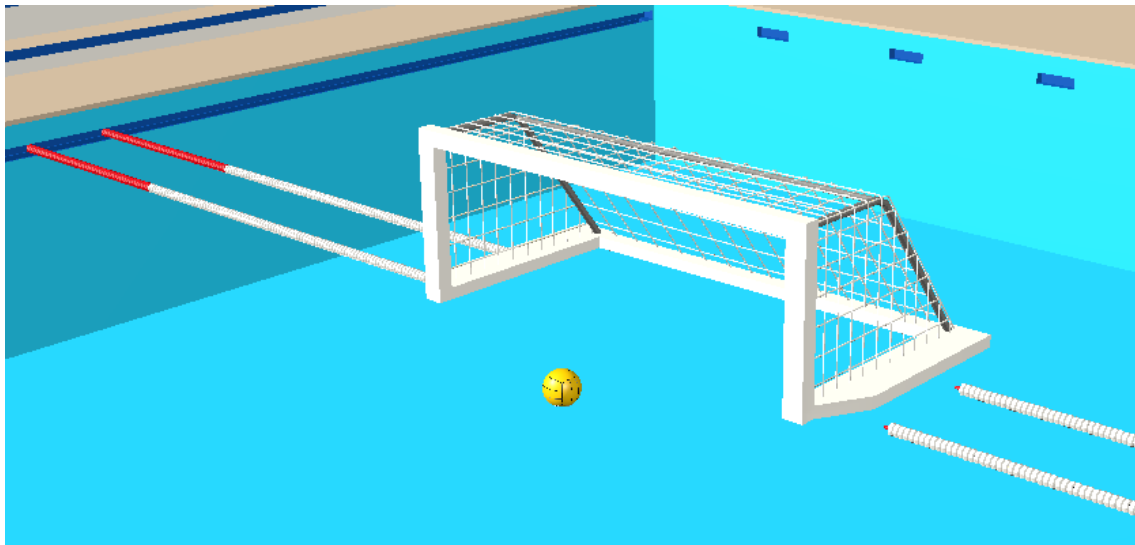
El modelado del reloj se hace tanto con elementos 3D como 2D. Para la inscripción y los elementos zodiacales del reloj se utilizan texto proporcionado por Maya y una textura, respectivamente. Mientras que la cara del reloj, los peces y la placa son modelados en 3D. La Figura 16 muestra el resultado de modelarlo en Maya.



*Figura 16. Modelo del reloj en Maya*

### Balón, portería y corcheras

Con los modelos del escenario acabados, el último paso para finalizar la fase de modelaje es crear los objetos relacionados con el deporte del waterpolo. El balón se modela a partir de una esfera a la cual se le añaden divisiones, estas divisiones se rotan posteriormente y se texturizan, obteniendo de esa manera un sencillo balón amarillo. Para la portería se toman referencias reales de porterías de waterpolo, y considerando el ancho de la piscina se obtienen las medidas necesarias para modelar el objeto. Las redes de la portería forman parte del objeto, por lo que no se moverán cuando un balón choque con ellas. Supongamos que están muy tensas y no pueden moverse. Finalmente, las corcheras se modelan a partir de un torus, un objeto primitivo con forma de rosquilla. Los torus se repiten hasta conseguir el ancho deseado y luego son texturizados dependiendo de su posición para representar diferentes partes del campo. La Figura 17 representa los modelos del balón, portería y corcheras en Maya.



*Figura 17. Modelos del balón, portería y corcheras en Maya*

Tras importar los elementos en la escena de Unity, se aprecia que el escenario parece vacío. Por lo que se propone buscar una skybox gratuita que se pueda utilizar para darle algo de vida al juego, incluyendo un nuevo objetivo en la iteración.

#### *6.1.3. Implementación de una skybox*

Una skybox [19] es una envoltura que se usa alrededor de la escena para mostrar cómo se vería el mundo más allá de su geometría. Las skybox son realmente importantes para poder observar el fondo de la escena, y es mucho mejor que tener un color gris detrás de los elementos del escenario.

---

[19] (Unity, Skybox, 2016)



Unity ofrece una tienda de assets para buscar elementos que los desarrolladores quieran introducir en sus juegos. A través de esta tienda se encuentra el asset Fantasy Skybox Free [20], una serie de skyboxes para representar diferentes partes del día y estaciones. Este asset resulta ideal para la escena del juego y se importa dentro del proyecto.

Con la skybox importada se puede ver la representación de la escena tras realizar las implementaciones gráficas del primer sprint, como se muestra en la Figura 18.



*Figura 18. Implementación del escenario inicial*

#### *6.1.4. Flotación del balón*

La última tarea del sprint consiste en implementar una pequeña funcionalidad al balón para que se pueda interactuar con él a través del móvil. La funcionalidad consiste en que el balón flote sobre la superficie del agua y cuando el jugador pulse la pantalla, este salga disparado hacia arriba.

Para saber más sobre la flotabilidad del balón se consulta una página web [21] que explica con detenimiento las diferentes fuerzas que toman parte en el balance. Con el estudio de las fuerzas se crea un algoritmo que calcula las diferentes fuerzas que afectan al balón cuando está en contacto con la superficie del agua, simulando de esa manera la flotabilidad del mismo.

Finalmente, se crea un pequeño script para que cuando el jugador presione la pantalla se cree una pequeña fuerza hacia arriba haciendo que el balón se eleve, intentando simular cuando un jugador coge una pelota y se levanta en el agua para lanzar.

Tras implementar la flotación, se construye el proyecto y se crea la aplicación, finalizando con la funcionalidad requerida para el primer sprint y cumpliendo con los objetivos propuestos a tiempo.

---

[20] (Knight, 2020)

[21] (Khan Academy, 2020)

## **6.2. Segunda iteración – Prototype (v0.1)**

Tras obtener un buen resultado en la implementación de la primera versión del proyecto, se decide que el concepto del juego es suficiente y que se puede continuar con el desarrollo del juego. Considerando la idea propuesta para el juego, se decide entre el tutor y el alumno invertir los esfuerzos de la segunda iteración en completar la mecánica principal del juego, es decir, el lanzamiento del balón.

Además de analizar requerimientos, se repasa también la gráfica burn-up para comprobar el progreso hasta el momento, que cumple con el objetivo de mantener una desviación positiva del desarrollo.

Para concluir la reunión, se deciden los diferentes subobjetivos que componen la mecánica principal:

- Lanzamiento del balón: 20 *storypoints*
- Físicas del balón en movimiento: 15 *storypoints*
- Rotación del balón: 1 *storypoints*

### *6.2.1. Lanzamiento del balón*

El lanzamiento del balón supone es sin lugar a dudas la mecánica más importante del juego, ya que el resto de funcionalidad depende de ella: puntuación, movimiento de los personajes... En definitiva, es el core del juego.

El lanzamiento del balón tiene tres pasos que hacen que la pelota salga disparada de cierta manera. El primero es el trazo del lanzamiento, que se guarda en un array de puntos y se simplifica gracias a la propiedad de tolerancia del line renderer. Una vez se ha simplificado comienza el segundo paso, donde se calculan las características del lanzamiento dependiendo de los ángulos verticales y horizontales que se obtienen del trazo. Finalmente las características se aplican como físicas aceleradas al balón, que hace que se mueva en la dirección deseada.

Este funcionamiento es el diseño inicial del algoritmo. Sin embargo, tras testear la mecánica la única clase de lanzamiento que se consigue generar es botando contra el agua, lo cual resulta muy repetitivo para ser la mecánica principal de un juego. Es por eso por lo que se propone que en la segunda fase se analice la clase de lanzamiento que el jugador desea realizar, y de esa forma conseguir tres tipos de lanzamientos distintos.

Estos tres lanzamientos son:

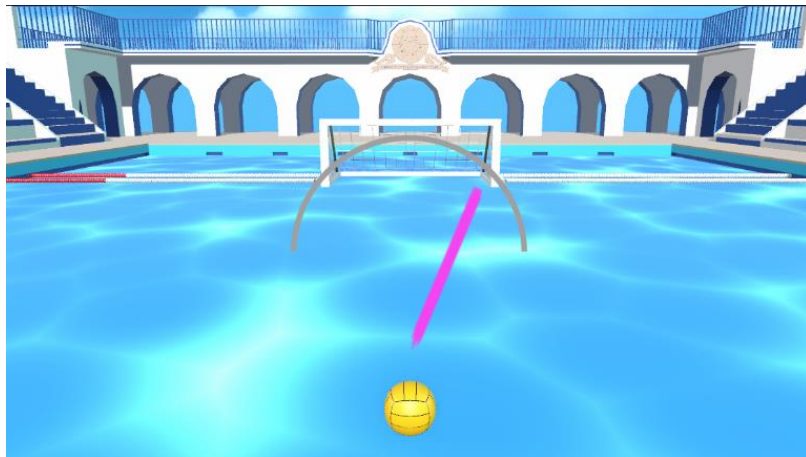
- Lanzamiento con bote
- Lanzamiento recto
- Vaselina

Los diseños con los que se identifican los lanzamientos y sus características son las siguientes:



### Lanzamiento a bote

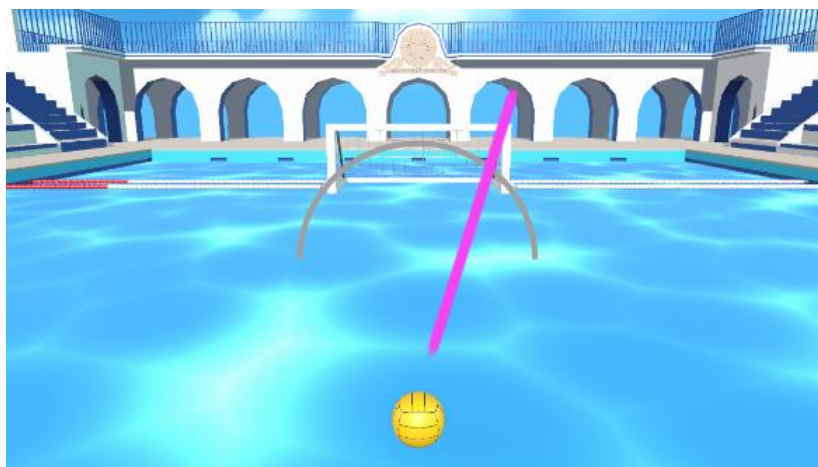
El lanzamiento a bote se genera cuando el trazo del lanzamiento es corto, y sus características son un lanzamiento con velocidad negativa en el eje Y, y velocidad positiva en el plano X-Z. La Figura 19 muestra el trazo del lanzamiento a bote.



*Figura 19. Trazo del lanzamiento a bote*

### Lanzamiento recto

El lanzamiento recto se genera cuando el trazo del lanzamiento es largo, y sus características son un lanzamiento con velocidad positiva en el eje Y, y en el plano X-Z. La Figura 20 muestra el trazo del lanzamiento recto.

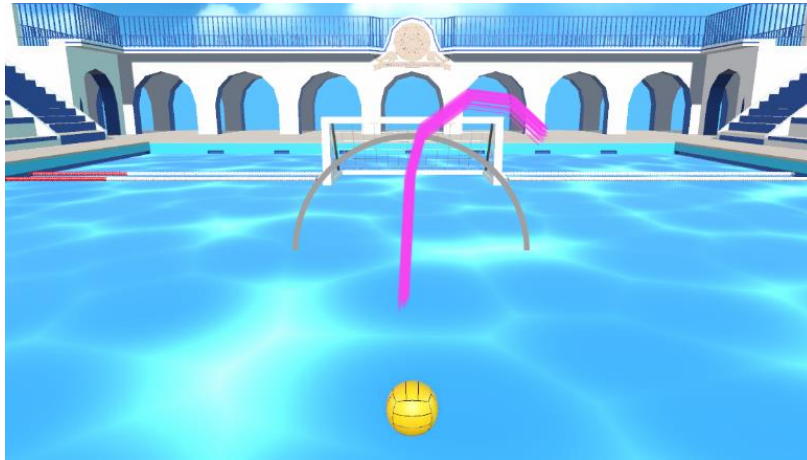


*Figura 20. Trazo del lanzamiento recto*

### Vaselina

La vaselina se genera cuando el trazo del lanzamiento es largo, curvo y el último punto del trazo no es el más alto. Sus características son un lanzamiento con velocidad positiva en todos los ejes, siendo la del eje Z la más pequeña de todas. La Figura 21 muestra el trazo de la vaselina.





*Figura 21. Trazo de la vaselina*

Estos tres tipos de lanzamientos dan por concluida la tarea de realizar un lanzamiento del balón. Sin embargo, este requerimiento se debe de apoyar en el que se desarrolla a continuación, implementar las físicas del balón en movimiento.

#### *6.2.2. Físicas del balón en movimiento*

Las físicas del balón hacen referencia al comportamiento del balón cuando sale disparado cuando el jugador realiza un trazo. Las características que se van a estudiar en este punto son:

- Velocidad del balón
- Bote del balón
- Colisión con el escenario

##### Velocidad del balón

La velocidad del balón viene dada en todo momento gracias al movimiento rectilíneo uniformemente acelerado que se produce en el plano X-Z. El MRUA se simula gracias a unas variables de velocidad y aceleración, siendo la última una constante. En el momento que el balón sale disparado lo hace a la máxima velocidad calculada, que disminuye en cada fotograma del juego. La velocidad del eje Y viene dada gracias al algoritmo de flotación que se comenta anteriormente. Cuando el balón no está en contacto con el agua no tiene una fuerza que lo haga ir hacia arriba, por lo que la única fuerza que se le aplica en ese momento es la de la gravedad, que hace que vaya hacia abajo. Por supuesto, si el lanzamiento presenta una velocidad positiva inicial en el eje Y, el balón irá hacia arriba, pero cambiará su curso con el tiempo hacia abajo tal y como lo hacen los cuerpos con masa en el mundo real.

##### Bote del balón

El bote del balón es una ocurrencia muy común en un partido de waterpolo, por lo que es imprescindible simular de la mejor manera este comportamiento. El bote suele ocurrir cuando el



balón colisiona contra la superficie del agua y tiene la suficiente energía para seguir en movimiento tras chocar y perder parte de esa energía en el agua. Para simular este comportamiento en el juego se hace uso del ángulo de colisión y la velocidad horizontal del balón. Si la velocidad es superior a un valor constante y el ángulo de colisión se encuentra entre un intervalo también fijo, el sistema cambiará el sentido de la velocidad en el eje Y tras añadirle una pequeña fuerza extra, haciendo así que el bote sea algo más exagerado que en el mundo real. Además, el balón perderá velocidad en el plano horizontal, simulando así la pérdida de energía en el choque. Por otra parte, si la velocidad es menor a la necesaria o el ángulo de colisión no es el adecuado, el balón chocará con el agua y se quedará en el sitio, simulando, como se dice en términos del deporte, “que se quede clavada”.

#### Colisión con el escenario

La implementación final de las físicas del balón se da cuando este colisiona con diferentes objetos del escenario. Estas colisiones se distinguen entre duras y blandas. Las colisiones duras son con objetos menos deformables que el balón, como los palos, el bordillo o incluso las gradas y arcos. Cuando el balón colisiona con estos objetos cambia la dirección horizontal a la deseada y reduce ligeramente su velocidad tras el choque. Por otra parte, cuando el balón colisiona con la red, que se considera blanda, pierde más energía que con el choque duro, simulando de esa manera la colisión real con el objeto.

Es preciso comentar también que al realizar las mecánicas de choque a mano el programa no para a los objetos de meterse unos dentro de otros. Este problema ocurre principalmente cuando el balón bota encima del bordillo de la piscina, haciendo que cuando deja de tener velocidad atraviese el bordillo. Para solucionar este problema se realiza una comprobación cuando la pelota se encuentra encima del bordillo, haciendo que deje de tener velocidad en el eje Y, haciendo de esa manera que se quede flotando justo encima y simule que el balón está apoyado encima del bordillo.

#### *6.2.3. Rotación del balón*

La última tarea de la segunda iteración consiste en simular la rotación del balón en movimiento. Para ello se analiza la rotación del balón durante un entrenamiento de waterpolo y se concluye que el balón rota siempre entorno a los ejes X e Y, dependiendo de la dirección en la que se lance. Es por eso por lo que se utilizan los valores de la velocidad horizontal para que de una manera similar a la del MRUA, se pueda simular esta rotación. Para implementar este comportamiento basta con pasar usar los valores de velocidad y usarlos para calcular la velocidad de rotación a la que se desea que rote el balón. Con el valor calculado se decelera hasta llegar al valor deseado hasta que este llega a cero.

Implementando este algoritmo de rotación se consigue que cuando el balón pare pueda seguir rotando como lo haría en la vida real, completando así todos los objetivos de la segunda iteración.



### 6.3. Tercera iteración – Alpha (v1.1)

La tercera reunión de análisis del proyecto repasa la implementación de la mecánica de lanzamiento en el sprint anterior, revisando también el cumplimiento de los *storypoints* propuestos. Habiendo terminado con la parte del juego controlada por el jugador, el siguiente punto de desarrollo se centra en los personajes no jugables y su comportamiento. Para esta iteración se decide desarrollar la parte gráfica de los personajes usando assets ya existentes o creándolos manualmente. Además, ya que se debe hacer uso de herramientas de modelado se propone también añadir más objetos a la escena para que no se sienta tan vacía.

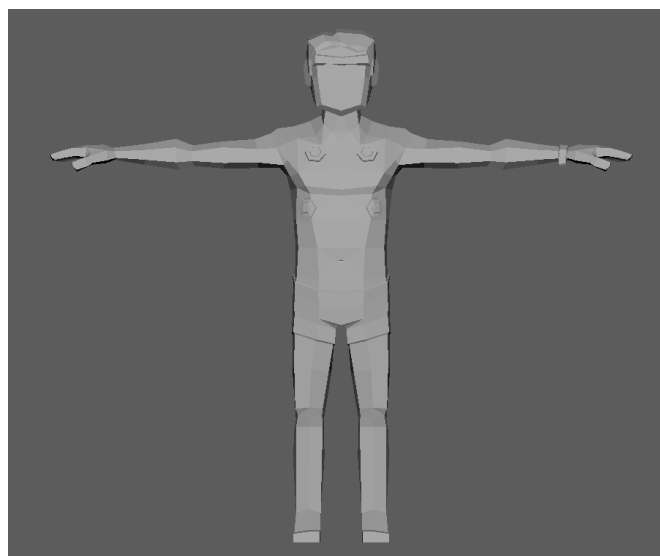
Con los requerimientos acordados se proponen los siguientes objetivos para el sprint:

- Añadir assets al fondo de la escena: 6 *storypoints*
- Obtener assets de personajes: 16 *storypoints*
- Crear animaciones de movimiento: 16 *storypoints*

#### 6.3.1. Obtener assets de personajes

La Universidad San Jorge posee una librería de assets que obtiene a partir de terceros para uso de los alumnos. Es por eso por lo que se hace uso de esos paquetes de assets para implementar los diseños de los personajes. Tras una búsqueda de objetos que se pudieran usar en el juego se encuentran los paquetes desarrollados por Synty Studios, que presentan escenas en las que aparecen naves espaciales con personajes y bosques llenos de diferentes tipos de árboles y flores.

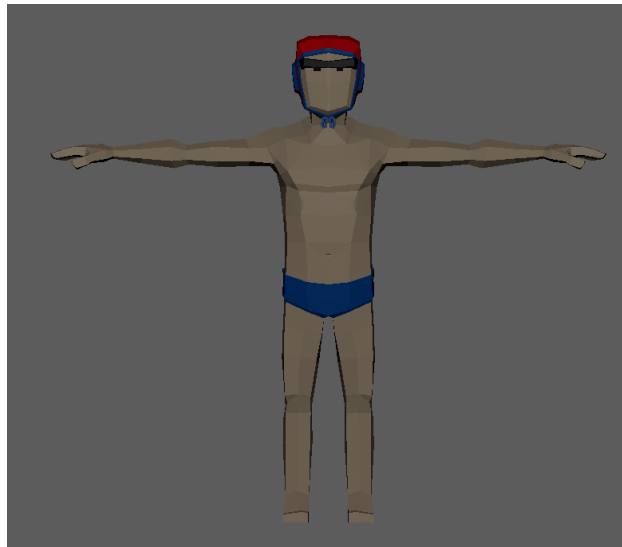
La escena del espacio contiene diferentes tipos de personajes, y tras analizarlos todos se encuentra uno que podría servir para el juego. Este personaje aparece semidesnudo y con una serie de objetos en el pecho. Dado el conocimiento del alumno en herramientas de modelaje se decide modificar este personaje para así obtener el modelo de los jugadores de waterpolo. La Figura 22 muestra el modelo original del personaje.



*Figura 22. Modelo original del personaje*



Para los personajes de waterpolo se requiere que lleven un gorro y bañador. Es por eso por lo que se modifica el modelo original eliminando la cadena de la muñeca izquierda y los objetos del pecho, modificando la ropa interior para asemejarla a un bañador y, finalmente, modificando la geometría de la cabeza para crear un gorro de waterpolo. Para concluir con el modelo del personaje se texturiza con el atlas ya creado para obtener el modelo final tal y como se muestra en la Figura 23. Para el juego se requieren dos tipos de NPCs, el portero con gorro rojo y los defensas con gorro azul, por lo que se crea otra versión del personaje de la Figura 23 con un color diferente.



*Figura 23. Modelo final del personaje*

#### *6.3.2. Añadir assets al fondo de la escena*

Tal y como se hace en el desarrollo de modelos de la primera iteración, se utiliza la piscina de 33m del Centro Natación Helios para obtener la idea de objetos a modelar. En este caso se requieren objetos que aparezcan fuera de la piscina, y que no destaquen como el resto, por lo que se decide modelar bancos, salvavidas y árboles.

El modelo de los bancos y salvavidas se basa en los objetos de la piscina, mientras que la colocación de los modelos de árboles se trata de hacer de tal manera que cubran homogéneamente el fondo de los arcos. Una vez incluidos los objetos en la escena se modelan también unas rayas en el fondo de la piscina para darle un poco más de detalle a la escena. Finalmente, se decide incluir también un asset de nubes del mismo paquete para tener algo de geometría acompañando a la skybox.

Para concluir con la tarea se crea un pequeño script que hace girar al objeto de nubes simulando así su movimiento natural gracias al viento. En la Figura 24 se puede observar la implementación de los nuevos objetos en la escena de Unity.



*Figura 24. Escena de la piscina en Unity*

### 6.3.3. Crear animaciones de movimiento

El último objetivo de la tercera iteración consiste en crear las animaciones del portero y los defensas. Para crear las animaciones se busca información sobre los diferentes procesos que se deben realizar para obtener el resultado deseado, y gracias al foro de Unity [22] se averigua que el primer paso es realizar el rigging del modelo.

El rigging es el proceso de desarrollo en el que se generan huesos digitales a partir de un modelo para poder implementar una animación en base al esqueleto. Para realizar este proceso se sigue un tutorial en YouTube creado por la cuenta oficial de Maya [23], lo que hace muy sencillo seguir las instrucciones.

Para crear el esqueleto del modelo del personaje se hace uso de la herramienta 'Quick Rig' que ofrece Maya, con la cual genera automáticamente un esqueleto para el personaje. Sin embargo, los puntos de articulación del modelo no son todo lo precisos que deberían, por lo que se elimina el esqueleto generado para crear uno más convincente. Este rigging se realiza otra vez con 'Quick Rig' salvo que esta vez se elige la opción de montar el esqueleto paso a paso, eligiendo en todo momento la posición dentro de la geometría del objeto donde se deben colocar las articulaciones. Tras ajustar todos los puntos necesarios se confirma la elección para crear el esqueleto final del personaje, que se comporta tal y como lo haría un esqueleto de verdad, obteniendo así el resultado deseado.

El siguiente paso del proceso es crear las animaciones en Maya, y para ello se sigue otro tutorial en YouTube [24] que enseña las bases para crear animaciones de forma rápida y sencilla.

---

[22] (Unity, 2021)

[23] (Maya Learning Channel, 2016)

[24] (Yawyee, 2021)

Antes de comenzar el tutorial se crea una lista con las diferentes animaciones que se desean elaborar, distinguiendo entre animaciones del portero y animaciones del jugador:

- Portero:
  - o Tensión: dos manos en el agua y patada alterna.
  - o Salto al medio: patada simultánea y movimiento hacia arriba de las manos.
  - o Salto escuadra: patada simultánea inclinando el cuerpo hacia un lado y extendiendo hacia arriba una de las manos. Una animación para cada lado.
  - o Salto medio: similar al salto de escuadra pero extendiendo una de las manos a una altura media del cuerpo. Una animación para cada lado.
  - o Salto bajo: similar al salto de escuadra pero extendiendo una de las manos hacia abajo. Una animación para cada lado.
- Jugador:
  - o Tensión: dos manos en el agua y patada alterna.
  - o Bloqueo: patada alterna con una mano fuera del agua mientras la otra realiza tensión. Una animación para cada mano.
  - o Salto: elevación de la mano alta y salto de piernas hacia arriba. Una animación para cada mano.

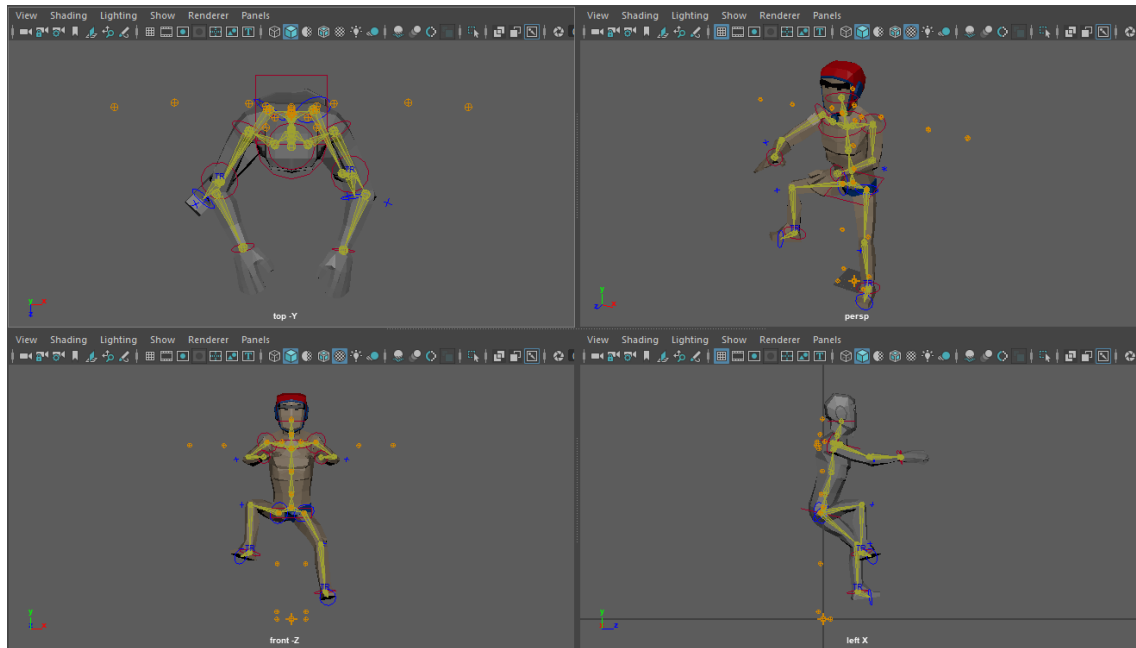
Con la lista de animaciones completas se procede a iniciar el tutorial, que divide la animación en cinco pasos. El primero es la elección de referencias para las animaciones, y así poder copiar los movimientos naturales de una persona en el modelo. Para ello se realizan una serie de videos en la piscina simulando los movimientos requeridos, y obteniendo así las referencias deseadas. El segundo paso consiste en crear las poses claves a lo largo de la cadena de fotogramas manipulando la posición y rotación de las articulaciones. En el tercer paso se deben crear las poses secundarias entre las claves, para obtener de esa manera una interpolación más suave. Tras crear las poses, en el cuarto paso se modifica la posición de la cadera en cada una de las poses establecidas para lograr movimientos más naturales a lo largo de la animación. Finalmente, para concluir con el proceso se modifican en el editor gráfico las curvas de interpolación obteniendo así las animaciones necesarias para los personajes del juego.

Con las animaciones creadas el último paso de la tarea es importarlas dentro del motor de Unity. Para ello se consulta otro tutorial en YouTube [25] que explica la configuración que se debe realizar para exportar exitosamente animaciones desde Maya. Tras seguir el tutorial se importan todas las animaciones en Unity y se crean los controladores necesarios para animar los personajes en la escena. Con los controladores asignados se crea una máquina de estados para cada tipo de personaje especificando las condiciones de transición entre estados. Después de probar las

---

[25] (Blackthornprod, 2018)

animaciones se observa que son muy lentas para el juego. En Maya se realizan a lo largo de 60 fotogramas a una velocidad de 30 fotogramas por segundo, haciendo que cada animación dure 2 segundos en completarse. Para solucionar el problema se modifica la velocidad de las animaciones en Unity hasta lograr el resultado deseado, completando así la tarea y consecuentemente los objetivos de la tercera iteración. En la Figura 25 se observa la creación de una pose clave dentro de Maya.



*Figura 25. Proceso de creación de una pose en Maya*

#### **6.4. Cuarta iteración – Alpha (v1.2)**

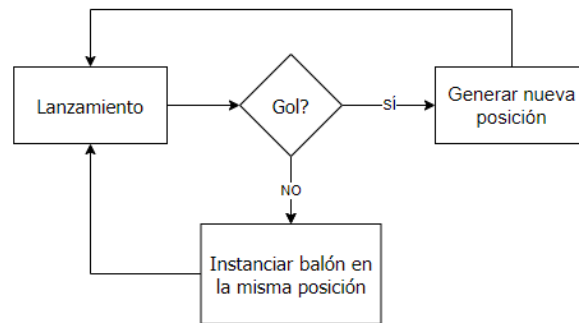
Tras la implementación de todos los assets necesarios para el desarrollo del juego, la cuarta iteración comienza con todos los *storypoints* propuestos hasta la fecha completados. El siguiente paso del desarrollo es la implementación del algoritmo de reposicionamiento, que hace que se pueda lanzar el balón desde cualquier lado de la piscina, y no solo desde el medio. También se propone diseñar la interfaz de usuario de la pantalla de juego, para poder visualizar la implementación completa de toda la mecánica principal.

Los subobjetivos de la iteración junto a sus respectivos *storypoints* son:

- Creación e implementación del algoritmo de lanzamiento y reposición: 20 *storypoints*
- Diseño de la interfaz de usuario de la pantalla de juego: 4 *storypoints*

##### *6.4.1. Creación e implementación del algoritmo de lanzamiento y reposición*

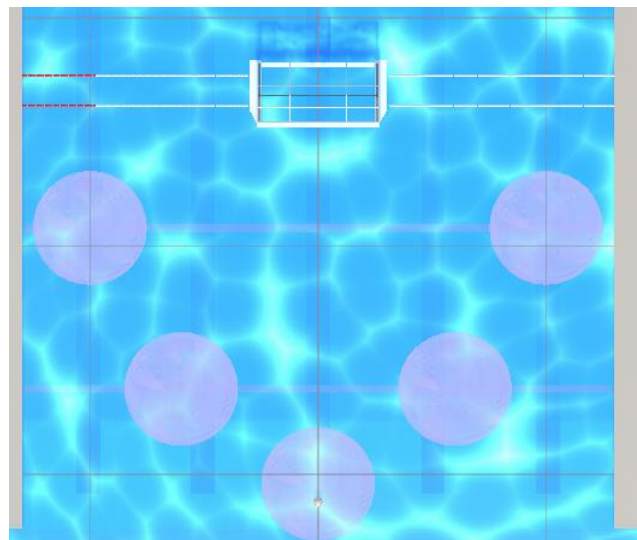
El primer paso en la implementación del algoritmo es diseñar su funcionamiento a través de un diagrama de flujo, para de esa manera entender su comportamiento y lo que se espera conseguir. La Figura 26 muestra el diagrama de flujo del algoritmo de reposición.



*Figura 26. Diagrama de flujo del algoritmo de reposición*

El primer paso del algoritmo es detectar si un lanzamiento ha sido gol. Para ello se crea un colisionador dentro de la portería que detecta si el balón ha chocado con él tras el lanzamiento. Si se detecta la colisión se considera que el lanzamiento es gol y se debe generar una nueva posición pseudoaleatoria.

Antes de generar la posición es preciso entender dónde puede posicionarse el balón. En waterpolo existen 5 posiciones de juego que forman una media luna alrededor de la portería, y son las 5 posiciones que se eligen para posicionar el balón previo a un lanzamiento. La Figura 27 muestra las diferentes posiciones de lanzamiento dentro del juego.

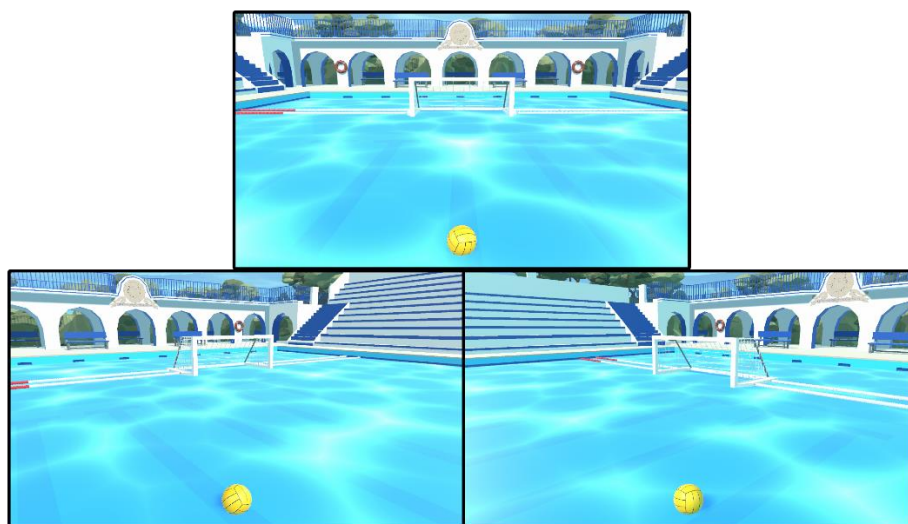


*Figura 27. Posiciones de lanzamiento del balón*

Continuando con la explicación del algoritmo, cuando se elige generar una nueva posición no se debe depender totalmente en la generación de un número aleatorio, por lo que se guarda en todo momento las posiciones de los dos últimos lanzamientos para no repetirlos cuando se elige desde donde lanzar. Una vez se elige la posición de lanzamiento se generan unas coordenadas aleatorias para que el balón pueda posicionarse de diferente forma dentro del círculo de cada posición, generando de esa manera una posición totalmente nueva para cada lanzamiento.

Sin embargo, tras la implementación del algoritmo aparecen unos problemas que no se habían considerado anteriormente relacionados a la cámara y al lanzamiento del balón. Al crear la mecánica de lanzamiento desde una posición centrada a la portería, no se consideran los ángulos con respecto a esta, por los que es necesario realizar cambios en la posición y rotación de la cámara.

Cada vez que se genera un nuevo lanzamiento se debe posicionar y orientar la cámara con respecto a un offset relativo al balón, para que siempre pueda apuntar en dirección a la portería, que es lo que se desea en todo momento. Estos cambios hacen que independientemente de la posición de lanzamiento, la sensación y visión que tiene el jugador sea siempre la misma. La Figura 28 muestra las diferentes vistas del juego dependiendo de la posición.



*Figura 28. Vistas del balón dependiendo de la posición*

De la misma forma, también ocurre un problema relacionado con los ángulos cuando se realiza el lanzamiento, haciendo el balón siempre vaya en dirección a los arcos y no a la portería. Para solucionar el problema se modifica el algoritmo original de la siguiente manera.

Cuando se genera una nueva posición se orienta el balón en dirección a la portería, para que siempre mire hacia ella. Después, cuando el jugador ha dibujado el trazo del lanzamiento se calcula el ángulo y se suma a la rotación del balón, haciendo que este apunte en la dirección calculada. De esta forma el lanzamiento ya no depende de la posición y puede generarse sin problema desde cualquier rincón de la piscina. Finalmente, ya no se aplican diferentes velocidades a los ejes X y Z independientemente para crear un vector, sino que se aplica la velocidad directamente en la dirección frontal del balón, consiguiendo así la implementación deseada.

Para concluir con la tarea, tras haber realizado un testeo de los diferentes tipos de lanzamiento junto al resto de compañeros de equipo, se acaba quitando la vaselina, ya que resulta difícil de realizar y no tan satisfactoria e intuitiva como el resto de los lanzamientos.

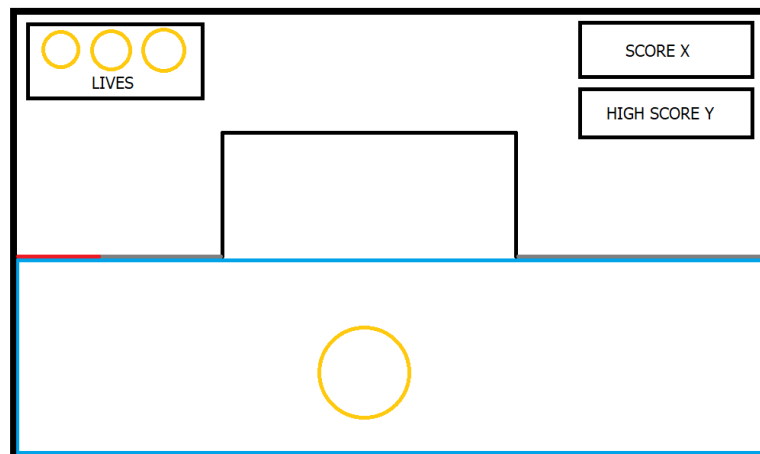




#### 6.4.2. *Diseño de la interfaz de usuario de la pantalla de juego*

Con la implementación del sistema de reposición los jugadores pueden realizar lanzamientos desde cualquier parte del campo indefinidamente. Sin embargo, no existe forma alguna de realizar un seguimiento del número de goles ni de cuantos intentos le quedan al jugador. En una de las reuniones de análisis del desarrollo se llegó a la conclusión de que el juego debía de tener un sistema por vidas en el cual estas decrementaran cada vez que se fallaba un lanzamiento, y que incrementaran cuando se metiera gol, por lo que se acabó acordando que hubiera un máximo de 3 vidas. Además, también se propuso tener un contador de la mayor puntuación jamás lograda al lado de la puntuación actual, por lo que los elementos necesarios para elaborar la interfaz están decididos.

Tras algunos diseños de la interfaz se acaba eligiendo el que se muestra en la Figura 29, junto a la implementación en el juego en la Figura 30.



*Figura 29. Diseño de la IU de la pantalla de juego*



*Figura 30. Implementación de la IU de la pantalla de juego*

En la interfaz de usuario del juego se decide situar los elementos fuera del medio de la pantalla para no impedir de esa manera el trazado del balón, que es lo más importante del juego. En el caso de las vidas, se decide utilizar capturas del modelo 3D para dar una sensación de

homogeneidad en la pantalla. Finalmente se decide añadir un poco de transparencia a los paneles en los que se encuentran los elementos gráficos para no ocultar del todo el escenario del juego y que no parezca que la pantalla este dividida en elementos totalmente distintos.

Tras acabar con la implementación de la interfaz de usuario se incluye un pequeño efecto en el que al perder una vida esta se oscurece hasta quedar totalmente negra, y cuando se marca un gol la vida vuelve a recobrar todo el color. Este efecto se consigue aplicando una corrutina que decrementa o incrementa el valor RGB de la imagen usada para las vidas. Creando así un efecto visual que ayuda al usuario a sentirse más involucrado con el número de intentos restantes que le quedan.

Finalmente, para acabar el progreso de la cuarta iteración se crea un pequeño menú que aparece cuando el jugador se queda sin intentos. Este menú informa al jugador sobre su puntuación y el récord de goles que tiene, preguntándole si desea volver a jugar o terminar la partida. El botón de salir de la partida todavía no está implementado, ya que no se han diseñado los menús. Sin embargo, el botón de volver a jugar resetea la pantalla de juego dándole la oportunidad al jugador de empezar de cero.

## **6.5. Quinta iteración – Beta (v2.1)**

La reunión de la quinta iteración repasa el progreso del desarrollo del juego en base a la planificación inicial que se realiza, observando que todos los subobjetivos propuestos se han cumplido a tiempo. Después de implementar las animaciones del juego y la reposición de los lanzamientos se propone realizar la implementación de los algoritmos de posicionamiento de defensores y de movimiento de todos los NPCs. Concluyendo así la reunión con la siguiente serie de objetivos para la iteración:

- Creación e implementación del algoritmo de aparición de defensores: 6 *storypoints*
- Creación e implementación del algoritmo de movimiento del portero: 6 *storypoints*
- Creación e implementación del algoritmo de movimiento de los defensores: 6 *storypoints*

### *6.5.1. Creación e implementación del algoritmo de movimiento del portero*

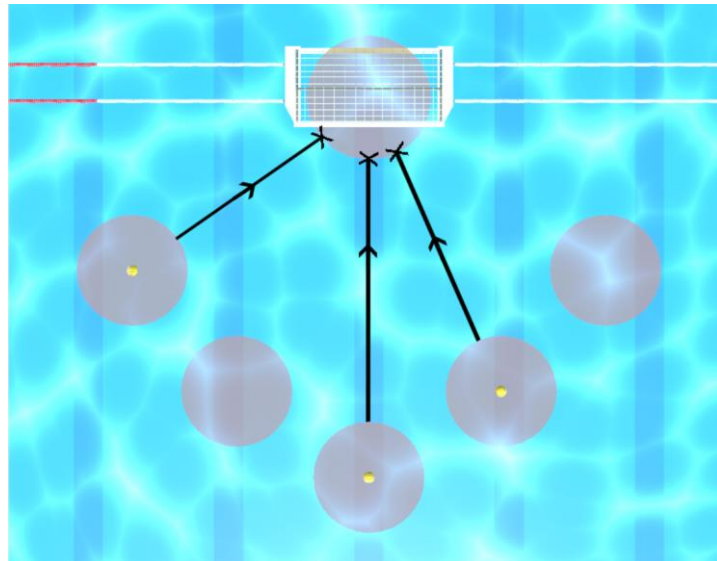
La misión de un portero es principalmente la de parar balones, sea como sea para evitar así que haya gol. Para ello, la herramienta principal que poseen los porteros en el agua es el salto. Saltar en la dirección del balón y parar el lanzamiento. Por lo que el algoritmo debe hacer lo mismo que en la vida real, analizar la trayectoria del balón y saltar en su dirección para pararlo.

Sin embargo, los porteros suelen parar más balones cuanto mejor posicionados están, por lo que el primer paso es colocar al portero en la posición ideal dentro del agua. Para lograr este comportamiento se crea una medialuna dentro de la portería que representa las posiciones óptimas que puede tomar el portero. Además conociendo la posición inicial del balón antes de





cada lanzamiento se puede colocar al portero de la mejor forma posible. La Figura 31 muestra el proceso de colocación del portero dependiendo del lanzamiento.

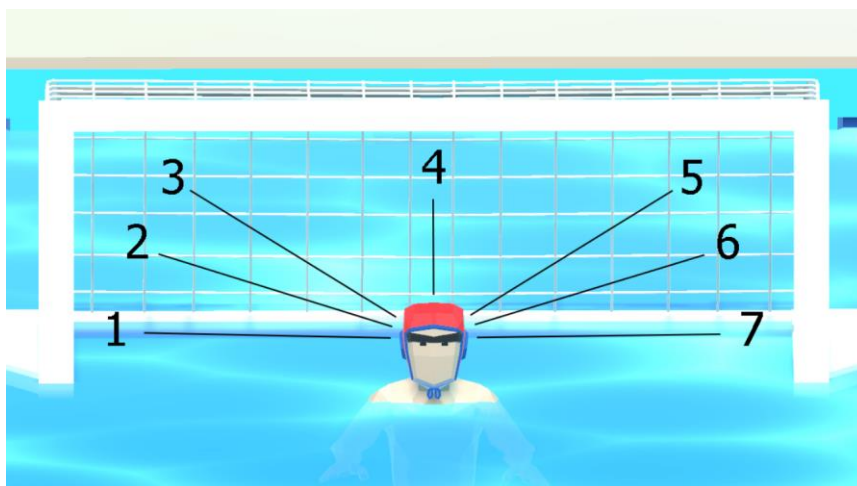


*Figura 31. Posicionamiento del portero dependiendo de la posición del balón*

El algoritmo de posicionamiento funciona dibujando un rayo desde la posición del balón hacia la medialuna del portero. Una vez se produce la colisión se obtiene el punto de choque y se coloca al portero en dicha posición. Finalmente se orienta al portero mirando hacia el balón, logrando así una posición natural y eficaz.

Tras posicionar al portero el siguiente paso es hacer que salte en la dirección adecuada. Para ello se utiliza la posición y altura del balón con respecto al portero dentro de un rango determinado. Una vez calculado el tipo de salto se aplican las físicas necesarias y las animaciones requeridas para simular la actuación del portero.

Existen 7 tipos de salto que el portero puede realizar dependiendo de la posición y altura del balón. Estos saltos coinciden con los de las animaciones y se pueden observar en la Figura 32.



*Figura 32. Posibles saltos del portero*

Con el objetivo de obtener una progresión constante de dificultad dentro del juego, se crea un sistema de niveles de dificultad que afectaran a los NPCs a medida que el jugador vaya marcando goles. Los atributos del portero que se modifican en cada nivel son su velocidad de movimiento, el radio de detección del balón y la distancia que recorren en cada salto. Los niveles de la inteligencia artificial aumentan una vez cada 5 goles hasta llegar al nivel 5. La Tabla 1 muestra el valor de las propiedades del portero dependiendo del nivel de dificultad.

Nivel	Velocidad	Radio Detección	Distancia de salto
1	1	6	0.2
2	1.5	6.5	0.4
3	2	7	0.6
4	2.5	7.5	0.8
5	2.75	8	1

*Tabla 1. Propiedades del portero dependiendo del nivel de dificultad*

La implementación del algoritmo permite a la entidad del portero posicionarse y saltar de la misma manera que lo haría en la vida real. Logrando de esa manera el comportamiento que se espera del personaje no jugable.

#### *6.5.2. Creación e implementación del algoritmo de aparición de defensores*

La labor de los jugadores en defensa es igual o mayor a la del portero, ya que dificultan la tarea de los lanzadores si hacen bien su trabajo, pudiendo incluso llegar a hacer que el balón no llegue a portería. Como el juego es un simulador de faltas y el movimiento de los defensas hacia el jugador está penalizado en el deporte, los jugadores tendrán la tarea de bloquear balones, y si el balón pasa cerca de ellos moverse para facilitar el bloqueo.

De la misma forma que el portero, los defensas disponen de radios de detección, velocidad de movimiento y una distancia de salto. Pero lo que de verdad hace que realicen bien su labor es la colocación de cada uno de ellos, que se explica a continuación. Para concluir con la explicación, se indica el número de defensores que hay en cada uno de los niveles del juego. En el nivel 1 no hay ninguno, en los niveles 2 y 3 hay un defensor, y finalmente en los niveles 4 y 5 hay 2 defensores. En los niveles donde solo hay 1 defensor éste puede cubrir ambos palos. Es decir, una jugada el primer palo y otra el segundo, como lo decida un número generado aleatoriamente.

El algoritmo de posicionamiento de los defensas funciona de la siguiente manera:

1. Obtener el vector que une cada palo con la pelota

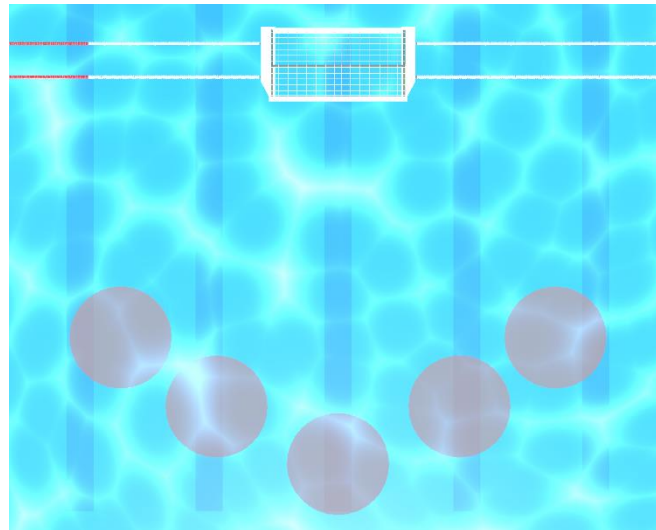


2. A lo largo de la línea del balón con el palo aplicar una distancia aleatoria
3. Mover la posición en el eje horizontal dependiendo de la puntuación del jugador

En el paso 2, para que el defensa no aparezca ni pegado al balón ni dentro de la portería, se genera el número aleatorio entre dos puntos determinados. Siendo uno de ellos una línea imaginaria a 2 metros de la portería, y el otro un punto a distancia mínima del balón.

Finalmente, en el paso 3 se genera un número aleatorio entre dos valores que van decrementando su valor a medida que la puntuación del jugador aumenta. Haciendo que en los lanzamientos con una puntuación baja el intervalo sea mucho más grande que en los lanzamientos con puntuaciones altas.

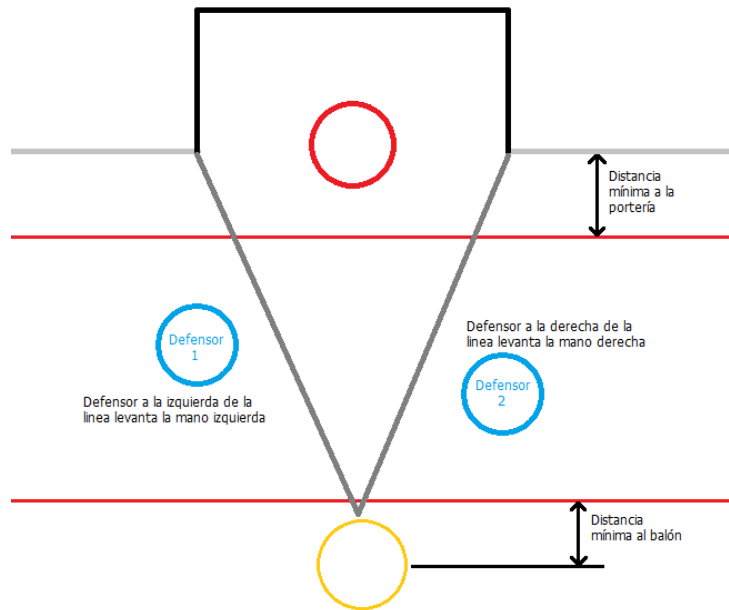
Tras generar el algoritmo de posicionamiento ocurren algunos problemas que hacen que la aplicación se congele. Esto se debe a que los defensores deben mantener una distancia mínima entre ellos mismos, la portería y el balón, lo que hace que puedan llegar a generarse bucles infinitos debido a la falta de espacio. Para solucionar este problema se modifican las posiciones de lanzamiento para dejar un mayor espacio entre el balón y los defensas, consecuentemente dándole al jugador un mayor ángulo para realizar los lanzamientos. En la Figura 33 se muestran las posiciones finales de lanzamiento.



*Figura 33. Posiciones finales de lanzamiento*

#### **6.5.3. Creación e implementación del algoritmo de movimiento de defensores**

Con la implementación del posicionamiento de defensas terminada, el siguiente paso en la iteración es el movimiento de los defensas respecto al balón. Para ello, aprovechando que los defensas ya están colocados, se decide con un simple algoritmo que brazo deben de levantar para tapar así el lanzamiento a portería. La Figura 34 muestra el esquema del algoritmo de levantamiento de brazo.



*Figura 34. Esquema de los defensores*

El esquema explica el objetivo de los defensas, que es tapar la línea entre el balón y el palo levantando el brazo más próximo a dicha línea. Para implementar este comportamiento se hace uso del desplazamiento horizontal que se aplica a los defensores cuando se calcula su posición, levantando un brazo dependiendo del valor del desplazamiento. En caso de que el jugador obtenga puntuaciones elevadas y los defensas se coloquen en línea del palo, se levanta para el primer palo (derecha) el brazo derecho, mientras que se levanta el izquierdo para el segundo palo (izquierda).

Con el brazo del defensa implementado el siguiente paso es implementar su reacción al balón. Para ello se hace uso de un algoritmo similar al del portero, reaccionando a los balones que entran dentro de un radio de detección y saltando hacia arriba para tapar más espacio. Las propiedades del defensor dependiendo de su nivel son tal y como se indican en la Tabla 2.

Nivel	Velocidad	Radio Detección	Distancia de salto
1	1	3	0.2
2	1.5	3.5	0.4
3	2	4	0.6
4	2.5	4.5	0.8
5	2.75	5	1

*Tabla 2. Propiedades del defensa dependiendo del nivel de dificultad*



Finalmente, una vez implementado el comportamiento de todos los NPCs se añade una variable para controlar la velocidad de animación de los personajes. Haciendo que se muevan más o menos lentos y den una sensación de personalidad que distinga la intensidad a la que trabajan. Con esta implementación se concluyen los objetivos de la quinta iteración, que completa las mecánicas del juego base. En la Figura 35 se puede observar el posicionamiento y reacción de los NPCs a un lanzamiento del jugador.



*Figura 35. Reacción de los NPCs a un lanzamiento*

## **6.6. Sexta iteración – Master Candidate**

Tras finalizar la quinta iteración se realiza otra sesión de testeo con los miembros del equipo de waterpolo del Centro Natación Helios, y el principal inconveniente es la temperatura que alcanza el teléfono móvil tras una larga sesión de juego. Para investigar sobre la razón de este calentamiento se hace uso de la herramienta Profiler, que permite visualizar diferentes aspectos del rendimiento del juego. Sin embargo, a pesar del uso de la herramienta no se obtiene una conclusión clara de que es lo que hace que funcione de esa manera. Tras un análisis de problemas similares en los foros de Unity se propone limitar el número de fotogramas por segundo en la aplicación a 30, que hasta el momento obtiene 60 de manera continua.

La reunión de desarrollo estudia el problema anteriormente mencionado y la necesidad de corregirlo durante la sexta iteración. Finalmente, se proponen las últimas tareas del desarrollo, que están centradas en la implementación de menús, efectos de sonido y un tutorial para introducir al jugador a las mecánicas del juego.

Los objetivos de la sexta iteración junto a sus respectivos *storypoints* son:

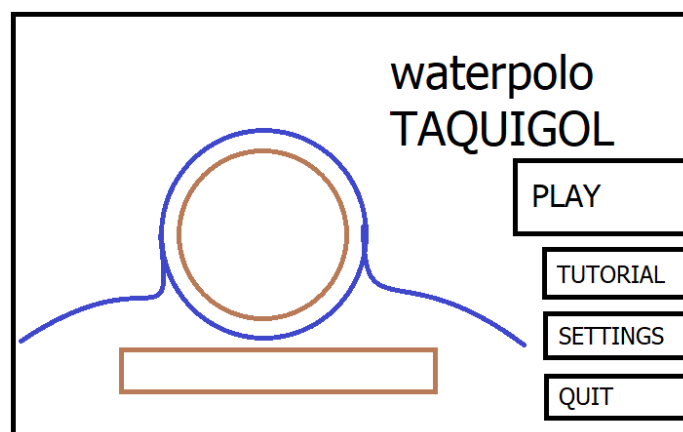
- Diseño e implementación de los menús: 4 *storypoints*
- Búsqueda y creación de efectos de sonido: 4 *storypoints*
- Edición e implementación de efectos de sonido: 4 *storypoints*



- Creación del tutorial: 8 *storypoints*

#### 6.6.1. Diseño e implementación de los menús

Tras el diseño de la interfaz que se muestra en la pantalla de juego se decide seguir con una estética similar para el resto de las pantallas del juego. La primera que se debe diseñar es el menú principal, el cual contiene diferentes botones que permiten al jugador moverse por las pantallas del juego. Para esta pantalla se decide usar una cámara que apunte al reloj de sol de la piscina, ya que a pesar de su importancia no se llega a apreciar del todo durante el progreso del juego. En cuanto a los botones se decide seguir usando los paneles negros con cierta transparencia para que la presentación visual sea homogénea a lo largo de todo el juego. La Figura 36 presenta el boceto del menú principal.



*Figura 36. Diseño del menú principal*

Tras implementar el boceto del menú dentro del juego se aprecia que la fuente usada para los menús no tiene la personalidad necesaria para mostrar de una forma estética el texto del juego. Es por eso por lo que se busca en librerías web open-source [26] diferentes fuentes para utilizar en el juego, y se acaba eligiendo Noto Sans JP, la cual encaja perfectamente con el juego. Para el título del juego se hace exactamente lo mismo. Sin embargo, con Unity es imposible crear el trazo necesario para que el título se pueda leer correctamente en la pantalla, por lo que se prueba a implementar un objeto en Maya con el fin de exportarlo posteriormente a Unity.

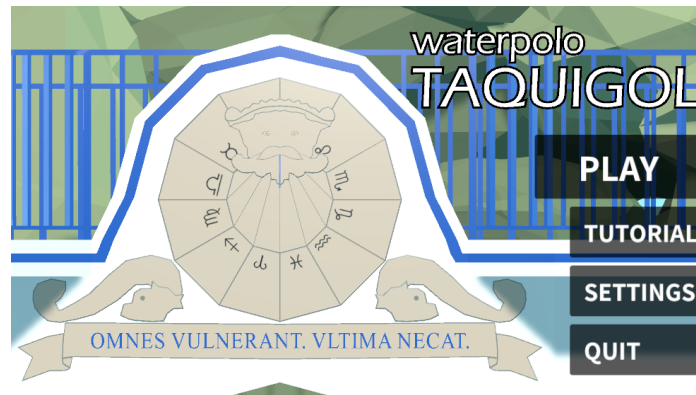
Una vez en Maya se crea un elemento de tipo texto y se configura su geometría para crear un trazo externo a partir del texto. Con la geometría generada basta con cambiar la fuente a una que se pueda aplicar al proyecto, como ITC Eras. Para finalizar con Maya se crean unas texturas blancas y negras para asignárselas al objeto y poder conseguir el diseño deseado, tras el cual se importa como un objeto en Unity.

---

[26] (Google, 2021)



Finalmente, una vez en Unity se añade el objeto del título a la escena y se obtiene el resultado final, tal y como muestra la Figura 37.



*Figura 37. Implementación final del menú principal*

La siguiente pantalla que se debe diseñar es la de los ajustes, donde el usuario puede cambiar la calidad de los gráficos, limitar el número de fotogramas por segundo y ajustar el volumen de los efectos sonoros. Para el diseño de la pantalla se decide usar una cámara que apunte a la piscina desde arriba, permitiendo al jugador observar el escenario desde otra perspectiva. En cuanto a los elementos del menú, se decide ordenarlos en una columna encima de un panel negro que permita ver el escenario del fondo sin problemas.

Para ajustar el volumen se utilizan deslizadores que usan como manilla una pelota de waterpolo. Mientras que para los elementos que cambian los gráficos y número de fotogramas se utilizan desplegables con las siguientes opciones disponibles:

- Gráficos:
  - Bajo
  - Medio
  - Alto
- Fotogramas por segundo:
  - 24
  - 30
  - 60

Para implementar los ajustes se sigue otro tutorial de Brackeys [27] donde explica cómo se mapea de una interfaz a las opciones del juego en tiempo real. Los gráficos se cambian a partir de los ajustes de calidad del motor, mientras que para el volumen se utilizan mezcladores de sonido, que permiten ajustar el volumen del juego en cualquier momento. En cuanto a la limitación de

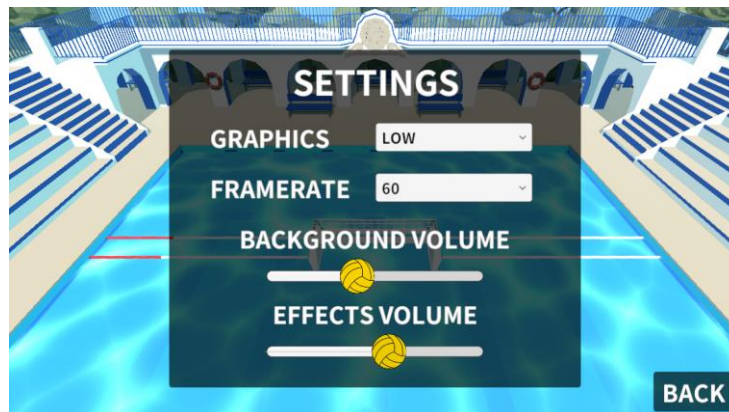
---

[27] (Brackeys, SETTINGS MENU in Unity, 2017)



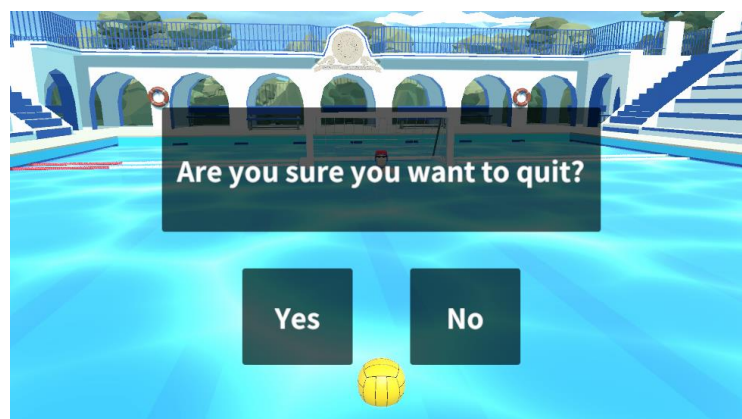
fotogramas, se crea una función de código que permite ajustar los fotogramas por segundo de la aplicación una vez el juego está en funcionamiento.

Finalmente, para guardar los ajustes del jugador, se crea un script capaz de guardar en un fichero la información de dichos ajustes. Este fichero se actualiza cuando el jugador sale del menú de ajustes y se lee cada vez que se abre la aplicación, aplicando en ese momento los ajustes deseados. La Figura 38 muestra la implementación final del menú de ajustes.



*Figura 38. Implementación del menú de ajustes*

Tras implementar la pantalla principal y la de ajustes, el siguiente paso es modificar la interfaz de usuario de la pantalla de juego para añadir un botón que permita salir al menú principal en cualquier momento. Este botón abre una nueva interfaz que pregunta al usuario si desea salir del juego, y en ese caso lo devuelve al menú principal. La implementación de la pantalla de salida del juego se muestra en la Figura 39.



*Figura 39. Implementación de la pantalla de salida del juego*

Ajustar la calidad de video y limitar el número de fotogramas por segundo resulta en un mejor rendimiento de la aplicación, evitando de esa manera el calentamiento del dispositivo móvil.



---

#### *6.6.2. Búsqueda y creación de efectos de sonido*

Los efectos de sonido y música juegan un rol importante en la experiencia del usuario en videojuegos modernos. Algunos efectos ayudan a los diseñadores a crear tensión, añadir emoción y crear una inmersión en el mundo que toma partida el juego. Este proyecto no es diferente, y en esta tarea se buscan los efectos de sonido necesarios para darle vida a la piscina del juego.

La Universidad San Jorge presenta un convenio con Epidemic Sound [28], una plataforma que contiene una inmensa librería de música y efectos sonoros que los estudiantes pueden utilizar en sus proyectos, y que se utiliza en el desarrollo del proyecto.

Sin embargo, antes de analizar la obtención de efectos de sonido es preciso hacer primero una lista con los diferentes sonidos que se necesitan para el juego:

- Pulsar un botón con efecto positivo
- Pulsar un botón con efecto negativo
- Sonido de fondo para la piscina
- Colisión con los palos
- Colisión con la pared
- Parada
- Pitido de gol
- Pitido de final de partido
- Bote del balón

Con Epidemic Sound se consiguen obtener los efectos de sonido necesarios para los botones y el sonido de fondo de la piscina. Sin embargo, no existe ningún efecto en la librería que pueda simular el resto, por lo que se decide grabar estos efectos manualmente en la piscina del club. Para ello se prepara una sesión de Foley para obtener todos los efectos directamente de su origen.

Los sonidos necesarios para simular la colisión con los palos se realizan pegando con un balón a una portería pequeña dentro de una piscina pequeña, mientras que el sonido de la pared se obtiene tras lanzar el balón contra una pared de cemento. El bote del balón se graba similarmente a los anteriores, cogiendo un balón pequeño y tratando de botarlo en la piscina pequeña del club desde fuera del agua, logrando así el distintivo sonido que surge cuando se bota un balón. Los pitidos se graban directamente de la fuente de origen, tratando de realizar un sonido constante y no con muchos picos de intensidad. Finalmente, para el sonido de la parada se utiliza un muñeco de un bebé para golpear un balón, simulando exitosamente el efecto que se produce cuando un balón choca con el cuerpo humano.

---

[28] (Epidemic Sound, 2021)

---

#### *6.6.3. Edición e implementación de efectos de sonido*

Tras la obtención de los efectos de sonido necesarios para el juego se deben editar las pistas de audio para obtener así el resultado deseado. En caso de los efectos de fondo, se unen usando la herramienta de Reaper [29] para obtener así una pista de audio lo suficientemente larga para que no se corte durante el juego.

El resto de los efectos son editados usando MP3 Cutter [30], una herramienta online que permite recortar sonidos y editar sus características como quiera el usuario. Usando esta herramienta se modifican los efectos para obtener un sonido propio de un escenario abierto, como el de una piscina al aire libre. Finalmente, se exportan dos o tres variaciones de cada efecto para poder usarlos de manera aleatoria cuando se necesite.

El siguiente paso del proceso es importar todas las piezas de audio a Unity y diferenciar entre las que deben tener un efecto 3D y las que no. El audio 3D es una característica que ofrece Unity que permite modificar las características de una fuente de sonido en tiempo real, de esa manera simulando una espacialización del sonido que la cámara puede capturar. Este efecto 3D se usa en los efectos que dependen del balón, como las colisiones con diferentes cuerpos. El resto de los sonidos tienen un efecto plano o 2D, ya que el no hace falta apreciar su origen. Todos estos efectos se gestionan con un mezclador de sonido para efectos, mientras que el ruido de fondo de la piscina usa otro mezclador totalmente distinto, distinguiendo de esta manera en el menú de opciones entre volumen de efectos y volumen de fondo.

#### *6.6.4. Creación del tutorial*

Para poder finalizar con la implementación de características dentro del juego, es precisa la creación del tutorial donde el jugador pueda aprender a jugar. Como se muestra en el diseño de la pantalla del menú principal, existe un botón para abrir la pantalla del tutorial. Tras un análisis de requerimientos con el tutor se decide que la mejor manera de implementar el tutorial es a través de pequeños videos. Estos videos muestran los gestos que deben realizar los jugadores y los efectos que se aplican al balón, así como una explicación sobre el sistema de vidas.

Para implementar este comportamiento es preciso aprender a introducir videos dentro del motor de Unity, por lo que se estudian diferentes tutoriales hasta dar con el indicado. Este tutorial [31] ayuda a crear una imagen que tiene como textura la salida de video de otro objeto.

Conociendo la forma de implementar los videos se crean una serie de animaciones con Unity para simular los gestos del usuario, las cuales usan imágenes de manos open-source. Con las animaciones generadas se editan los scripts de lanzamientos para obtener las características

---

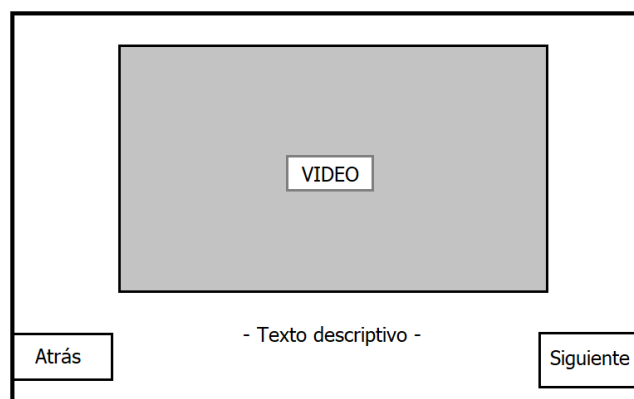
[29] (Reaper, 2021)

[30] (123APPS, 2021)

[31] (Vegas, 2020)

necesarias para cada animación. Finalmente, se graban con la capturadora de video de Windows para poder ser exportadas a Blender.

Una vez en Blender se recortan los videos para obtener solo la parte necesaria para el tutorial, tras lo cual se elige una resolución y calidad de video suficiente para poder exportarlo al proyecto. El último paso de la tarea es crear la pantalla con el texto necesario para mostrar en el tutorial. Para el diseño se elige mostrar el video en medio de la pantalla con un pequeño texto debajo, teniendo a los lados unos botones para avanzar o retroceder en el tutorial. En cuanto a la cámara se utiliza la misma que en el menú de opciones, mirando hacia el escenario desde el cielo. El diseño de la pantalla del tutorial se muestra en la Figura 40.

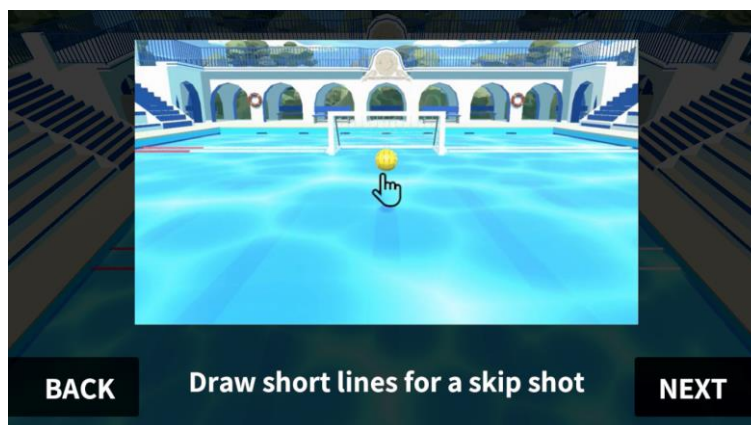


*Figura 40. Diseño de la pantalla del tutorial*

Finalmente se crea el script que maneja los distintos estados del tutorial, con sus consecuentes videos y textos descriptivos. Las diferentes fases que cubre el tutorial son:

- Cómo levantar el balón
- Cómo lanzar a bote
- Cómo lanzar recto
- Sistema de pérdida de vidas

La Figura 41 muestra la pantalla del tutorial en la fase del lanzamiento a bote.



*Figura 41. Fase de lanzamiento a bote del tutorial*

---

La compleción de la tarea del tutorial finaliza los objetivos especificados en la reunión de análisis, y consecuentemente, la fase de desarrollo del proyecto. De la misma forma, también se cumple con éxito el objetivo de mantener un seguimiento de desarrollo con una desviación por encima de la propuesta, un tema que se desarrollará en profundidad en el capítulo 7 de la memoria.

En este capítulo se estudia cómo se construye la demo del proyecto, incluyendo la adaptación de la metodología, el estudio de los objetivos y la superación de retos o problemas surgidos a lo largo del periodo de desarrollo.

## **7. Control y seguimiento del desarrollo**

Tal y como se establece en los objetivos del capítulo 4, uno de los objetivos del proyecto es el control y seguimiento del desarrollo para evitar una desviación negativa del mismo. Este capítulo estudia el progreso del desarrollo desde la perspectiva de producción, analizando los requerimientos de cada sprint, sus *storypoints*, y finalmente, la desviación con respecto a la planificación deseada.

El primer elemento que se debe analizar es la pila de producto del proyecto, la cual contiene todos los requerimientos que se han implementado a lo largo del desarrollo junto a sus respectivos *storypoints*. La pila de producto distribuida entre las 6 versiones de desarrollo es:

### Concept – 32 *storypoints*

- Idea del juego: 4 *storypoints*
- Shader del agua: 2 *storypoints*
- Paleta de color: 1 *storypoint*
- Modelaje del escenario: 20 *storypoints*
- Implementación de una skybox: 1 *storypoint*
- Flotación del balón: 4 *storypoints*

### Prototype 0.1 – 36 *storypoints*

- Lanzamiento del balón: 20 *storypoints*
- Físicas del balón en movimiento: 15 *storypoints*
- Rotación del balón: 1 *storypoint*

### Alpha 1.1 – 36 *storypoints*

- Obtener assets para el fondo de la escena: 4 *storypoints*
- Obtener assets de personajes: 4 *storypoints*
- Modificar los personajes: 12 *storypoints*
- Crear animaciones para los personajes: 16 *storypoints*

### Alpha 1.2 – 24 *storypoints*

- Crear e implementar algoritmo de lanzamiento y reposición: 20 *storypoints*
- Diseño de la interfaz de usuario: 4 *storypoints*

### Beta 2.1 – 20 *storypoints*

- Crear e implementar algoritmo de aparición de defensores: 6 *storypoints*
- Crear e implementar algoritmo de movimiento del portero: 6 *storypoints*
- Crear e implementar algoritmo de movimiento de los defensores: 6 *storypoints*
- Implementar las animaciones de los personajes: 2 *storypoints*

### Master Candidate – 34 storypoints

- Diseño e implementación de los menús: 4 *storypoints*
- Creación del logo y búsqueda de fuentes para el juego: 2 *storypoints*
- Creación del título del juego: 2 *storypoints*
- Creación de efectos de sonido: 2 *storypoints*
- Búsqueda de efectos de sonido: 2 *storypoints*
- Edición e implementación de efectos de sonido: 4 *storypoints*
- Creación de animaciones y videos para el tutorial: 8 *storypoints*
- Corrección de bugs: 10 *storypoints*

El gráfico burn-up al comienzo del desarrollo contiene los diferentes *storypoints* necesarios para completar las versiones del juego y las desviaciones, tal y como se muestra en la figura 42.

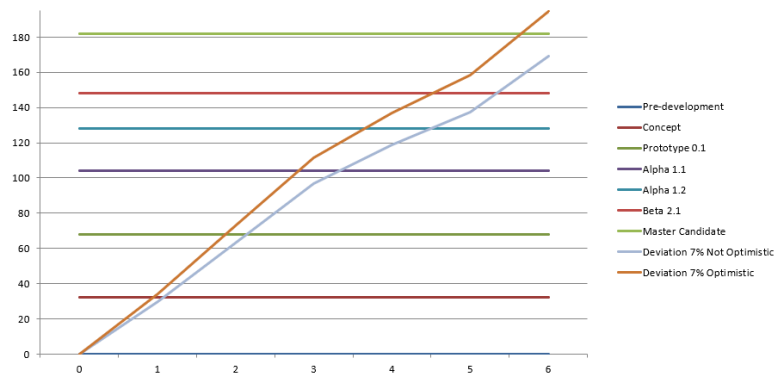


Figura 42. Gráfico burn-up previo al desarrollo

### **7.1. Concept**

El concepto supone la primera versión del proyecto, en la cual se implementan los requerimientos necesarios para obtener una imagen del potencial del juego, y así decidir si se sigue adelante con la idea o se propone una nueva. Al completar la versión, el gráfico presenta la siguiente forma, tal y como se muestra en la figura 43.

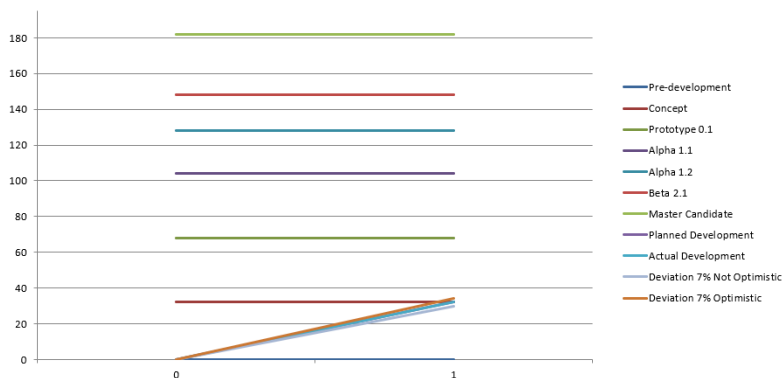


Figura 43. Gráfico burn-up tras la primera versión de desarrollo

Tal y como se muestra en la Figura 43, se puede observar que el progreso real del desarrollo cumple con el esperado, alcanzando el número de *storypoints* necesarios para completar la versión 1 del proyecto.

## 7.2. Prototype (v0.1)

El prototipo es la segunda versión de desarrollo del juego, donde se deben implementar las mecánicas de lanzamiento que se utilizan en el resto de las versiones. Tras la implementación los requerimientos necesarios el estado del gráfico es el siguiente, representado en la Figura 44.

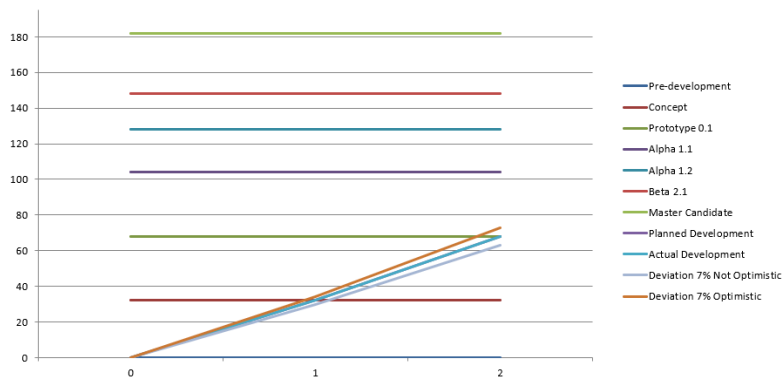


Figura 44. Gráfico burn-up tras la segunda versión de desarrollo

Como se puede apreciar en la Figura 44, el número de *storypoints* implementados llega al necesario para completar con el prototipo del juego, cumpliendo con el objetivo de la segunda iteración de desarrollo.

## 7.3. Alpha (v1.1)

La primera versión Alpha del juego incluye la creación de los personajes y animaciones necesarias, así como la compleción de la escena con elementos de la piscina. Tras la implementación de los requerimientos necesarios, el gráfico presenta la siguiente forma.

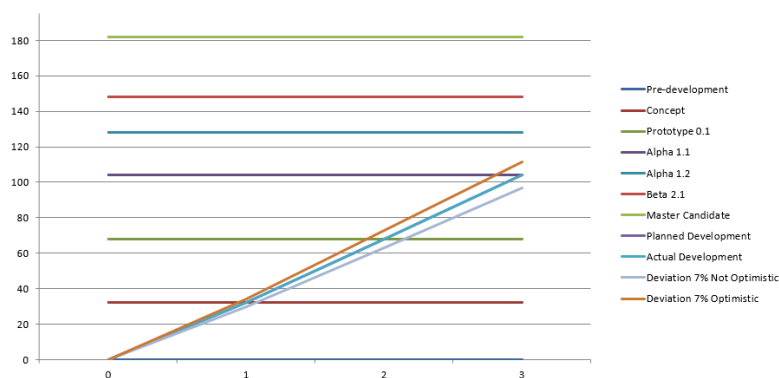


Figura 45. Gráfico burn-up tras la tercera versión de desarrollo

Tal y como se muestra en la Figura 45, el progreso real del desarrollo cumple con el número de *storypoints* necesarios para cerrar la tercera versión del juego.

#### 7.4. Alpha (v1.2)

La segunda versión Alpha del juego consiste en la implementación del algoritmo de reposicionamiento del balón, así como las modificaciones necesarias para realizar lanzamientos desde dichas posiciones. El gráfico burn-up presenta la siguiente forma tras la implementación de dichos requerimientos, como se muestra en la Figura 46.

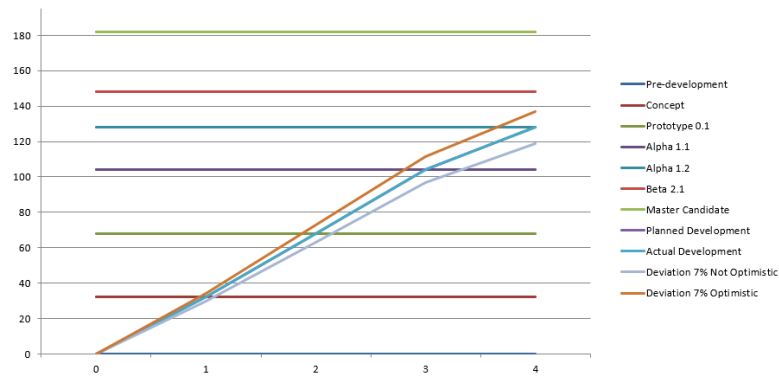


Figura 46. Gráfico burn-up tras la cuarta versión de desarrollo

En la Figura 46 se puede apreciar la compleción de los *storypoints* necesarios para concluir exitosamente con la cuarta versión del proyecto.

#### 7.5. Beta (v2.1)

La quinta versión del juego desarrolla los algoritmos necesarios para dar vida a los personajes no jugables del juego. La implementación de dichos requerimientos concluye con el desarrollo de los elementos que interactúan con el jugador. Representando dicho progreso a través de la Figura 47, que muestra el gráfico burn-up tras completar la quinta versión del juego.

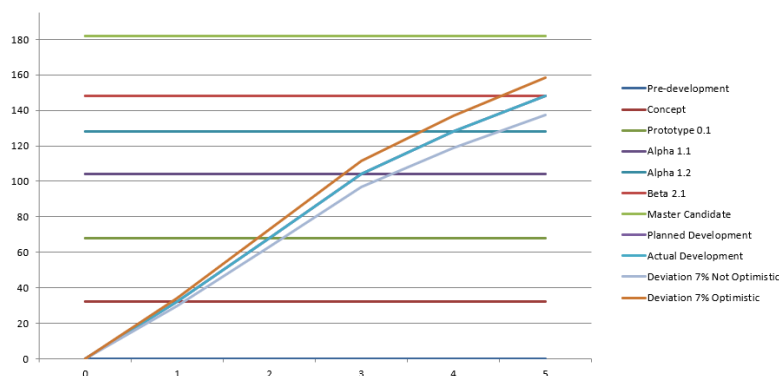


Figura 47. Gráfico burn-up tras la quinta versión de desarrollo

Una vez más se consigue completar todos los requerimientos propuestos para la quinta iteración, cumpliendo con el objetivo de control del progreso del proyecto.



## 7.6. Master Candidate

La sexta iteración de desarrollo supone la conclusión del proyecto, y en ella se crean los elementos del juego necesarios para ofrecer al jugador efectos de sonido, menús y tutoriales para aprender a jugar. Tras el desarrollo del sprint el gráfico burn up presenta la siguiente forma.

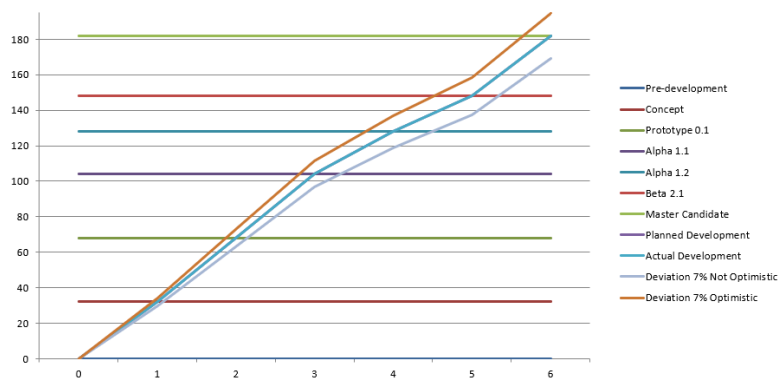


Figura 48. Gráfico burn-up final

La Figura 48 presenta la forma final del gráfico burn-up, en el cual se visualiza la compleción de todos los requerimientos propuestos por el equipo de trabajo. De la misma forma se completa el desarrollo del proyecto manteniendo el desarrollo propuesto, y de esa manera cumpliendo exitosamente con el objetivo de mantener dicho desarrollo por encima de la desviación pesimista. Finalmente, en la Figura 49 se muestra también la tabla usada para controlar el gráfico burn-up a lo largo del desarrollo de todo el proyecto.

	Pre-development	Concept	Prototype 0.1	Alpha 1.1	Alpha 1.2	Beta 2.1	Master Candidate
	02/02/21	02/02/21	02/26/21	03/24/21	04/28/21	05/14/21	05/26/21
<b>Pre-development</b>	0	0	0	0	0	0	0
<b>Concept</b>	32	32	32	32	32	32	32
<b>Prototype 0.1</b>	68	68	68	68	68	68	68
<b>Alpha 1.1</b>	104	104	104	104	104	104	104
<b>Alpha 1.2</b>	128	128	128	128	128	128	128
<b>Beta 2.1</b>	148	148	148	148	148	148	148
<b>Master Candidate</b>	182	182	182	182	182	182	182
<b>Planned Development</b>	0	32	68	104	128	148	182
<b>Actual Development</b>	0	32	68	104	128	148	182
Deviation 7% Not Optimistic	0	29,76	63,24	96,72	119,04	137,64	169,26
Deviation 7% Optimistic	0	34,24	72,76	111,28	136,96	158,36	194,74

Figura 49. Tabla de control del gráfico burn-up





## 8. Estudio Económico

El objetivo del estudio económico del proyecto es el cálculo de los distintos tipos de indicadores que informen sobre la viabilidad de este. El estudio económico [32], al contrario que un presupuesto, estima los costes de explotación. Es decir, los costes que se producen durante la vida útil del proyecto. Para el cálculo del estudio se efectuará el método económico, el cual evalúa la relación costes-ingresos.

### 8.1. Método económico

El método económico consiste en la estimación de los costes e ingresos de funcionamiento del proyecto para calcular los ratios de rentabilidad. La estimación de los costes se determina a partir de la naturaleza de estos, siendo los más importantes los costes anuales por naturaleza, los costes por secciones y finalmente, los costes por producto.

Los costes por naturaleza pueden clasificarse a su vez en costes de materiales, costes de mano de obra y costes por utilización de equipos e instalaciones, que a su vez dependen del coste de amortización, consumo y mantenimiento.

Por otra parte, el ingreso se obtiene a través del producto del precio de venta unitario y el volumen de ventas. El precio de venta se estima a partir del estudio de productos similares en el mercado, mientras que el volumen de ventas se trata como variable independiente debido a la dificultad que supone realizar su estimación.

Conociendo tanto costes como ingresos del proyecto se puede determinar el beneficio bruto de la siguiente manera:

$$\text{Beneficio bruto } (B) = \text{Ingresos } (I) - \text{Costes totales } (C)$$

$$\text{Tasa de rentabilidad } (r)$$

Otro concepto importante de entender y determinar es el punto de equilibrio, el cual indica el volumen de ventas necesario para compensar los costes totales del proyecto. Suponiendo que el precio de venta y el coste unitario del producto sea independiente del volumen de ventas.

$$\text{Punto de equilibrio } (qe) = \frac{\text{Costes fijos } (Cf)}{\text{Precio de venta unitario } (p) - \text{Coste variable unitario } (cv)}$$

El estudio a través del método económico resulta realmente interesante para poder estimar la relación ingresos-costes del juego. Sin embargo, la publicación y estudio de monetización del proyecto no se tiene en cuenta a lo largo del desarrollo del proyecto, sino que se trata de un trabajo a futuro. Por esta razón se procede a analizar solamente los costes del desarrollo del proyecto, que se distribuyen de la siguiente manera:

---

[32] (UPV, 2017)



### Costes de mano de obra

Los costes de mano de obra abarcan todos los costes que se generan a través de la inversión de esfuerzo en forma de tiempo por los integrantes del proyecto. Este esfuerzo se calcula a través del número de horas invertidas en el proyecto, ya sean reuniones de estudio, análisis de herramientas o desarrollo del producto final.

Para calcular el número de horas invertidas se hace uso del apartado de desarrollo, en el cual se referencia el número de reuniones organizadas, siendo 6 las reuniones a lo largo del mismo. Además de las reuniones también se invierte tiempo en forma de estudio y análisis de requerimientos por parte del tutor, por lo que se estima que relacionada con las reuniones se invierte un valor de esfuerzo de 10 horas.

Para obtener el número de horas invertidas en la fase de desarrollo se hace uso de todos los *storypoints* de las tareas desarrolladas a lo largo de las iteraciones, calculando de esa manera el número final de 162 horas invertidas en el desarrollo.

Para el desarrollo del estudio económico solo se considera el trabajo invertido por el alumno, y no por el tutor, ya que se desconocen los datos necesarios para realizar las estimaciones. De la misma forma, tampoco se considera el tiempo realizado por el testeo del juego realizado por terceros. Ya que se desconoce el tiempo de juego y la calidad de las respuestas varía dependiendo del individuo. La Tabla 3 muestra las tareas realizadas a lo largo del proyecto y las horas de trabajo invertidas en cada una de ellas.

<b>Tarea</b>	<b>Estimación de horas alumno</b>
Reuniones organizadas	10
Fase de desarrollo	162
Total de horas	172

*Tabla 3. Tabla de tareas y horas del alumno*

Para calcular el coste de la mano de obra [33] es preciso conocer el coste que supone mantener a un empleado con las características requeridas para realizar las labores del alumno y docente [34].

La Tabla 4 muestra la tabla de los salarios a través de los cuales se obtiene el coste bruto de mantener a un trabajador.

---

[33] (Jobted, 2021)

[34] (Trecet, 2021)

<b>Posición</b>	<b>Salario neto por año</b>	<b>Salario neto por hora</b>	<b>Coste bruto por hora</b>
Programador de videojuegos junior	21.500€	11.2€	11.2 + 24% = 13.88€

*Tabla 4. Tabla de salarios*

Tras obtener el coste que le supone a la empresa mantener a un trabajador se puede calcular el coste total de la mano de obra, que se muestra a través de la Tabla 5.

<b>Posición</b>	<b>Horas</b>	<b>Coste bruto por hora</b>	<b>Coste total trabajador</b>
Programador de videojuegos junior	172 horas	13.88€ por hora	2388.73€
<b>Coste total de mano de obra</b>			2388.73€

*Tabla 5. Tabla de coste de mano de obra*

#### Costes de materiales

El coste de materiales hace referencia al gasto que se ha realizado a través de la pérdida de valor de los equipos de trabajo usados durante el desarrollo del proyecto. El desarrollo se lleva a cabo con un ordenador portátil y un teléfono móvil Android, por lo que los costes de materiales se calculan a través de la pérdida de valor del equipo de trabajo, como se indica en la Tabla 6.

<b>Equipo de trabajo</b>	<b>Coste del equipo</b>	<b>Tiempo de vida</b>	<b>Pérdida de valor</b>
HP ProBook 440 G7	889€	6 años	49.38€
LG G8s ThinQ	700€	5 años	46.66€
<b>Coste total de materiales</b>			96.04€

*Tabla 6. Tabla de coste de materiales*

#### Costes de instalaciones

El último tipo de coste se calcula a través del valor de las instalaciones y/o equipos usados a lo largo del proyecto. Dentro de esta categoría entraría el coste de los paquetes de assets adquiridos en el desarrollo del proyecto.

A pesar de ser gratuitos gracias al convenio con la Universidad, Epidemic Sound presenta un coste para todos los profesionales que hacen uso de sus servicios, por lo que se incluye en el coste del proyecto. De la misma manera se hace lo mismo con los paquetes de modelos usados en el juego.

La Tabla 7 presenta la tabla de coste de instalaciones.

<b>Servicio usado</b>	<b>Coste mensual</b>	<b>Meses</b>	<b>Coste del servicio</b>
Licencia personal de Epidemic Sound	13€	6	78€
Polygon Nature	44.66€	1	44.66€
Polygon Sci-Fi Space	107.19€	1	107.19€
<b>Coste total de instalaciones</b>			<b>229.85€</b>

*Tabla 7. Tabla de coste de instalaciones*

#### Costes totales del proyecto

Para concluir con el estudio económico se calculan los costes del proyecto en base a los costes previamente calculados, obteniendo de esa manera el coste total del proyecto. La Tabla 8 presenta el cálculo del coste total del proyecto.

<b>Tipo de coste</b>	<b>Valor</b>
Coste de mano de obra	2388.73€
Coste de materiales	96.04€
Coste de instalaciones	229.85€
<b>Coste total del proyecto</b>	<b>2714.62€</b>

*Tabla 8. Tabla de coste total del proyecto*



## 9. Resultados

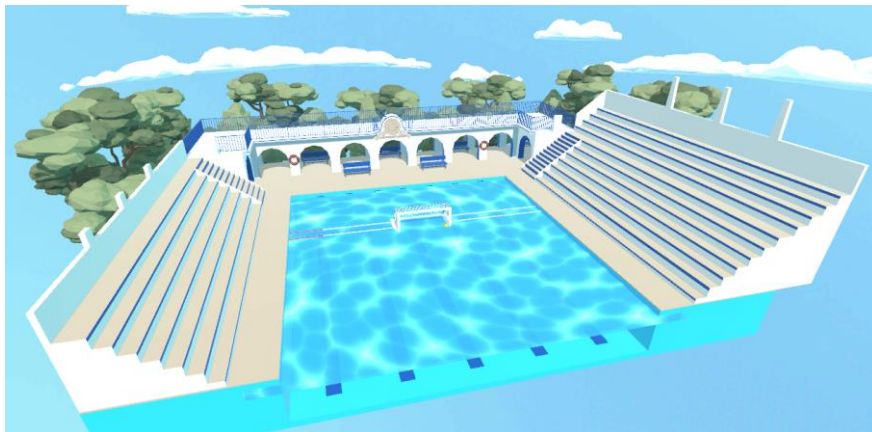
Este capítulo de la memoria recoge los diferentes resultados obtenidos a lo largo del desarrollo del proyecto. Para una mejor comprensión se divide el capítulo en 2 puntos. En el primero se hace un análisis del artefacto generado. Mientras que en el segundo se estudian las desviaciones con respecto a la metodología propuesta y los objetivos que se han cumplido o modificado a lo largo del proyecto.

### 9.1. Análisis del artefacto

Uno de los objetivos principales del proyecto es el desarrollo de un juego móvil con la suficiente calidad y características para poder competir en el mercado actual. Considerando el cumplimiento de este objetivo, en este punto de la memoria se realiza un pequeño recorrido a través de los diferentes elementos que componen el juego.

#### Escenario

Basado en una piscina real, el escenario del proyecto cuenta con múltiples elementos que ayudan al jugador a transportarse al interior de la pantalla. El elemento más importante de todos es el agua, que aunque tenga un diseño simple y su implementación no fuera muy costosa, une el resto de los detalles en la escena de forma homogénea. La portería, corcheras y balón son sin lugar a dudas elementos imprescindibles, uniendo el tema del juego con las mecánicas principales. Las gradas y arcos ayudan a obtener una sensación de lugar real, mientras que los árboles y demás detalles completan los detalles de la piscina, creando así el escenario deseado para el juego. La Figura 50 muestra el escenario final del juego.



*Figura 50. Escenario final del juego*

#### Personajes

Obtenidos a partir de un paquete de assets y modificados para aparentar jugadores reales de waterpolo, los personajes del juego forman la competencia a la que se enfrenta el jugador. Con el objetivo de parar balones, tanto el portero como los defensas utilizan algoritmos que adaptan



su rendimiento para darle al jugador una sensación incrementada de dificultad y satisfacción. Además, los defensores simulan la sensación de intensidad que permite que sus movimientos se acentúen con respecto al de los demás, diferenciando así un personaje del otro. La Figura 51 muestra al portero dentro del agua.



*Figura 51. Personaje del portero*

### Balón

El único elemento del juego con el que interactúa el jugador es el balón, el objeto más importante de todos. A través del dibujo de trazos en la pantalla, el jugador puede realizar lanzamientos hacia portería para conseguir marcar gol. El balón presenta físicas que permiten una simulación de lo más real posible, haciendo que bote, colisione, y vuele en la dirección que el jugador indique. Creando así una mecánica sencilla, entendible y satisfactoria para el jugador.

De la misma manera, el juego ofrece un pequeño tutorial que enseña a los jugadores los diferentes trazos que deben realizar para obtener el lanzamiento deseado. El tutorial ofrece además una descripción sobre el sistema de vidas que se usa para lograr que incluso sin haber jugado, el jugador pueda comprender qué, cómo y por qué del juego.

Un sistema de reposicionamiento proporciona al jugador la oportunidad de realizar lanzamientos desde múltiples partes de la piscina, añadiendo a la mecánica principal del juego la importante característica de rejugabilidad. Esta característica añade un valor extra al juego permitiendo que el jugador pueda disfrutar durante más tiempo del producto final.

### Interfaz de usuario

Con el objetivo de no quitar importancia a los elementos del escenario y jugabilidad, la interfaz de usuario presenta un diseño simple y minimalista. Para dar una sensación de homogeneidad en el diseño, se utiliza en todas las pantallas un panel negro translucido que ayuda a apreciar mejor los elementos que contiene, permitiendo al mismo tiempo ver los elementos que contiene la escena. Finalmente, tal y como se comenta anteriormente, se utilizan diferentes cámaras en





los menús para resaltar elementos de la escena, apuntando al reloj de sol y a la piscina desde diferentes ángulos que no se aprecian durante la fase de juego.

### Sonido

Otro de los elementos del juego que ayudan a generar una sensación de inmersión es el sonido. A través del sonido se puede crear tensión, añadir emoción y muchas otras cualidades que mejoran de forma indudable la calidad del juego. El proyecto trata de generar una sensación de realismo a través de efectos de sonido similares a los que se obtendrían de un partido de waterpolo real.

Estos diferentes elementos forman las bases del juego generado durante el desarrollo del proyecto. Obteniendo así un artefacto totalmente jugable que cumple con uno de los objetivos del trabajo de fin de grado, el desarrollo de un juego móvil para Android. La Figura 52 muestra la reacción de los defensores a un lanzamiento del jugador.



*Figura 52. Reacción de los defensas a un lanzamiento*

## **9.2. Desviaciones del proyecto**

En el capítulo cinco de la memoria se realiza una planificación estimada de la metodología a seguir a lo largo del desarrollo del proyecto. Sin embargo, en la fase de desarrollo se producen múltiples modificaciones en lo que se refiere a la pila de producto, desviando así el proyecto de la metodología original.

En el capítulo de metodología se especifican en la pila de productos una serie de requerimientos que se deben desarrollar a lo largo de las iteraciones. Si bien al principio del desarrollo los requerimientos permanecen iguales, conforme se van realizando reuniones y se avanza el proyecto aumenta la pila de producto. Estos cambios van desde el estudio de aplicaciones para realizar ciertas implementaciones a modificaciones en requerimientos anteriores para adaptarlos a otros requerimientos.

La desviación de los objetivos y la explicación detrás de dicha modificación se lleva a cabo en el capítulo final.





---

## **10. Conclusiones**

El capítulo nueve concluye el desarrollo del proyecto analizando el cumplimiento de los objetivos propuestos y futuras mejoras que se pudieran realizar para ofrecer más características dentro del juego o mejorar los actuales.

### **10.1. Cumplimiento de objetivos**

Los dos objetivos propuestos al comienzo del proyecto se cumplen de manera exitosa a lo largo del desarrollo. Uno de ellos consiste en la creación de un juego móvil, mientras que el otro consiste en mantener un proceso de trabajo por encima de una desviación no optimista. A continuación se realiza un análisis explicando el cumplimiento de ambos objetivos y las modificaciones ocurridas a lo largo del proceso de desarrollo.

#### *10.1.1. Creación de un juego móvil*

A pesar de que se consigue completar este objetivo del del proyecto, algunos de los subobjetivos propuestos para lograr alcanzarlo varían a lo largo del desarrollo. Para poder analizar detalladamente por qué se modifican o introducen nuevos subobjetivos, se realiza un estudio iteración por iteración explicando las razones detrás de los cambios.

##### Primera iteración

La primera iteración del flujo de desarrollo abarca la creación del concepto del juego. Es decir, una pequeña escena para poder estudiar las características del juego y su posible potencial. En esta fase se proponen tres tareas diferentes:

- Crear el shader de agua
- Modelar el escenario
- Aplicar físicas de flotación al balón

Sin embargo, durante el desarrollo de dichas subtareas se da la necesidad de implementar otros subobjetivos diferentes como son:

- Creación de una paleta de color: Para poder dar vida a los modelos una vez exportados a Unity es precisa la creación de una paleta o atlas. Esta paleta se crea a partir de los colores más comunes en la piscina durante un partido de waterpolo, y resulta necesaria para el modelaje del escenario.
- Obtención de una skybox: Tal y como se comenta en el capítulo 6, se deben cubrir los alrededores de la escena de alguna manera para obtener una atmósfera para el juego. Obtener la skybox adecuada supone buscar en la Asset Store y probar diferentes modelos hasta dar con el idóneo, lo que se considera otro subobjetivo dentro de la iteración.



### Sexta iteración

La sexta fase de desarrollo del proyecto supone la creación de la última versión del juego programada para el proyecto. En esta iteración se añaden los últimos detalles para elevar la calidad del juego todo lo posible. Las tareas que se proponen durante esta iteración son:

- Diseño e implementación de los menús
- Búsqueda y creación de efectos de sonido
- Edición e implementación de efectos de sonido
- Creación del tutorial

Sin embargo, durante el desarrollo de las tareas se necesitan implementar algunas otras, como son:

- Creación del logo del juego: Con el fin de crear la aplicación final es necesaria la creación del logo de la aplicación. Para ello, se utiliza una captura de una de las cámaras del juego simulando la flotación del balón pegado a la portería. Más tarde se edita la imagen añadiéndole los efectos de postprocesado necesarios para obtener la versión final del logo. La Figura 53 muestra el logo del juego.



*Figura 53. Logo de la aplicación móvil del juego*

- Creación del título del juego: Tal y como se comenta durante el desarrollo de la sexta iteración. Al no disponer de las herramientas suficientes para poder mostrar el título del juego, se hace uso de otras aplicaciones como Maya. Sin embargo, también se realizan otras pruebas con otras herramientas para obtener el resultado deseado, prefiriendo finalmente la versión de Maya. Esto hace que se incorpore la tarea como un objetivo de la iteración, ya que su cumplimiento resulta necesario para implementar los menús de la aplicación.

Finalmente, tras analizar las modificaciones de las tareas, cabe resaltar otro cambio en cuanto al objetivo de desarrollar el juego móvil. Este cambio se da cuando se propone el número de *storypoints* requerido para la implementación de los subobjetivos, que siempre acaba

umentando una vez se completan los subobjetivos, incrementando consecuentemente el tiempo de desarrollo del proyecto.

### 10.1.2. Control del proceso de trabajo

El segundo de los objetivos que se proponen para el proyecto es el seguimiento y control del progreso realizado para garantizar que no cae por debajo de cierta desviación. Como se comenta en el capítulo de la metodología, se hace uso de una herramienta gráfica llamada burn-up, que sirve para visualizar las cargas de trabajo a lo largo del desarrollo. Con esta herramienta se gestiona el progreso del trabajo y la proyección a futuro.

El gráfico se actualiza conforme se realizan las tareas, para de esta manera visualizar cuanta carga de trabajo queda para finalizar la iteración o versión propuesta. La Figura 54 muestra la tabla de trabajo con sus diferentes versiones.

	Pre-development	Concept	Prototype 0.1	Alpha 1.1	Alpha 1.2	Beta 2.1	Master Candidate
	02/02/21	02/02/21	02/26/21	03/24/21	04/28/21	05/14/21	05/26/21
<b>Pre-development</b>	0	0	0	0	0	0	0
<b>Concept</b>	32	32	32	32	32	32	32
<b>Prototype 0.1</b>	68	68	68	68	68	68	68
<b>Alpha 1.1</b>	104	104	104	104	104	104	104
<b>Alpha 1.2</b>	128	128	128	128	128	128	128
<b>Beta 2.1</b>	148	148	148	148	148	148	148
<b>Master Candidate</b>	182	182	182	182	182	182	182
<i>Planned Development</i>	0	32	68	104	128	148	182
<b>Actual Development</b>	0	32	68	104	128	148	182
Deviation 7% Not Optimistic	0	29,76	63,24	96,72	119,04	137,64	169,26
Deviation 7% Optimistic	0	34,24	72,76	111,28	136,96	158,36	194,74

Figura 54. Tabla del gráfico burn-up

Tal y como se aprecia en la tabla, la planificación del desarrollo coincide con el desarrollo actual de los objetivos en las iteraciones. Cumpliendo con el objetivo en todo momento de permanecer por encima del límite de desviación pesimista. La Figura 55 muestra el gráfico burn-up.

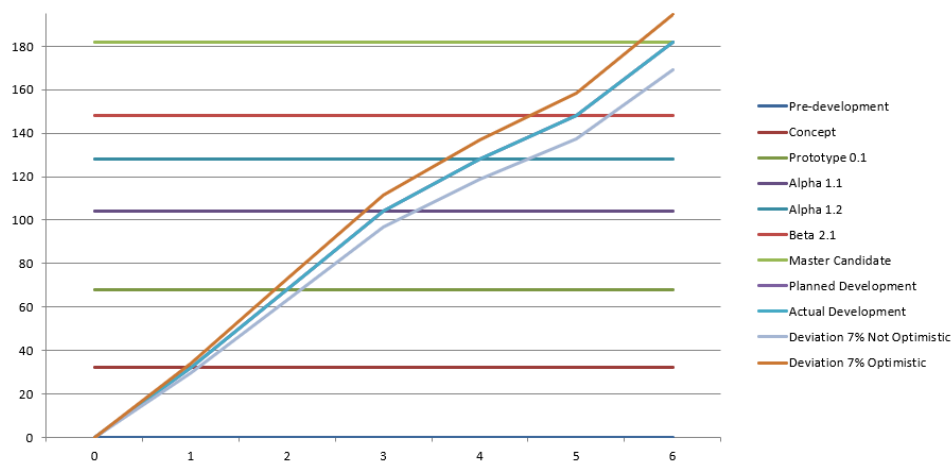


Figura 55. Gráfico burn-up

La conclusión que se puede obtener a lo largo del desarrollo del proyecto es se cumple con todos los objetivos propuestos en la planificación, a pesar de que algunos hayan tenido que ser modificados. En definitiva, Waterpolo Taquigol es un juego móvil funcional que cumple con la misión especificada desde el principio, proporcionar a jugadores de waterpolo un videojuego dirigido especialmente para ellos.

### **10.2. Futuras mejoras**

El trabajo de fin de grado no abarca todas las posibles implementaciones que se podrían realizar en un juego móvil con las características de Waterpolo Taquigol, y es por eso por lo que se decide crear una lista de mejoras y ampliaciones que se podrían realizar en el futuro para cubrir el potencial que posee el proyecto:

- Opción para cambiar de idioma
- Subir el juego a la Play Store
- Crear una versión para iOS
- Incluir un modo partido
- Personalizar jugadores y colores

La implementación actual de Waterpolo Taquigol cuenta con los requerimientos suficientes para poder cubrir la misión del proyecto, proporcionar a jugadores de waterpolo un videojuego dirigido especialmente para ellos. Sin embargo, esta razón no puede hacer que se dé por terminado el proyecto, ya que existen muchas formas de incrementar la calidad de la aplicación.

Es por eso por lo que el proyecto conocido como Waterpolo Taquigol no termina con la versión actual, sino que se continuará la implementación para construir de esa manera la mejor versión posible del juego móvil.



## **11. Bibliografía**

- [1] González, D. J. (abril de 2011). El origen y la historia del waterpolo. Obtenido de <https://www.efdeportes.com/efd155/el-origen-y-la-historia-del-waterpolo.htm>
- [2] Ruiz, E. (11 de enero de 2015). Posiciones en el waterpolo. Obtenido de <https://waterpolomadrid.com/2015/01/11/posiciones-en-el-waterpolo/>
- [3] PubNative. (20 de abril de 2020). From Tetris to Candy Crush: The History of Mobile Gaming. Obtenido de <https://pubnative.net/blog/from-tetris-to-candy-crush-the-history-of-mobile-gaming/>
- [4] Brown, A. (26 de marzo de 2020). How the Iconic "Snake" Game was Created for Nokia Phones? Obtenido de <https://www.tabbloidx.co/how-the-iconic-snake-game-was-created/>
- [5] Admixer. (10 de junio de 2020). In-Game Advertising: 8 Ad Formats to Monetize Mobile Games. Obtenido de <https://blog.admixer.com/in-game-advertising-8-ad-formats-monetize-mobile-games/>
- [6] Wikipedia. (31 de mayo de 2021). Sports video game. Obtenido de [https://en.wikipedia.org/wiki/Sports\\_video\\_game](https://en.wikipedia.org/wiki/Sports_video_game)
- [7] Minijuegos. (6 de junio de 2021). minijuegos. Obtenido de <https://www.minijuegos.com/>
- [8] Wikipedia. (26 de mayo de 2021). Hyper-casual game. Obtenido de [https://en.wikipedia.org/wiki/Hyper-casual\\_game](https://en.wikipedia.org/wiki/Hyper-casual_game)
- [9] Google. (junio de 2021). GooglePlay. Obtenido de [https://play.google.com/store/apps?utm\\_source=emea\\_Med&utm\\_medium=hasem&utm\\_content=Jun3015&utm\\_campaign=Evergreen&pcampaignid=MKT-EG-emea-es-1001280-Med-hasem-ap-Evergreen-Jun3015-1%7cONSEM\\_kwid\\_43700007551007979&gclid=Cj0KCQjw5PGFBhC2ARIsAIFIMNddq4fRjO-](https://play.google.com/store/apps?utm_source=emea_Med&utm_medium=hasem&utm_content=Jun3015&utm_campaign=Evergreen&pcampaignid=MKT-EG-emea-es-1001280-Med-hasem-ap-Evergreen-Jun3015-1%7cONSEM_kwid_43700007551007979&gclid=Cj0KCQjw5PGFBhC2ARIsAIFIMNddq4fRjO-)
- [10] García, M. (mayo de 2020).
- [11] Brackeys. (19 de mayo de 2019). SIMPLE CARTOON WATER in Unity. Obtenido de [https://www.youtube.com/watch?v=Vg0L9aCRWPE&ab\\_channel=Brackeys](https://www.youtube.com/watch?v=Vg0L9aCRWPE&ab_channel=Brackeys)
- [12] Synty Studios. (mayo de 2021). Synty Studios 3D Art For Game. Obtenido de <https://www.syntystudios.com/>
- [13] Petty, J. (mayo de 2021). What is 3D Rigging For Animation & Character Design? Obtenido de <https://conceptartempire.com/what-is-rigging/>
- [14] Epidemic Sound. (mayo de 2021). Epidemic Sound. Obtenido de <https://www.epidemicsound.com/music/featured/>

- 
- [15] Unity. (28 de mayo de 2021). Profiler overview. Obtenido de <https://docs.unity3d.com/Manual/Profiler.html>
- [16] Beas, J. M. (19 de mayo de 2015). Burn-up chart. Obtenido de <https://blog.jmbeas.es/2015/05/19/burn-up-chart/>
- [17] Brackeys. (septiembre de 2020). GameDev Tutorials Brackeys. Obtenido de <https://www.youtube.com/c/Brackeys/videos>
- [18] Unity. (4 de junio de 2021). Universal Render Pipeline overview. Obtenido de <https://docs.unity3d.com/Packages/com.unity.render-pipelines.universal@12.0/manual/index.html>
- [19] Unity. (2016). Skybox. Obtenido de <https://docs.unity3d.com/es/530/Manual/class-Skybox.html>
- [20] Knight, R. (18 de septiembre de 2020). Fantasy Skybox FREE. Obtenido de <https://assetstore.unity.com/packages/2d/textures-materials/sky/fantasy-skybox-free-18353>
- [21] Khan Academy. (2020). What is buoyant force? Obtenido de <https://www.khanacademy.org/science/physics/fluids/buoyant-force-and-archimedes-principle/a/buoyant-force-and-archimedes-principle-article>
- [22] Unity. (2021). Unity Forum. Obtenido de <https://forum.unity.com/>
- [23] Maya Learning Channel. (24 de agosto de 2016). Quick Rigging and Skinning a character for beginners. Obtenido de [https://www.youtube.com/watch?v=c538zkwxgTQ&ab\\_channel=MayaLearningChannel](https://www.youtube.com/watch?v=c538zkwxgTQ&ab_channel=MayaLearningChannel)
- [24] Yawyee, L. C. (7 de enero de 2021). Body Mechanics - Maya Beginner's Animation Tutorial | In 5 simple steps. Obtenido de [https://www.youtube.com/watch?v=7CBcvu8HLEQ&ab\\_channel=LEARNCGIwithYawyee](https://www.youtube.com/watch?v=7CBcvu8HLEQ&ab_channel=LEARNCGIwithYawyee)
- [25] Blackthornprod. (13 de enero de 2018). HOW TO EXPORT ANIMATIONS FROM MAYA TO UNITY - TUTORIAL. Obtenido de [https://www.youtube.com/watch?v=i8cmDK\\_O-1g&ab\\_channel=Blackthornprod](https://www.youtube.com/watch?v=i8cmDK_O-1g&ab_channel=Blackthornprod)
- [26] Google. (2021). Google Fonts. Obtenido de <https://fonts.google.com/>
- [27] Brackeys. (6 de diciembre de 2017). SETTINGS MENU in Unity. Obtenido de [https://www.youtube.com/watch?v=YOaYQrN1oYQ&ab\\_channel=Brackeys](https://www.youtube.com/watch?v=YOaYQrN1oYQ&ab_channel=Brackeys)
- [28] Epidemic Sound. (2021). Subscriptions. Obtenido de <https://www.epidemicsound.com/pricing/>
- [29] Reaper. (18 de mayo de 2021). Reaper Digital Audio Workstation. Obtenido de <https://www.reaper.fm/>
-



- 
- [30] 123APPS. (2021). MP3 CUT. Obtenido de <https://mp3cut.net/es/>
- [31] Vegas, J. (6 de febrero de 2020). HOW TO EASILY PLAY A VIDEO IN UNITY TUTORIAL. Obtenido de [https://www.youtube.com/watch?v=p7iXEZGx2Mc&ab\\_channel=JimmyVegas](https://www.youtube.com/watch?v=p7iXEZGx2Mc&ab_channel=JimmyVegas)
- [32] UPV, U. P. (19 de octubre de 2017). EL ESTUDIO ECONÓMICO EN PROYECTOS DE INGENIERÍA. Obtenido de [https://www.youtube.com/watch?v=rCMPE3utkEQ&ab\\_channel=UniversitatPolit%C3%A8cnica-de-Valencia-UPV](https://www.youtube.com/watch?v=rCMPE3utkEQ&ab_channel=UniversitatPolit%C3%A8cnica-de-Valencia-UPV)
- [33] Jobted. (2021). Sueldo del Programador de Videojuegos en España. Obtenido de <https://www.jobted.es/salario/programador-videojuegos>
- [34] Trecet, J. (29 de marzo de 2021). IRPF 2021: así funcionan los tramos. Obtenido de <https://www.finect.com/usuario/Josetrecet/articulos/como-funciona-renta-tramos-irpf>