



Universidad de Jaén

Centro de Estudios de Postgrado

Trabajo Fin de Máster

DESARROLLO DE JUEGOS PARA DISPOSITIVOS MÓVILES

Alumno: Pérez Sánchez, Carlos

Tutora: Prof. D.^a Carmen Martínez Cruz
Dpto: Informática

Junio, 2021

Contenido

I.	Resumen	5
I.1	Resumen y palabras clave	5
I.2	Abstract and keywords.....	5
II.	Introducción al TFM	6
III.	Bloque 1. Fundamentación epistemológica.	8
III.1	Introducción tema.....	8
III.2	Antecedentes	8
III.2.1	Introducción a los videojuegos	8
III.2.2	Historia.....	10
III.2.3	Clasificación	16
III.2.4	Perfiles laborales.....	21
III.2.5	Fases de desarrollo de un videojuego	25
III.2.6	Características específicas de los dispositivos móviles	30
III.2.7	Formatos de hardware	30
III.2.8	Sistemas Operativos	31
III.3	Desarrollo de juegos móviles	32
III.3.1	Lenguajes y motores de desarrollo.....	32
III.3.2	Desarrollo para dispositivos móviles	37
III.3.3	Publicación y distribución	52
III.4	Tendencias actuales y futuras.....	54
III.4.1	Realidad Virtual (VR).....	54
III.4.2	Realidad Aumentada (AR).....	55
III.4.3	Juegos en la nube.....	57
III.4.4	Otros sectores.....	58
III.5	Conclusiones	59
IV.	Bloque 2. Proyección Didáctica	62
IV.1	Introducción	62
IV.2	Contextualización del centro	63
IV.2.1	Plan de centro.....	63
IV.2.2	Características del alumnado.....	65

IV.3	Unidad Didáctica	66
IV.3.1	Introducción	66
IV.3.2	Descripción de la Unidad Didáctica	66
IV.3.3	Normativa	67
IV.3.4	Competencia General	67
IV.3.5	Orientaciones pedagógicas	67
IV.3.6	Competencias profesionales, personales y sociales	68
IV.3.7	Líneas de actuación	68
IV.3.8	Objetivos Generales	68
IV.3.9	Resultados de aprendizaje	69
IV.3.10	Contenidos	69
IV.3.11	Objetivos didácticos	69
IV.3.12	Metodología	70
IV.3.13	Evaluación y Recuperación	76
IV.3.14	Transversalidad	79
IV.3.15	Atención a la Diversidad	80
IV.3.16	Anexos	81
V.	Índice de Tablas	85
VI.	Índice de ilustraciones	85
VII.	Bibliografía	87

I. Resumen

I.1 Resumen y palabras clave

Este Trabajo de Fin del Master de Profesorado tiene como objetivo sentar las bases y desarrollar una Unidad Didáctica dentro del campo de la creación de aplicaciones para dispositivos móviles.

La tendencia actual al uso de masivo de estos dispositivos debido a su facilidad y comodidad en el transporte, así como la potencia cada vez más elevada de los mismos, asemejándose a ordenadores personales, hace que en el campo del entretenimiento, se eleven las expectativas de encontrar aplicaciones cada vez más sofisticadas. En este sentido, el objetivo de este TFM es centrarse en el desarrollo de videojuegos dirigidos a plataformas móviles.

La Unidad Didáctica "Desarrollo de juegos para dispositivos móviles" se encuadra dentro del módulo de "Programación multimedia y dispositivos móviles (0489)" y que se enmarca dentro del Ciclo Formativo de Grado Superior de Desarrollo de Aplicaciones Multiplataforma.

Palabras clave: Videojuegos, dispositivos móviles, Motores de videojuegos, Unreal Engine, Android

I.2 Abstract and keywords

This Master's Tesis arises with the aims to lay the foundations and develop a Didactic Unit within the field of the creation of applications for mobile devices.

The current trend towards the massive use of these devices due to their ease and convenience in transport, as well as their increasing power, making them similar to personal computers, means that in the field of entertainment, the expectations of finding increasingly sophisticated applications are rising. In this sense, the aim of this TFM is to focus on the development of video games aimed at mobile platforms.

The Didactic Unit "Development of games for mobile devices" is part of the module "Multimedia programming and mobile devices (0489)", which is part of the Advanced Vocational Training on Multi-platform Application Development.

Key words: Videogames, mobile devices, Game engines, Unreal Engine, Android

II. Introducción al TFM

Este Trabajo Fin de Máster se estructura siguiendo la siguiente línea: en el **Bloque 1**, se lleva a cabo un “Estudio Epistemológico” con el fin de entender los antecedentes del desarrollo de los videojuegos, las características más importantes para desarrollar juegos destinados a plataformas digitales y las optimizaciones pertinentes necesarias a tener en cuenta cuando se desarrollan juegos destinados a este tipo de plataformas. Finalmente, se definen las tendencias actuales y futuras que se están produciendo en esta industria. En el **Bloque 2**, se desarrolla la “Proyección Didáctica” cuya base se sustenta en la normativa educativa sobre Formación Profesional vigente. Este TFM también contempla la situación anómala surgida con la pandemia de COVID-19. La Unidad Didáctica “Desarrollo de juegos para dispositivos móviles” desarrollada en este bloque, pretende ser una guía docente para conseguir una formación de calidad que permita al alumnado adaptarse a los distintos aspectos de su futuro profesional. Finalmente se incluye una sección de Anexos donde es posible consultar información detallada relativa a la Unidad Didáctica desarrollada.

BLOQUE 1

FUNDAMENTACIÓN EPISTEMOLÓGICA

III. Bloque 1. Fundamentación epistemológica.

III.1 Introducción tema

En este bloque se realiza un Estudio Epistemológico del desarrollo de videojuegos, primero haciendo una contextualización global de la industria del videojuego y de su historia para posteriormente centrarse en el desarrollo de juegos móviles.

Se estructura siguiendo el siguiente esquema: Primero se realiza un estudio de la bibliografía de los antecedentes de los videojuegos, describiendo el concepto de videojuego, la historia, los métodos de clasificación de videojuegos, los distintos perfiles laborales dentro de la industria así como las fases de desarrollo que se sigue a la hora de la creación de juegos.

Los anteriores puntos ayudan a realizar una contextualización general de todo lo relacionado con los videojuegos. Las siguientes secciones se centran específicamente en el desarrollo de juegos móviles: Características específicas de los dispositivos móviles, los formatos de hardware así como de los sistemas operativos más utilizados.

Los siguientes capítulos se centran en el desarrollo de juegos para plataformas móviles analizando los motores de videojuegos y los lenguajes utilizados para este propósito. También se analizan las diferencias entre el desarrollo de juegos para consolas/PC y móviles, así como las pautas necesarias para este tipo de desarrollo.

Posteriormente se listarán las diferentes formas de publicar y distribuir videojuegos, parte fundamental de esta industria.

Para finalizar se analizan las tendencias futuras que pueden marcar el porvenir del desarrollo de juegos, algunas de ellas implantadas actualmente en la industria.

III.2 Antecedentes

III.2.1 Introducción a los videojuegos

¿Qué es un **videojuego**? Existe en la bibliografía una gran multitud de definiciones de videojuego, aunque no todos los autores de las mismas se ponen de acuerdo en lo que cada uno de ellos entiende por dicho término.

Quizá, bajo mi punto de vista, una de las definiciones menos acertadas, quizá debido al tiempo que ha pasado desde que se concibió y la evolución posterior del término, la propone la **Real Academia Española**: “1. m. Dispositivo electrónico que permite, mediante mandos apropiados, simular juegos en las pantallas de un televisor o de un ordenador.” (Real Academia Española, s.f.). Esta definición no contemplaría entre otros,

a cualquier producto desarrollado para consolas portátiles, que es el motivo de este TFM. La RAE, en esta acepción, incluye la palabra juego, que en su definición previa expone que *“Un juego es un ejercicio recreativo o de competición sometido a reglas, y en el cual se gana o se pierde”* (Real Academia Española, s.f.), es decir, que un juego es una actividad que requiere de al menos un jugador, tiene unas reglas y objetivos claros conocidos por el jugador y al menos tener una condición para perder o ganar (Rogers, 2014).

Para (Vigueras, 2001) *“Los videojuegos son un medio de entretenimiento con características propias que tiene diferencias y similitudes con otros medios. Su narrativa es cautivadora, pues [...] y posee representaciones simbólicas [...] y logran una concentración absoluta en los jugadores”*, poniendo su foco de atención en la narrativa de los videojuegos, perdiendo por el camino una multitud de productos que actualmente podemos encontrar en el mercado.

(Provenzo, 1991), enfatiza el poder de los videojuegos como una puerta abierta por la sociedad de consumo, que permite conectarnos con enormes y poderosas redes de información, comunicación, ocio y entretenimiento. Mientras que (Estallo, 1995) defiende los videojuegos como una forma de ejercicio cognitivo y gimnasia mental, realizado por el usuario para satisfacer necesidades intrínsecas y disfrutar del tiempo de ocio jugando con objetos interactivos que permiten: alcanzar un determinado objetivo, un estado de ánimo, algunas metas, ejecutar acciones fantásticas y adquirir algunos niveles de control y destreza.

(Martínez Borda, 2006), describe los distintos niveles de interacción entre el medio y el jugador, *“porque, no es lo mismo jugar en una máquina tragaperras (electrónica, lúdica, informática o soporte visual) que jugar con un joystick en una pantalla de ordenador o televisión. La interacción que se da entre el medio y el jugador no es la misma”*. Mientras que, por ejemplo (Levis, 1997) no toma en consideración esta interacción: *“Un videojuego consiste en un entorno informático que reproduce sobre una pantalla un juego cuyas reglas han sido previamente programadas”*

A partir de las definiciones anteriores, y a modo de resumen, podemos definir un videojuego como (García Gigante, s.f.):

- Un **tipo especial de juego** (de esta forma, en la medida en que hay juegos educativos también existen los videojuegos educativos, y en la medida en que el objetivo final de un juego es la diversión, el carácter lúdico del videojuego está presente).
- Que para su uso requiere de algún medio o tipo de **soporte electrónico**, y que suele contar con distintos periféricos para conectarse a él: mando (*gamepad*), *joystick*, teclado y ratón...

- Que dicho medio se caracteriza por permitir la **interacción** entre el videojuego y uno o varios jugadores (sin necesidad de limitar su número para que tengan cabida los juegos on-line). Esta interacción o acciones que el jugador puede realizar dentro del videojuego recibe el nombre de **mecánicas**.
- Que dicha interacción es consecuencia de la información que el usuario recibe del medio y que puede ser visual, auditiva -voces y música- o táctil.

En las puntualizaciones anteriores se ha establecido que uno de los elementos importantes en el mundo y desarrollo de videojuegos es el concepto de las **mecánicas**.

Si tomamos la definición que propone **Navarro**, *“las mecánicas son los verbos, las acciones que el jugador ejecuta y son posibles gracias a las reglas. Vinculan objetos del videojuego con consecuencias para el game state –estado del juego- de una manera prescrita, fija y casual. Cuando el jugador ejecuta una mecánica, sabe que las reglas de juego determinarán unas consecuencias particulares. Utilizar las mecánicas es modificar los atributos de los objetos o incluso el gameworld – el mundo dentro del videojuego”* (Navarro Remesal, 2016)

Además, (Navarro Remesal, 2016) argumenta que las mecánicas requieren de un aprendizaje y pueden ser dominadas por el jugador, que *“ejecuta una acción que crea una reacción en el sistema central que a su vez envía un output al jugador, a causa de la cual el jugador ve modificados sus conocimientos y habilidades”*

A partir de la definición que Navarro propone, podemos definir las mecánicas como todas las acciones que hace el jugador dentro del juego. En *Super Mario*, las mecánicas serían saltar o avanzar, en juegos como *Call of Duty* o *Battlefield*, disparar, agacharse o correr y en *FIFA*, esprintar o disparar a puerta.

III.2.2 Historia

En 1952, *AS Douglas* presentó *OXO* como parte de su tesis doctoral para la Universidad de *Cambridge*. Estaba inspirado en el conocido tres en raya y permitía a un jugador competir contra la máquina. Se considera como el primer videojuego de la historia.

En 1958 llegó, de la mano de *William Higginbotham*, *Tennis for Two*, un simulador que mediante un programa de cálculo y un osciloscopio permitía el juego entre dos jugadores (Ilustración 1). Cuatro años más tarde, en 1962 en el MIT veía la luz *Spacewar!*, un juego para ordenador que usaba gráficos vectoriales diseñado por *Steve Russell*.

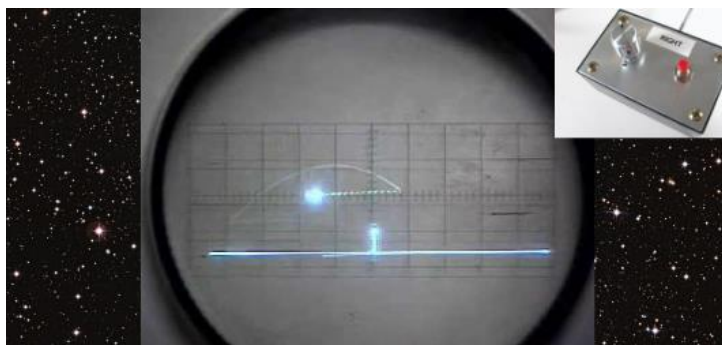


Ilustración 1: *Tennis for Two*. Fuente: (*The History of Video Games*, s.f.)

En el 1966, *Ralph Baer* comenzó a trabajar en el prototipo de un dispositivo para conectarlo a un televisor y que permitiera al usuario poder jugar con éste. La denominó *Brown Box*. En un principio no tuvo mucho éxito, pero dos años más tarde, *Magnavox* se interesó por el prototipo y en 1972 fue presentado con el nombre de *Magnavox Odyssey*, la primera videoconsola de la historia.

A finales de los años 60, un estudiante de la Universidad de *Stanford*, *Bill Pitts*, tuvo la idea de hacer una versión de *Spacewar!* que funcionara con monedas para salones recreativos. Desarrolló *Galaxy Game*, la primera máquina recreativa de videojuegos de la historia.

En 1972, *Nolan Bushnell* y *Ted Dabney* fundaron *Atari Inc*, siendo una empresa pionera en los juegos *arcade*, las consolas de videojuegos y ordenadores domésticos. Productos como *PONG*, que en 1974 generaba 250 millones de dólares anuales, hizo que se reestructurara por completo el negocio del entretenimiento. *Atari* comenzó a publicar juegos como, *Space Race*, *Rebound*, *Gotcha*, *Tank*, *Qwark!*, ampliando los géneros de los videojuegos que hasta entonces existían. En 1975, *Tele-Games Pong*, la primera consola doméstica de *Atari*, fue un éxito en ventas.

Mientras que las máquinas *arcade* y las consolas domésticas eran todo un éxito, los ordenadores no permitían la creación de juegos de acción (muy populares en esta época) al carecer de monitor y por disponer de un sistema lento. Los únicos juegos posibles para ordenadores eran los conversacionales o por turnos. En 1976, *Willie Crowther* desarrolló *Colossal Cave Adventure* uno de los juegos más influyentes para la creación de los juegos de rol modernos y las aventuras gráficas.

Entre los años 1977 y 1980, la industria de los videojuegos fue frenética, las compañías rivales de *Atari* comenzaban a superarla fabricando consolas que incluían microprocesadores más potentes e innovadores. *Atari* fue vendida a *Warner Bros* para poder lanzar al mercado la *Atari VCS 2600*.

En Japón, *Taito Corporation* lanzó en 1978, *Space Invaders* que se convirtió en el videojuego más popular de la compañía, y uno de los más memorables juegos de la historia de las máquinas de salón. Este éxito colocó a Japón en el epicentro del desarrollo

de videojuegos. *Space Invaders* estaba inspirado en tanques y aeronaves pero su creador *Toshihiro Nishikado* cambió el diseño inicial por naves espaciales, debido a la influencia de “La Guerra de las Galaxias” de *George Lucas* en aquellos años.

En los años 80, fue tal el éxito de los videojuegos que las principales compañías de entretenimiento pasaron a tener su división de videojuegos. *Lucasfilm*, *Walt Disney Studios* o *20th Century* son ejemplos de ello. Lo que realmente revolucionó la industria fue la llegada de los gráficos a color con *Galaxian*, desarrollada por la empresa *Namco*.

En 1980, *Toru Iwatani* de la empresa *Namco*, desarrolló *Pac-Man*, un juego de laberintos libre de violencia y con una dinámica de juego relajada (Ilustración 2). Consiguió disparar las cifras de recaudación atrayendo masivamente al público femenino.



Ilustración 2: *Pac-Man*. Fuente: (Espacio Psicofamiliar, s.f.)

Paralelamente, *Activision* se convirtió en la primera empresa externa en desarrollar juegos para *Atari*.

En 1981, inicio de la era de los 8 bits, *Sinclair* fabricó su ordenador personal *ZX81* y poco después apareció *Football Manager* dando inicio al género de estrategia deportiva. Un año después salió al mercado el *ZX Spectrum* y juegos como *Jet Set Willy* y *Manic Miner* que inauguran el género de plataformas.

En 1984, *Alekséi Pázhitnov* desarrolló *Tetris*, un videojuego de lógica, tan innovador que su influencia e impacto sigue patente entre las nuevas generaciones de jugadores y desarrolladores.

Mientras que la industria norteamericana entró en la crisis del videojuego a inicios de los 80, Japón era todo lo contrario. En 1983, *Nintendo* lanzó la *Famicon*, una consola inspirada en la *Atari 2600* pero con mejores prestaciones. Y desarrolló títulos como *Donkey Kong*, *Mario Bros*, *Dragon Quest*, *Legend of Zelda* o *Final Fantasy*. Aprovechando

la crisis norteamericana, *Nintendo* se asentó en EEUU, cambiando el nombre de la *Famicon* por *NES*, *Nintendo Entertainment System*, y con juegos como *Super Mario Bros*, diseñado por *Shigeru Miyamoto*, vendió millones de copias, colocando a la empresa como referente a nivel mundial y estableciendo estándares para el desarrollo de videojuegos. Otra de las empresas niponas a destacar es *Capcom* con juegos como *Commando*, *Mega Man* o la saga de *Street Fighter*.

Uno de los éxitos de finales de los 80 que cabe destacar es *Prince of Persia*, desarrollado íntegramente por *Jordan Mechner* para *Apple II*. Este juego sería de una gran influencia para juegos como *Tomb Raider*.

Años 90, época en la que aparecieron los 16 bits, y con ello el renacimiento de las aventuras gráficas para ordenador. *Lucas Arts* sería una de las compañías más importantes a nivel mundial con sagas como *Indiana Jones* o *The secret of Monkey Island*.

Tras varios intentos fallidos, *SEGA* se introdujo en la industria con la *MegaDrive* y el juego de plataformas *Sonic: The Hedgehog*. En 1991, *Nintendo* lanzó la *Super Nintendo* con el juego *Mario Kart* como juego principal. Previamente *Nintendo* en 1989 lanzó la *Game Boy*, consola portátil que fue un éxito absoluto gracias a su extenso catálogo de juegos y a su reducido precio.

En 1993, la empresa *id Software* desarrolló *DOOM*, un juego de disparos de extrema violencia. Es considerado como uno de los videojuegos de disparos en primera persona pioneros en implementar características novedosas como la posibilidad de jugar *online* y los gráficos en 3D. Esto supuso un punto de inflexión en el desarrollo de videojuegos, dejando atrás el 2D para dar paso al 3D.

Sony creó una división de videojuegos que le permitió entrar en la industria. Ni *SEGA* con la *Sega Saturn*, ni *Nintendo* con la *Nintendo 64* y juegos como *Super Mario 64*, *Legend of Zelda Ocarina of Time* o *Donkey Kong 64*, pudieron hacer frente al éxito de la nueva consola de *Sony*, la *PlayStation* o *PS1* en 1994. Juegos como *Tekken*, *Final Fantasy VII*, *Tomb Raider II*, *Crash Bandicoot*, *Need For Speed*, *Metal Gear Solid* o *Resident Evil* fueron algunos de los juegos lanzados por *Sony*, convirtiendo a la *PS1* a finales del siglo XX en una de las consolas más importantes y vendidas de la industria superando las cien millones de unidades.

Sony lanzó en el 2000 la *PlayStation 2* o *PS2* y se convertiría en la consola más vendida del mundo con más de 160 millones de unidades. Un año más tarde *Microsoft* entra en el mercado presentando la *Xbox*, que no consiguió el éxito esperado y *Nintendo* lanzó la *Game Cube* que fue un fracaso de ventas.

Nintendo lanza en el 2004 la *Nintendo DS*, una videoconsola portátil que tenía como característica una segunda pantalla táctil. Las nuevas versiones de ésta, como la

Nintendo DSi implementan cámara y conexión mediante *Wifi*. *Sony* lanzo la *PSP*, una consola portátil que no llegó al nivel de ventas esperado.

Mientras tanto, para ordenadores, tenían gran éxito juegos de acción *online* como *Call of Duty* o *Battlefield 1942* o los juegos de estrategia en tiempo real como *Age of Empires* o el *Warcraft* de la empresa *Blizzard*. Esta compañía también desarrolló *Starcraft* y la saga de *Diablo* y revolucionó el juego *online* creando *Battle.net*, un servicio que permite a jugadores de ciertos videojuegos comunicarse por medio de canales de *chat* y jugar partidas de los juegos que tienen la capacidad de conectarse a esta plataforma. En 2004 publica *World of Warcraft* con el que bate todos los records tanto de ventas como en jugabilidad. Es el inicio de los juegos *MMORPG*, videojuegos de rol multijugador masivos en línea.

Un año antes, en 2003, la empresa *Valve* lanza *Steam*, una plataforma de distribución digital de videojuegos, gestión digital de derechos, comunicaciones y servicios *multijugador*. Aunque al principio no tuvo mucho éxito, *Valve* agregó mejoras como la lista de amigos, el mercado, los foros o el *steam cloud*, el almacenamiento en la nube. Esto, sumado al lanzamiento del *Half Life 2* exclusivo para *Steam* hizo que la plataforma empezara a crecer rápidamente, convirtiéndose en la principal referencia de los usuarios con más de 6000 juegos y 150 millones de jugadores activos.

Microsoft lanzó la *Xbox 360* en 2005 y *Sony* hizo lo mismo con la tercera versión de la *PlayStation*, la *PS3*, que integraba tecnología *Blu-ray*. *Nintendo* presentó la *Wii*, una consola que se manejaba captando el movimiento del cuerpo usando mandos.

El gran triunfo de *Grand Theft Auto III*, *GTA*, de *Rockstar Games* que mezclaba historias complejas con libertad de movimiento del jugador dio lugar a una de las mayores franquicias de los videojuegos modernos. Al mismo tiempo se desarrollaron populares sagas como: *Final Fantasy*, *Call of Duty*, *Prince of Persia*, *Halo*, *Medal of Honor*, *Street Fighter*, *Resident Evil*, *Silent Hill* o *Assassins Creed*.

En 2011, *Nintendo* lanzó la consola portátil *Nintendo 3DS*, que permite la visión 3D sin necesidad de utilizar gafas. *Sony* por su parte lanzó la *PS Vita*. Estas consolas están marcadas por la usabilidad de internet como eje principal de jugabilidad. Las consolas se convierten en ordenadores y se caracterizan por su posibilidad de multijugador en tiempo real. En 2012, *Nintendo* lanza *Wii U*.

En 2013, *Sony*, saca al mercado la *PlayStation 4*, *PS4* y un año más tarde *Microsoft* hace lo propio con la *XboxOne*. Ambas consolas permiten la interactividad del jugador con otras aplicaciones como *Skype*, la compra de juegos *online* o permitir la navegación por internet.



Ilustración 3: Consolas de Sony hasta la PS4. Fuente: (Alvarez, s.f.)

Con respecto a los PC, algunos de los juegos más conocidos son: *Fallout*, *The Witcher*, *Dark Soul*, *Skyrim*. *World of Warcraft* sigue siendo una referencia para el público y el *League of Legends*, se convierte en uno de los más jugados. Las competiciones de estos videojuegos han ido cobrando más importancia. Los conocidos como los deportes electrónicos o *eSports*, han logrado tener una gran repercusión. Juegos como *DOTA 2*, *League of Legends* o el videojuego de disparos en primera persona *Counter-Strike* son los más populares en este tipo de torneos.

En 2011 se publica *Minecraft*, un juego *indie* – juegos desarrollados por pequeñas empresas que no suelen contar con apoyo financiero de distribuidores y publicistas- de construcción en el que el jugador puede interactuar con el mundo y construir nuevos objetos a partir de otros. Acabó revolucionando el mundo de los videojuegos siendo adaptado para las principales consolas. También cuenta con una sección dedicada a la educación *Minecraft EDU*.

Con la llegada de los *smartphones* y las *tablets*, la industria de los videojuegos llega a más público y permiten disfrutar de juegos simples pero muy adictivos a un bajo precio. Varios de los más conocidos son *Angry Birds*, *Candy Crush*, *Temple Run* o *Fruit Ninja*.

En 2017, *Nintendo* lanza la *Nintendo Switch*, una consola híbrida que se puede usar tanto como consola de sobremesa como portátil. Ese mismo año aparece *Fortnite* desarrollado por la empresa *Epic Games* que a las dos semanas de su publicación, más de 10 millones de jugadores ya lo habían jugado (Makuch, 2017) y en 2018 la cifra alcanzaba ya los 125 millones de jugadores (Statt, 2018). Los ingresos de *Fortnite* durante 2018 se estimaron en cientos de millones de dólares por mes (Thier, 2018).

En 2020, *Sony* lanzó su última consola hasta la fecha, la *PlayStation 5*, *PS5* y *Microsoft*, la *Xbox Series X*. Estas consolas forman parte de la novena generación de consolas.

Con la llegada de la realidad virtual, la industria de los videojuegos y las consolas está en plena revolución, *Oculus Quest*, *Sony PlayStation VR*, *Oculus Rift* o *HTC Vive* son algunas de las gafas de realidad virtual que están logrando tener una gran repercusión en el mercado. El uso de esta tecnología supondrá las tendencias futuras de la industria de los videojuegos, centrándose en el usuario y en su inmersión dentro del videojuego.

III.2.3 Clasificación

La extremada velocidad con la que ha crecido y evolucionado el campo de los videojuegos, ha provocado la creación de clasificaciones dependiendo de varios parámetros. Una clasificación tiene en cuenta la diferenciación entre videojuegos según la edad recomendada para el jugador. Desde otra perspectiva, podemos clasificar los videojuegos atendiendo a su temática o género.

III.2.3.1 Por edades

Muchos padres y madres desconocen los tipos de juegos que existen y lo que sus hijos llegan a consumir. Es primordial para ellos, como responsables de su educación y formación, disponer de información acerca de la idoneidad de los videojuegos dirigidos al público infantil. Para ello existen organismos que asignan una calificación y unas etiquetas a cada videojuego, dependiendo del contenido del mismo. Algunos de los organismos más conocidos para la clasificación por edades de videojuegos –*ratings*– son: **ESRB** en EEUU, **PEGI** en gran parte de los países europeos, **USK** en Alemania, **ACB** en Australia, **CERO** en Japón, **KMRB** en Corea...

Usando la clasificación (Ilustración 4) del organismo **PEGI** (*Pan European Game Information*) los videojuegos se pueden clasificar:

- **PEGI 3.** Nada inapropiado. Juego para mayores de 3 años.
- **PEGI 7.** Elementos de estrés o miedo. Violencia ocasional contra personajes de fantasía, o desnudos en un entorno no sexual. Juego para mayores de 7 años.
- **PEGI 12.** Violencia grafica contra personajes de fantasía, violencia no grafica con humanos / animales, sexualidad moderada o palabrotas suaves. Juego para mayores de 12 años
- **PEGI 16.** Violencia grafica contra humanos / animales no realistas, fuerte contenido sexual, uso drogas ilegales o exaltación del crimen. Juego para mayores de 16 años
- **PEGI 18.** Violencia grafica contra humanos / animales, actos sexuales explícitos. Drogas, racismo, etc. Juego para mayores de 18 años



Ilustración 4. Clasificación PEGI. Fuente: (Fantasía Gamer, 2016)

Estas etiquetas aparecen en la **carátula** de todos los videojuegos, sin excepción (Ilustración 6). Y viene acompañado de unos iconos (Ilustración 5) que muestran aquellos contenidos por los que se caracteriza el juego.



Ilustración 5. Etiquetas PEGI. Fuente: (Fantasía Gamer, 2016)



Ilustración 6. Ejemplo de aplicación del PEGI. Fuente: (García Gigante, s.f.)

III.2.3.2 Por temática o género

Los videojuegos también se pueden clasificar por temáticas o géneros dependiendo de su contenido (GamerDic, s.f.).

III.2.3.2.1 Acción

Los videojuegos de acción requieren que el jugador haga uso de sus reflejos, puntería y habilidad, a menudo en un contexto de combate o de superación de obstáculos y peligros, distinguiendo los siguientes tipos:

- **Lucha:** El jugador se enfrenta en combates uno contra uno contra otros personajes, controlados por la máquina o por otro jugador humano. Ejemplos: *Street Fighter*, *Mortal Kombat*, *Tekken*, *Super Smash Bros...*
- **Beat 'em up.** El jugador se enfrenta a infinidad de enemigos, generalmente en combates cuerpo a cuerpo, mientras se desplaza por un escenario que avanza en una única dirección. Ejemplos: *Double Dragon*, *Final Fight*
- **Hack and Slash.** Parecido al *Beat 'em up* pero el personaje controlado por el jugador lleva una espada, hacha o alguna otra arma de filo contundente para enfrentarse a las oleadas de enemigos. Ejemplos: *Diablo*, *God of War*
- **Arcade.** Jugabilidad simple, repetitiva y de acción rápida. Ejemplos: *Space Invaders*, *Asteroids*, *Pac-Man*

III.2.3.2.2 Disparos (Shooters)

El principal objetivo es disparar y matar enemigos, generalmente con armas de fuego. Entre otros subtipos, aparecen:

- **Primera persona (FPS):** La perspectiva con la que jugamos es desde el punto de vista del propio personaje que controlamos. Normalmente sólo vemos el arma y/o el brazo del personaje. Ejemplos: *Doom, Halo, Call of Duty, Far Cry*
- **Tercera persona (TPS):** La perspectiva con la que jugamos es desde detrás del personaje. Ejemplos: *Uncharted, Fornite, GTA, Tomb Raider*
- **Shoot 'em up:** El jugador controla un único personaje o vehículo, y debe eliminar a multitud de enemigos que aparecen en pantalla de manera simultánea al tiempo que esquivas sus disparos. Ejemplos: *Space Invader, Gradius, Metal Slug*

III.2.3.2.3 Estrategia

El jugador controla a un numeroso grupo de personajes, objetos o datos, haciendo uso de la inteligencia y la planificación, para lograr los objetivos. Aunque la mayoría de estos juegos son fundamentalmente de temática bélica, los hay también de estrategia económica, empresarial o social, entre otros:

- **Estrategia en tiempo real (RTS):** Todos los sucesos y acciones del juego suceden de forma continua. En este tipo de estrategia la agilidad mental y capacidad de reacción es tanto o más importante que la capacidad táctica. Ejemplos: *Age of Empires, Age of Mythology, StarCraft, Command & Conquer*
- **Estrategia por turnos:** El tiempo está pausado y el jugador cuenta con una serie de acciones a realizar durante su turno. Una vez agotado, el turno pasa al enemigo o a otros jugadores rivales. En este tipo de estrategia prima la capacidad táctica y analítica del jugador antes que la agilidad mental o la capacidad de reacción. Ejemplos: *Civilization, Dofus, Heroes of Might and Magic, Total War*
- **4X (Explora, Expándete, Explota y Extermina):** El jugador controla a un imperio o raza, con el objetivo de hacerla evolucionar y crecer lo suficiente para dominar y sobreponerse al resto de contrincantes. Ejemplos: *Sid Meier's Civilization*

III.2.3.2.4 Simulación

Representan situaciones de la vida real de la manera más fiel posible, como por ejemplo:

- **Simulación de vehículos:** Permiten al jugador operar una variedad de vehículos de forma más o menos realista (conducción, vuelo, vuelo de combate, combate en vehículos, ferroviarios). Ejemplos: *Gran Turismo, Need For Speed, Microsoft Flight Simulator, Mario Kart*.
- **Simulación de construcciones.** El programa le proporciona al usuario todas las herramientas para construir un proyecto, en el cual se tienen en cuenta desde

gastos de construcción y mantenimiento, hasta una línea de tiempo, física y clima que afecta todas las decisiones que tome (ciudades, imperios, empresarial, granja). Ejemplos: *Theme Park, Sim City, Civilization, Total War, Minecraft*

- **Simulación de vida:** El jugador controla una o más formas de vida artificial, ya sea de seres humanos o de animales. Ejemplos: *Los Sims, Nintendogs, Pou, Animal Crossing*

III.2.3.2.5 Deporte

Simulan deportes del mundo real, golf, tenis, fútbol, baloncesto, hockey... Ejemplos: *FIFA, PES, NBA 2K, Virtua Tennis*

III.2.3.2.6 Carreras

Sitúan al jugador en un recorrido en el que debe llegar a una meta antes que sus contrincantes o dentro de un tiempo límite. Habitualmente el jugador maneja un vehículo a motor como un coche.

- **Arcade:** Nivel de fidelidad respecto al mundo real bajo. Ejemplos: *Mario Kart, Out Run, Crash Team Racing, F-Zero, Midnight Club, Sonic & Sega All-Stars Racing*
- **Simuladores:** Nivel de fidelidad respecto al mundo real bajo. Ejemplos: *Gran Turismo, Project Gotham Racing, Forza Motorsport, Need for Speed.*

III.2.3.2.7 Aventura

El protagonista debe avanzar en la trama interactuando con diversos personajes y objetos:

- **Aventura gráfica:** El jugador debe conversar con personajes e interactuar con elementos del entorno para resolver los puzzles que propone la trama del juego a través de diversas situaciones.
 - **Point and click:** Se usa exclusivamente el ratón del PC para interactuar con el juego. Ejemplo: *Monkey Island*
 - **Walking Simulator:** Gran enfoque en la narración, fundamentalmente lineal, y una interactividad mínima, limitada a la búsqueda de objetos que permiten proseguir el relato o la resolución de algunos puzzles. Ejemplos: *Dear Esther, Gone Home o Firewatch*
- **Survival horror:** Se caracterizan por su ambientación de terror y porque el jugador siente en todo momento la sensación de ser inferior al enemigo. Al carecer de las armas, munición o medios necesarios para acabar con ellos, el jugador se ve obligado en muchos casos a evadir o escapar de los enemigos. También se caracterizan por contar con gran cantidad de puzzles o acertijos que resolver para avanzar. Ejemplos: *The Last of Us, Resident Evil*

- **Sigilo:** El objetivo es emplear el sigilo para avanzar, evitando en la medida de lo posible ser vistos por el enemigo. Ejemplos: *Assassin's Creed, Hitman, Thief*.
- **Metroidvania:** Exploración de los escenarios, dando cierta libertad para avanzar por ellos y controlando el acceso a ciertas zonas mediante la necesaria adquisición de nuevas habilidades u objetos, lo que obliga a menudo a hacer *backtracking*, es decir "retroceder" en los escenarios. Ejemplos: *Ori, Hollow Knight, Castlevania*

III.2.3.2.8 Rol

Se caracteriza por el desarrollo de las habilidades y características de un personaje a través de la experiencia que obtenemos completando las misiones propuestas a través del juego, tomando ciertas decisiones o realizando ciertas acciones, personalizándolo al gusto del jugador. Este desarrollo está vinculado a una escala numérica. Ejemplos: *World of Warcraft, Final Fantasy*

III.2.3.2.9 Agilidad mental/Puzle/Lógica

Ponen a prueba la capacidad del jugador para la resolución de problemas, que pueden ser de índole matemático, espacial o lógico. Ejemplos: *Brain Age, Brain Academy, Tetris*

III.2.3.2.10 Educativos

Son aquellos dirigidos a aportar unos conocimientos o habilidades al jugador durante el transcurso del mismo. Se combina así formación con entretenimiento. Ejemplos: *Fine Artist, El autobús mágico*.

III.2.4 Perfiles laborales.

La tendencia de ocio y tiempo libre en estos años ha encontrado en el videojuego uno de los grandes atractivos y más afinidad por partes de los sectores en general, más jóvenes de la población, sin obviar otros sectores.

En términos económicos, la industria del videojuego ya obtiene mayores ingresos que la industria del cine y la música juntos. Actualmente, los videojuegos representan la primera opción de ocio audiovisual en España y lo que ha supuesto una facturación de 713 millones de euros anuales, según el Libro Blanco del Videojuego Español de la DEV (Desarrollo Español de Videojuegos, 2020). Dichos datos hacen de esta industria una fuente de empleo en expansión, y que inspiran los deseos laborales de muchos jóvenes que en la actualidad cursan etapas de ESO, Bachillerato o ciclos formativos.

La configuración actual de la formación profesional, ofrece varias titulaciones que pueden llevar a la especialización en esta industria, como pueden ser los títulos de: Técnico superior en Administración de Sistemas Informáticos en Red, Técnico Superior en Desarrollo de Aplicaciones Multiplataforma o Técnico Superior en Desarrollo de

Aplicaciones Web (Todo FP, s.f.). Dada la creciente demanda en este campo, ya mencionada anteriormente, el Ministerio de Educación tiene previsto la implantación en centros públicos de un curso de Especialización en Desarrollo de videojuegos y realidad virtual. Se espera que el proyecto de Real Decreto que establece el currículo de este curso de especialización vea la luz a partir del curso 2021/2022 (Ministerio de Educación, 2021).

El mundo del desarrollo de videojuegos requiere la participación de profesionales de la más diversa índole, como pueden ser Diseñadores, Programadores, Artistas... El buen trabajo cooperativo de dichos profesionales hará que el producto que se elabore, sea un producto de éxito.

III.2.4.1 Diseñador.

Perfil encargado de crear las ideas, las mecánicas y las reglas de las que se quieren llevar al producto final. El diseñador debe tener habilidades y conocimientos en diversos campos tales como arquitectura, comunicación, historia, matemáticas, música... (Schell, 2019), siendo también pilares fundamentales el saber escuchar al equipo y el comunicar y transmitir con eficacia sus ideas. Es el encargado de redactar el "GDD"- *Game Design Document*- que es la documentación que contiene todo lo relacionado con el juego y que servirá como referencia a programadores y artistas, y en general al resto de los componentes del equipo de trabajo.

Podemos encontrar varios tipos de diseñador:

- **Diseñador de niveles.** Crea los mapas del nivel, así como el primer prototipo del juego (*Grayboxing*, Ilustración 7), en los que ya se pueden incluir algunos elementos en su fase inicial para estudiar viabilidades.

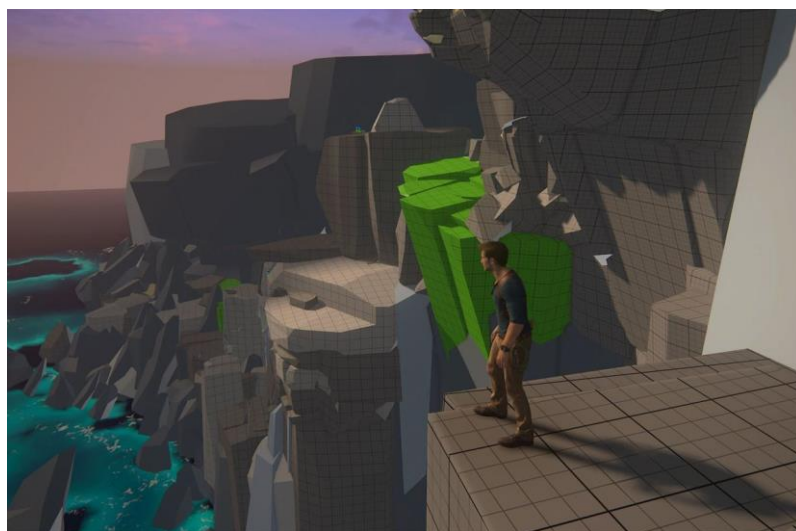


Ilustración 7: Graybox de Uncharted, primer prototipo. Fuente: (Schreier, s.f.)

- **Diseñador de sistema.** Desarrolla las relaciones entre los distintos elementos del juego.
- **Diseñador de mecánicas.** Encargado de crear las reglas y las mecánicas del juego.
- **Scripters.** Desarrolla el código para que el producto sea funcional en versiones preliminares.
- **Diseñador de combate.** En casos de productos en los que interrelacionen personajes entre sí, es el encargado de diseñar la experiencia de movimientos de los mismos.
- **Diseñador narrativo.** Se encarga de que la jugabilidad vaya acompañada de la historia. Crea los diálogos de los personajes y las secuencias cinemáticas.
- **Diseñador de UX y UI:** Se encarga de analizar la experiencia de usuario e interfaz de usuario utilizando especialmente la psicología de los usuarios finales del producto.

III.2.4.2 Programador.

Perfil encargado de utilizar los lenguajes de programación. Escribe el código que necesita el producto para su desarrollo, basándose en la guía que proporciona el Diseñador. Es el encargado de dibujar gráficos, textos que se ven en pantalla...

Podemos encontrar varios tipos de programador:

- **Programador de IA.** Se encarga de programar la inteligencia artificial de los "NPC", *Non Playable Character*, objetos y personajes no protagonizados por el jugador.
- **Programador de Físicas:** Encargado de la programación de la simulación de las leyes de la física en los gráficos.
- **Programador de Herramientas:** Se encarga de crear aplicaciones externas al producto, que pueden facilitar su desarrollo.
- **Programador de UI.** Se encarga de la programación de la interfaz de usuario.
- **Programador de Multijugador.** Ayuda al uso compartido de la aplicación.
- **Programador de Online.** Ayuda al uso compartido de la aplicación *online*.

III.2.4.3 Artista.

Perfil encargado de plasmar la idea del departamento de diseño de una manera visual y atractiva.

Podemos encontrar varios tipos de artista:

- **Artista Conceptual 2D:** Realiza los bocetos iniciales del producto para que el equipo tenga una idea preliminar del desarrollo del mismo: entornos y

ambientación, personajes, referencias de *VFX*- efectos visuales-, cinemáticas, menús...

- **Artista Modelador 3D.** Transforma el boceto inicial 2D a objetos 3D que serán incluidos dentro del juego: Personajes, objetos de decoración, entornos...
- **Artista de textura:** Encargado de elaborar superficies en modelos 3D o 2D (texturizar).
- **Artista *Rigger*.** Es el encargado de construir el esqueleto de los personajes para su posterior animación y que determinan la forma que tiene dicho personaje así como su movimiento.
- **Artista Animador.** Realiza las cinemáticas y los movimientos de los personajes.
- **Artista de *VFX*.** Realiza los efectos visuales: fuego, viento, nieve...
- **Artista de *UI*.** Encargado de realizar y diseñar la interfaz de usuario, así como la iconografía empleada y el “HUD”- *Head Up Display*- información visible por pantalla en todo momento.

III.2.4.4Productor.

Perfil encargado de supervisar a todo el equipo de desarrollo. Gestiona y organiza las etapas de desarrollo y entre sus responsabilidades se encuentran: el contratar y construir equipos, planificar, resolver disputas, redactar contratos. Suele ser la cara visible del equipo y el encargado de las relaciones externas al equipo de desarrollo.

III.2.4.5Marketing.

Perfil encargado de la promoción del producto, encargándose también de la visibilidad posterior del producto a través de revistas, páginas webs, televisión, redes sociales... Es el encargado de vender el juego a los futuros jugadores.

III.2.4.6Sonido.

Perfil encargado de la creación de la música y de los sonidos del producto. El sonido consigue ambientar el juego intentando conseguir que la experiencia del usuario sea gratificante. Sonoriza todo tipo de animaciones, *foliage*- efectos sonoros-, grabación de voces, banda sonora y en general diseña todos los elementos acústicos que necesita el producto...

III.2.4.7Tester o QA.

Perfil encargado de chequear el producto, normalmente con el objetivo de encontrar fallos o *bugs* en el videojuego. Cada *bug* que encuentre debe informar al equipo de

desarrollo para que se solucione y con posterioridad debe comprobar si los fallos encontrados se han corregido.

III.2.5 Fases de desarrollo de un videojuego

El objetivo que el equipo tiene marcado ha de ser que el producto final se desarrolle con éxito y deberá contar con una serie de fases desde su concepción hasta su distribución al usuario final. No todas las fases deberán desarrollarse de manera obligatoria ya que podrá depender de la organización del propio equipo de desarrollo, del tipo de videojuego que se quiere desarrollar o incluso del presupuesto asignado para el proyecto.

Las fases siguen la siguiente estructura: Primera fase, la concepción de la idea del juego así como el análisis de riesgos que puede conllevar la elaboración del proyecto y la creación de la primera documentación. La segunda fase, preproducción, engloba el periodo de pruebas iniciales para analizar la viabilidad del proyecto, así como la planificación y la ampliación de la documentación creada en la fase anterior. La tercera fase, la producción, es la etapa en el que el videojuego se desarrolla por completo, desde los primeros prototipos hasta un producto funcional. Las fases de finalización, parche y archivado, son las últimas que pueden llegar a realizarse en el proceso de desarrollo, en el que el producto alcanza su óptima calidad, listo para ser distribuido entre los distintos usuarios y archivar los datos necesarios para el caso en el que el proyecto se pudiera ampliar.

III.2.5.1 Concepción

III.2.5.1.1 Brainstorming

Será el primer acercamiento al desarrollo de un videojuego, etapa en la que se buscará una idea amplia de lo que quiere ser el videojuego. Se trata de una tormenta de ideas colaborativa entre todos los miembros del equipo (o al menos del departamento de diseño, dependiendo del tamaño del equipo). Tras el descarte de algunas ideas y modificación otras, se obtiene un concepto inicial del juego en el que se puede definir el o los elementos más atractivos que puede tener el videojuego (*el gancho*), el género, la plataforma y el modelo de negocio.

III.2.5.1.2 Análisis de riesgos

Tomando como referencia el concepto inicial, se deben analizar juegos similares que haya en el mercado y hacer una comparativa con la idea desarrollada para el videojuego que el equipo pretende desarrollar. Se realiza un grafo **DAFO** (Ilustración 7 Ilustración 8) en el que se analizan las fortalezas, debilidades, oportunidades y amenazas.



Ilustración 8. Diagrama DAFO. Fuente: (Gestión de proveedores, s.f.)

III.2.5.1.3 Documentación inicial

El departamento de diseño se encarga de realizar dos documentos con el contenido de las ideas propuestas en la primera fase:

- **One Pager:** Es un documento de una sola página en el que se resume el juego. Debe contener: Título provisional, género, plataformas, edad objetivo de los jugadores, un resumen de la historia, las mecánicas, el gancho o elementos diferenciadores que hace que el juego destaque y juegos similares desarrollados por la competencia.
- **Ten Pager:** Es un documento de diez páginas que expande el contenido del *One Pager*. El objetivo es que los lectores entiendan los conceptos básicos del juego.

III.2.5.1.4 Pitch Elevator

Breve descripción del producto que se quiere vender, el receptor debe entender la idea en un corto periodo de tiempo (*pitch elevator*: el tiempo que tarda un ascensor en subir a una planta determinada). Hay que incluir todos los elementos diferenciadores para que la idea se muestre atractiva al receptor.

III.2.5.2 Preproducción

III.2.5.2.1 Proof of Concept

El *Proof of Concept* o prototipo es el desarrollo de una idea dentro del motor de juegos para comprobar que puede funcionar, no es necesario que tenga elementos artísticos complejos pero sí podría tener elementos simples que puedan ser eliminados una vez el arte final del juego lo sustituya (*placeholder*) y debe tener un tamaño pequeño. No siempre es necesario su realización y varía según el tipo de proyecto o el equipo de desarrollo.

III.2.5.2.2 Vertical Slice

Según un post en la web del videojuego *Hellblade* (Ninja Theory, s.f.), un *vertical slice* es una sección del juego con una duración de juego de entre 10 y 30 minutos, que es representativa del aspecto, el sonido y la jugabilidad final del juego.

Al finalizar el *vertical slice* se puede analizar, en una etapa de desarrollo inicial, la viabilidad del proyecto.

III.2.5.2.3 Pipeline

Definir las herramientas básicas que se van a utilizar durante el desarrollo del proyecto. Motores de videojuegos, *software* específico, gestión de las copias de seguridad del proyecto y control de versiones.

III.2.5.2.4 Planificación

Listado priorizado de todas las tareas a realizar y los miembros del equipo que las van a desarrollar.

III.2.5.2.5 GDD

El *Game Design Document*, **GDD**, o comúnmente denominado “la biblia del juego” es el archivo que contiene toda la información referente al proyecto, los elementos que tiene y cómo funcionan cada uno de ellos. Servirá de guía para todos los miembros del equipo durante el desarrollo del juego.

La creación de este documento es tarea del departamento de diseño con la colaboración del resto de departamentos.

III.2.5.3 Producción

Siguiendo y respetando el GDD, se realizará el desarrollo completo del videojuego dividiéndolo en diferentes etapas:

- **Funcional:** Se debe poder jugar a la totalidad del juego con una calidad visual y de detalle mínima. Se utilizan *placeholders* o elementos temporales para que el desarrollo sea lo más rápido posible.
- **Par Standard.** El juego debe tener una calidad mínima de mercado. Se incluyen los elementos de arte finales.
- **Polish.** La calidad visual y funcional es cercana a la óptima.

Durante las etapas anteriores, hay otros elementos que hay que tener en cuenta.

- **Localización:** Traducir a otros idiomas los elementos del juego, tanto textos como voces si las tuviese.
- **Testeo a personas externas.** Obtener información de jugadores que prueben versiones previas del juego. Posteriormente se debe analizar los resultados para determinar: qué elementos han gustado, qué elementos no se han entendido correctamente... para tomarlo como referencia para posibles modificaciones o mejoras del juego.
- **Demos:** Demostraciones del videojuego en versiones avanzadas del proyecto. Se deberán obtener comentarios que puedan ayudar a mejorar el proyecto (*feedback*).
- **Marketing.** Captura de contenidos, creación de *teasers* o *trailers*, notas de prensa para promocionar y dar visibilidad al juego a través de revistas, páginas webs, televisión, redes sociales, eventos (Ilustración 9)...



Ilustración 9. Feria del videojuego Madrid Games Week. Fuente: Madrid Games Week

III.2.5.4 Finalización

III.2.5.4.1 Polish

Es esencial en todo desarrollo y diferencia a un gran producto y uno mediocre. En esta fase la calidad visual y funcional es cercana a la óptima. Y se incluyen detalles no

esenciales pero pueden hacer lograr que la atracción hacia el juego por parte de un jugador sea máxima.

III.2.5.4.2 Testeo o QA.

Búsqueda de fallos o *bugs* del juego. Cada bug debe ser informado para que el equipo de desarrollo pueda solucionarlo. También se encargan de validar aquellos *bugs* que se han corregido. Hay 3 tipos de *bugs*:

- **Tipo A.** Rompe la continuidad del juego. No se puede continuar jugando. Son los más graves y hay que solucionarlos inmediatamente. Ej. Obstáculo que no te deja continuar, cierre repentino de la aplicación...
- **Tipo B.** Severos. No rompen la continuidad del juego pero no favorecen a la experiencia del jugador. Ej. IA enemigos no funcionales.
- **Tipo C.** Menor. Ej. Detalles visuales defectuosos, objetos flotando...

III.2.5.4.3 Submission

Cuando el juego está preparado para salir al mercado se prepara una versión llamada "**Golden Candidate**" o "**Golden Master**" (TechTerms, s.f.) que será la que acabará en las versiones físicas del juego o en las versiones digitales distribuidas en tiendas como *Steam*, *Epic Games* etc. Al pasar el juego a una versión "**Golden Candidate**" no se puede modificar el contenido, ni para añadir contenido ni para solucionar fallos. Estas modificaciones, tanto para añadir contenido o para solucionar fallos, se podrán hacer con una versión posterior al lanzamiento llamada parche, que se tratará en la siguiente sección.

La versión candidata en formato físico de *DVD* debe estar acabada mínimo 6 semanas antes de la fecha de lanzamiento del juego puesto que se tienen que fabricar dichos soportes (*DVDs*), realizar las copias y distribuirlas a los distintos puntos de venta.

III.2.5.5Parche

Cuando el juego está en el mercado, es muy posible que haya jugadores que encuentren *bugs* o incluso es posible que se quiera añadir algún contenido. Los desarrolladores deberán realizar una actualización del juego que se ocupe de la mejora en los posibles errores detectados. En dicha actualización los errores deben estar corregidos. A esa actualización se la conoce como parche y dependiendo del tipo de distribución que se ha seguido pueden ser de varios tipos.

- **Parche "día 1".** La versión se cierra 6 semanas antes del lanzamiento, y a partir de ese momento, el equipo sigue trabajando en encontrar fallos que puedan aparecer. A través de internet, esta actualización se distribuye al usuario final.

- **Parches por *bugs* posteriores o inclusión de nuevos contenidos.** El equipo de desarrollo deberá seguir corrigiendo posibles *bugs* y añadiendo posibles contenidos que puedan mejorar el producto.

III.2.5.6 Archivado

Es la última etapa del desarrollo en el que el equipo guarda y almacena los archivos relacionados con el proyecto por si en algún momento futuro, se necesitará volver a consultar dicha información. Posibles necesidades futuras.

- Hacer una continuación o secuela
- Hacer un *porte*, es decir, modificar el código para poder distribuir el juego en otras plataformas o consolas.
- Hacer un *DLC*, un contenido extra descargable en formato digital.

III.2.6 Características específicas de los dispositivos móviles

Se hace necesario conocer con cierta profundidad el funcionamiento de aquellos dispositivos a los que va dirigido el proyecto que se desarrolla. En este sentido es importante conocer tanto el *Hardware* como el *Software* de los dispositivos móviles. Se debe conocer qué características los hacen diferentes frente a otras plataformas. Su rendimiento creciente y su constante adaptación a distintas plataformas gracias a su pequeño formato y a sus múltiples sensores, hacen que sea una de las plataformas más populares para los jugadores (Burón García, Manzano Vázquez, & Paredes Velasco, 2020).

III.2.7 Formatos de hardware

Aunque actualmente existen en el mercado dispositivos móviles de diversos tamaños, procesadores, procesamiento gráfico, memoria, podemos agruparlos en los siguientes grupos (Burón García, Manzano Vázquez, & Paredes Velasco, 2020):

- **Smartphones:** Son los principales dispositivos móviles por su gran popularidad. No solo sirven para la comunicación por voz, sino que también actúan como un ordenador portátil completo de bolsillo, con el que poder navegar, chatear, jugar, visualizar archivos multimedia etc. Entre todos los distintos fabricantes, los cinco primeros que copan un 75% del mercado son: *Samsung* (32%), *Apple* (23%), *Huawei* (10%), *Xiaomi* (8%) y *Oppo* (4,5%)
- **Tablets:** Muy similares a los *smartphones* en características técnicas, siendo la principal diferencia el tamaño y la resolución de pantalla del dispositivo. Normalmente con un tamaño entre 7 y 12" y con resoluciones por encima de

FullHD. Los cinco primeros fabricantes son: *Apple* (73%), *Samsung* (15%), *Amazon* (3%), *Asus* (1%) y *Huawei* (1%).

- **Phablets:** El nombre se construye de la combinación entre *phone* y *tablet*. Se definen como *smartphones* de gran pantalla en los que prevalece la capacidad multimedia frente a su movilidad. Cada vez, los *smartphones* se están fabricando con tamaños mayores, por lo que se puede decir que el grupo de *phablets* se está fusionando con el de *smartphones*.
- **Consolas Portátiles:** Son dispositivos diseñados específicamente para jugar, suelen tener una mayor capacidad gráfica y unos periféricos específicos. Sus características en sensores, cámaras y comunicación suelen estar más limitadas que en los *smartphones*. Hasta hace poco, los principales fabricantes eran *Nintendo* (*Game Boy Color*, *Game Boy Advance*, *Nintendo DS*, *Nintendo 3DS*, *Nintendo Switch*) y *Sony* (*PSP*, *PSP Go*, *PS Vita*). En la actualidad queda en el mercado mayoritariamente la consola *Nintendo Switch* (Ilustración 10), pensada tanto para sobremesa como portátil.



Ilustración 10. Consola portátil Nintendo Switch. Fuente: (Nintendo, s.f.)

III.2.8 Sistemas Operativos

Además del *hardware*, otro de los elementos importantes en un dispositivo móvil es el software que los gestiona. Los principales sistemas operativos para plataformas móviles son los siguientes:

- **Android:** desarrollado por *Google*, basado en el *kernel* de Linux. El lenguaje de programación utilizado es *Java*, aunque también puede desarrollarse en *Kotlin*, C y C++. La última versión disponible es *Android 11* lanzada en septiembre de 2020 y en agosto de 2021 está previsto el lanzamiento de *Android 12*. Cuenta con más del 75% de la cuota del mercado y con más de 2.000 millones de

usuarios activos mensuales. Salvo *Apple* y *Nintendo*, prácticamente el resto de fabricantes usa *Android*, por lo que hay una amplia variedad de dispositivos con diversas características y precios. Todas las aplicaciones y juegos que se distribuyen en *Android*, casi 3 millones, están disponibles en su tienda *Google Play*. Existen otras tiendas para la distribución de aplicaciones desarrolladas con *Android*, gracias a la compatibilidad del mismo, siendo la más destacada *Amazon AppStore*.

- **IOS:** desarrollado por Apple para sus *smartphones iPhone* y *tablets iPad*, utiliza el *kernel XNU*. Aunque en *smartphones* solo tenga una cuota de mercado del 23% frente a su gran competidor *Android*, en el mercado de *tablets* lo domina con un 73%. El lenguaje de programación utilizado es *Objective-C*. La última versión disponible es *iOS 14*, lanzada en septiembre de 2020. La tienda de distribución de aplicaciones es *Apple Store* que cuenta con un enorme control de sus aplicaciones por parte de *Apple* para garantizar la calidad. Dispone de casi 2 millones de aplicaciones y genera muchos más ingresos que *Google Play*.
- **Windows:** Aunque *Microsoft* dejó de desarrollar *Windows 10 Mobile* en 2017, en las *tablets* que usan este sistema operativo, es frecuente controlarlas con *Windows 10* y eso las hace ser dispositivos más cercanos a ordenadores.
- **Otros:** Aunque los anteriores sistemas son los más populares y controlan prácticamente el 99% del mercado, existen otros sistemas operativos tanto de código abierto como cerrado.

III.3 Desarrollo de juegos móviles

La sección fundamental y principal que se dedica al desarrollo de un videojuego para dispositivos móviles contempla especialmente dos ámbitos importantes, los lenguajes y los motores de desarrollo y las aplicaciones que implementan estos lenguajes para el desarrollo en dispositivos móviles. Este capítulo abarca desde la definición de motor hasta su utilización como medio de programación de videojuegos destinados a plataformas móviles.

III.3.1 Lenguajes y motores de desarrollo

Actualmente, el desarrollo de un videojuego puede llegar a ser algo muy complejo y puede llegar a tener muchas ramificaciones. En el pasado, los videojuegos se desarrollaban desde cero, teniendo que volver a escribir el mismo código una y otra vez cuando se empezaba un proyecto. Con el paso del tiempo y el avance de la tecnología, con el auge de los videojuegos, en la década de 1980, las compañías empezaron a desarrollar sus propios motores, es decir, entornos de desarrollo que permitían reutilizar

gran parte del código. Sin embargo el uso de motores de terceros no se popularizó hasta la década de 1990, cuando aparecieron los gráficos 3D (González Soto, 2018).

En la actualidad, la mayoría de videojuegos se desarrollan utilizando estas herramientas ya que entre otras cosas, permite abaratar costos y facilitar el desarrollo en múltiples plataformas.

III.3.1.1 ¿Qué son los motores de videojuegos?

Los **motores de juegos** son herramientas de desarrollo que facilitan la creación de los mismos. Estos proporcionan funcionalidades comunes para el desarrollo de un juego, sin tener que duplicar el esfuerzo que supone reescribir todas las funcionalidades desde cero. Entre las características funcionales que tiene un motor podemos encontrar la simulación de físicas, la detección de colisiones, el manejo de sonido, el *scripting*, la inteligencia artificial, las funcionalidades de red, el manejo de la memoria, el soporte para localización- término utilizado para hacer referencia a las traducciones que puede tener un videojuego-, el soporte para realizar cinemáticas... Todo lo anterior y más, permite que el equipo de desarrollo pueda centrarse en los detalles respectivos únicamente del juego que quieren crear.

III.3.1.2 Motores más populares

Existen muchos motores de videojuegos. Si bien a continuación se relacionan algunos de ellos, no obstante particularizaremos en los dos motores más importantes que se usan en la actualidad, haciendo hincapié en el lenguaje de scripting que utilizan.

- RPG Maker - Ruby/Javascript
- GameMaker Studio - GameMaker Language (GML)
- Godot Engine - C# / Python / Go/ Rust
- CryEngine (Crytek) - Lua / C#
- Source Engine - Lua
- Frostbite (EA) - C++ / C#
- RAGE Engine (Rockstar Games) - C# / Python

III.3.1.2.1 Stencyl / Scratch

Aunque no es de los más importantes de la industria, mención aparte merece el motor **Stencyl**, que fue desarrollado por el Instituto de Tecnología de *Massachusetts* utilizando un lenguaje de programación visual llamado *Scratch*. Es un método para enseñar a los estudiantes cómo funciona la programación sin requerir conocimientos técnicos. Al ser un lenguaje de programación visual, la programación se realiza por bloques

asemejándose a un puzle en el que el alumno debe encontrar las piezas correctas para crear acciones del software.

Debe destacarse que los dos motores más utilizados en la industria de los videojuegos son: **Unity** y **Unreal Engine**

III.3.1.2.2 Unity

Cuenta con más de 5 millones de desarrolladores registrados en la plataforma haciendo que su comunidad sea de las más grandes. Está enfocado tanto para juegos 2D como 3D y el lenguaje de programación utilizado es **C#** aunque también se puede programar visualmente usando **Bolt**. Tiene soporte para hacer juegos en 27 plataformas incluyendo las últimas consolas como *Nintendo Switch*. Existen muchos tutoriales y cursos en la red ya que la herramienta es gratuita. Algunos de los juegos notables que se han desarrollado con Unity son: *Cuphead* (2017), *Hollow Knight* (2017) o *Monument Valley* (2017) (Unity, n.d.).



Ilustración 11: Logo Unity. Fuente: Unity

III.3.1.2.3 Unreal Engine

Enfocado para su uso en el que los gráficos y la iluminación cobren protagonismo, es por lo que está considerado como uno de los motores con mejores gráficos visuales en el mercado. Debido a eso, es un motor muy popular entre los *estudios AAA*, es decir, estudios que realizan mega producciones de juegos con alto presupuesto (Industria Animación, 2020).



Ilustración 12: Logo Unreal Engine 4. Fuente: Epic Games

Muchos de los juegos más exitosos han utilizado este motor, lo que lo ha hecho muy popular, dando lugar a numerosos tutoriales y cursos *online*. Tiene uno de los mejores soportes y constantemente están actualizando y mejorando las herramientas que se incluyen en el motor.

Está enfocado a juegos 3D, aunque también se pueden hacer juegos en 2D. Y el lenguaje de programación utilizado es **C++** y un lenguaje de programación visual llamado **Blueprints**. Los juegos se pueden desarrollar para todas las plataformas que hay en el mercado y el costo de su utilización es gratuito a menos que el juego desarrollado con este motor supere en el mercado un millón de euros en ingresos, costo que pasaría a ser de un 5% de dichos ingresos (Hobby Consolas, 2020).

Algunos de los juegos notables que se han desarrollado con *Unreal Engine* son: *Batman Arkham Knight* (2015), *Street Fighter V* (2016) y *Fortnite* (2017).



Ilustración 13. Juegos desarrollados con Unreal Engine (de izquierda a derecha: *Batman Arkham Knight*, *Street Fighter V*, *Fortnite*).
Fuente: (González Soto, 2018)

III.3.1.3 Programación visual: Bolt y Blueprints

Los motores *Unreal Engine* y *Unity* comparten un lenguaje propio de **programación visual**. Este tipo de programación es muy útil para aquellos desarrolladores que no tienen conocimientos técnicos profundos de programación y esto ayuda a que todos los miembros del equipo puedan programar al menos una base inicial del prototipo mediante gráficos visuales, con nodos que se arrastran y se sueltan, en lugar de escribir código de forma tradicional (Unity, n.d.). Artistas y diseñadores pueden programar sin necesidad de la ayuda de los programadores agilizando la creación de prototipos rápidos con los que poder probar las ideas, hacer cambios y poder iterar sobre ellas.

III.3.1.3.1 Bolt

Es una herramienta de programación visual para *Unity*. Permite a los usuarios crear lógicas para juegos sin necesidad de escribir ninguna línea de código. *Bolt* tiene gráficos visuales basados en nodos (Ilustración 14), los cuales los pueden utilizar tanto programadores como personas sin conocimientos previos de programación, para diseñar la lógica final o crear prototipos con rapidez. *Bolt* también incluye una *API* para que los programadores puedan realizar tareas más avanzadas o para crear nodos

personalizados. Esta herramienta está disponible gratuitamente para todos los usuarios de *Unity* desde mayo de 2020 (Unity, 2020).

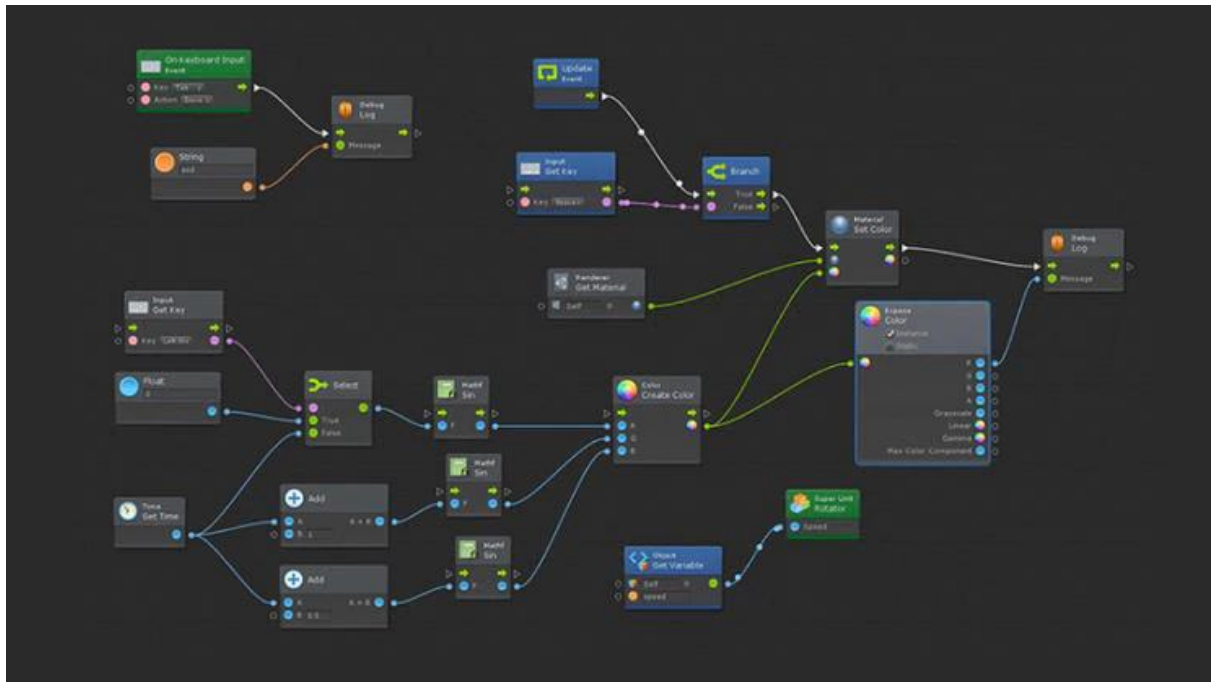


Ilustración 14. Bolt. Fuente: Unity.com

III.3.1.3.2 Blueprints

El sistema de *Blueprints* permite a los desarrolladores crear juegos de forma sencilla y visual, añadiendo funcionalidades a las clases de juego existentes. Este método no sustituye a la programación *C++*, sino que es un complemento, ya que los dos sistemas están integrados entre sí. Que sean complementarios, no quiere decir que no se puedan desarrollar juegos exclusivamente realizados en *C++* o en *Blueprints*.

Los *Blueprints* están integrados en el editor de *Unreal Engine* y se organizan en nodos conectados entre sí (Ilustración 15). Tienen multitud de opciones como por ejemplo (BaboonLab, s.f.):

- Crear, implementar o modificar cualquier elemento.
- Crear mallas, materiales y personajes
- Interacción con menús
- Creación de cámaras y perspectivas, items, partículas como fuego o lluvia
- Interacción con el entorno

Además, los programadores disponen de herramientas en *C++* para la creación de nuevos nodos específicos personalizados que se pueden utilizar en el sistema de *Blueprints*.

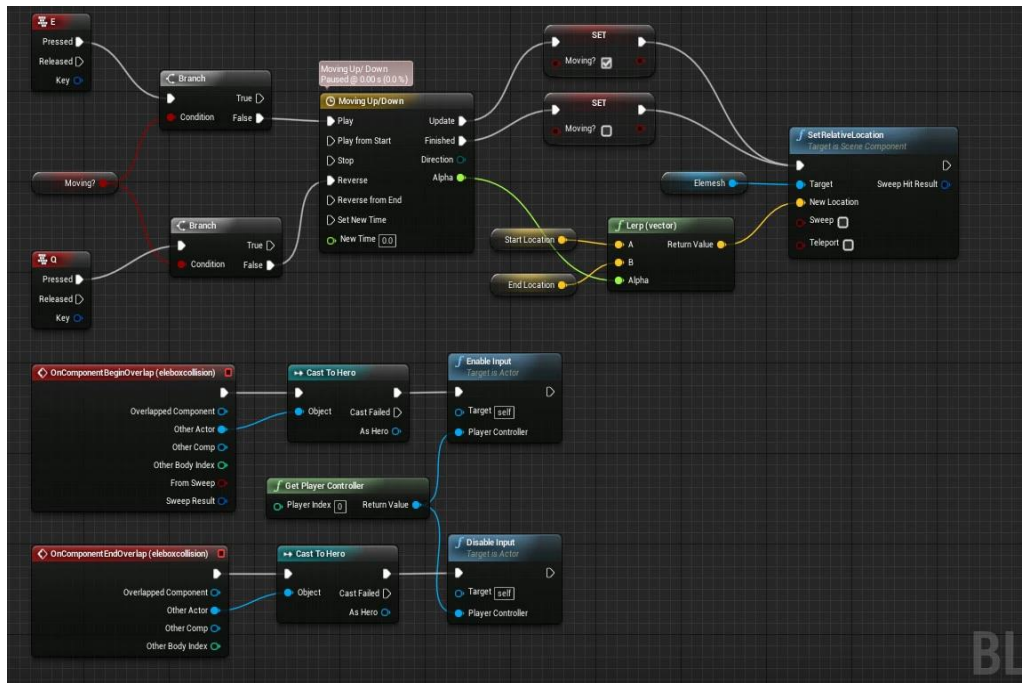


Ilustración 15. Blueprints. Fuente: Elaboración propia

III.3.2 Desarrollo para dispositivos móviles

Serán el motor *Unreal Engine* y el sistema operativo *Android*, las referencias que serán utilizadas para el desarrollo del TFM a partir de este punto.

Aunque la programación dirigida al desarrollo de juegos móviles es muy similar al desarrollo de cualquier juego dirigido a otras plataformas, existen diferencias a tener en cuenta, sobre todo a la hora de empezar el proyecto, la creación de recursos para el juego y la distribución o creación del archivo ejecutable.

III.3.2.1 Inicio del proyecto

III.3.2.1.1 Android Studio

Instalar *Android* en *Unreal Engine* es un requisito indispensable para poder desarrollar juegos en teléfonos móviles o *tablets* que tengan dicho sistema operativo (Salinas, 2020).

En primer lugar hay que comprobar que se tienen instalados los archivos de la plataforma *Android* en la versión del motor de *Unreal Engine* instalada. Si no está instalada hay que instalar el paquete correspondiente (Ilustración 16) (Ilustración 17).

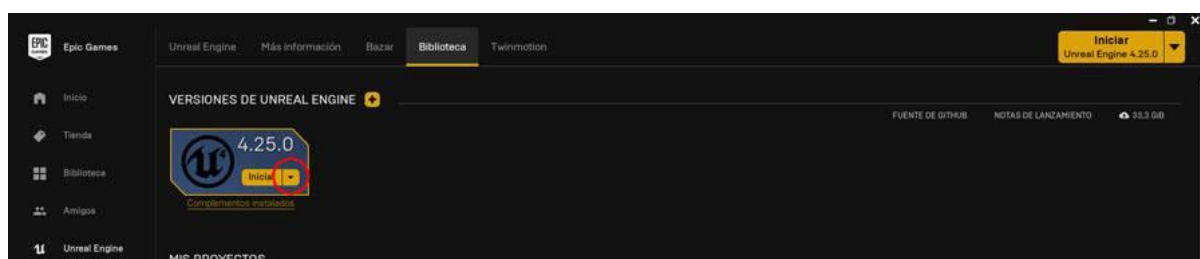


Ilustración 16. Comprobación de instalación Android (1). Fuente: (Salinas, 2020)

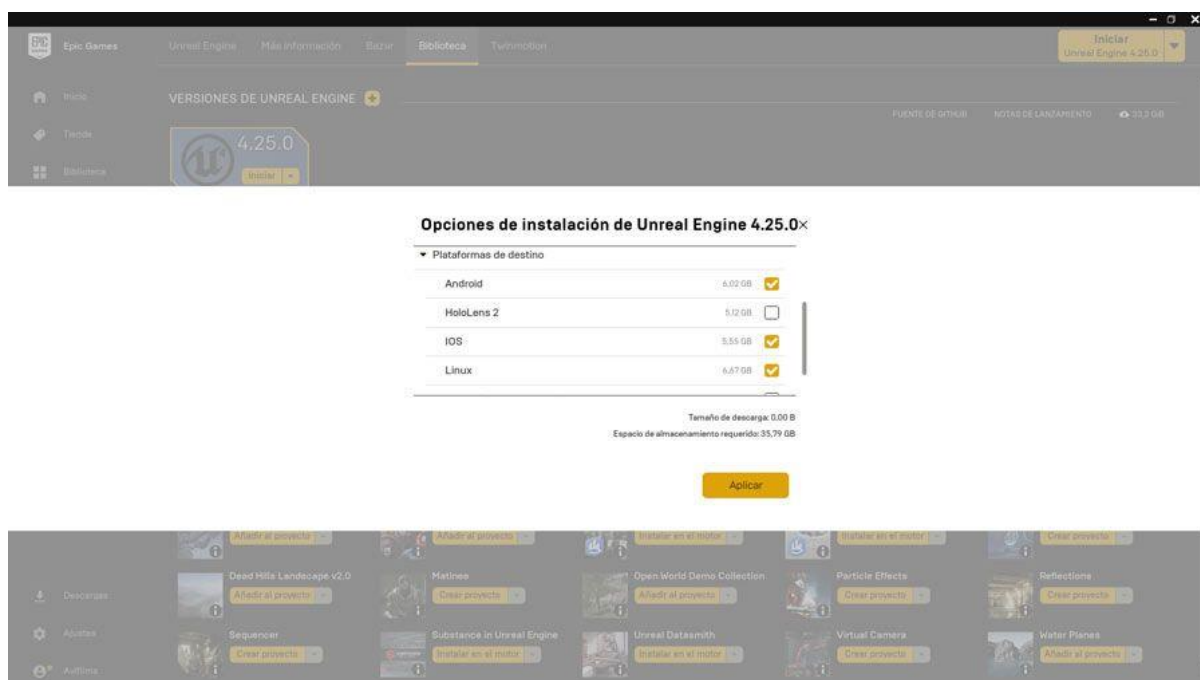


Ilustración 17. Comprobación de instalación Android (2). Fuente: (Salinas, 2020)

A partir de la versión 4.25 de *Unreal Engine*, se requiere la revisión 21 de *Android Native Development Kit (NDK r21)*. Hay que ir a la web de *Android Studio* (Android Studio, s.f.) y descargar e instalar la versión 3.5.3.

Una vez descargado, abrir la carpeta “*Engine/Extras/Android*” en el directorio de instalación de *Unreal Engine*. Por ejemplo “*D:\Unreal\UE_4.25\Engine\Extras\Android*” y ejecutar el archivo *SetupAndroid*.

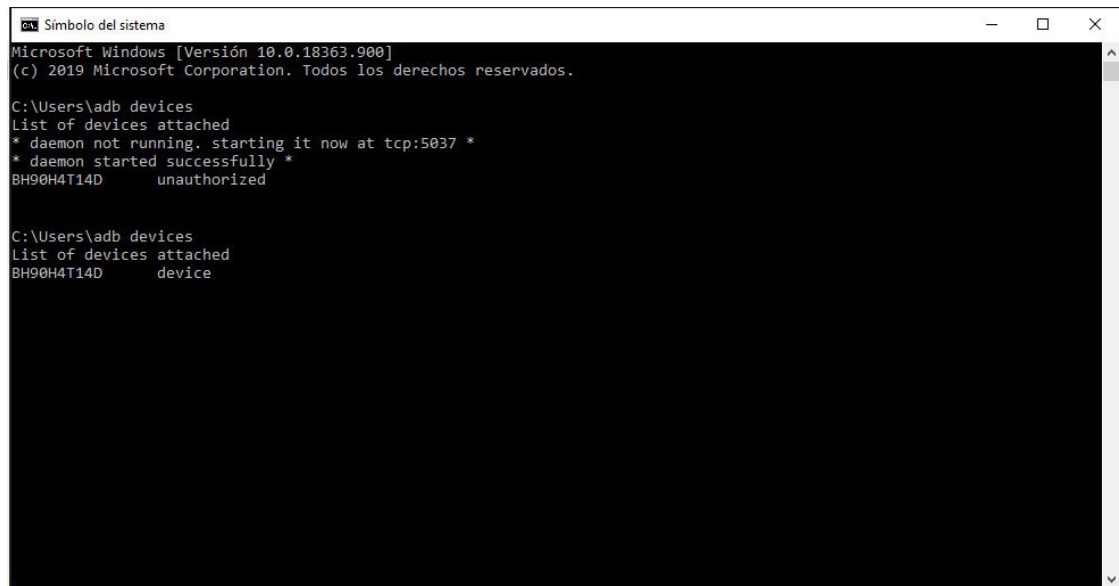
III.3.2.1.2 Modo desarrollador en dispositivos móviles Android

Para poder ejecutar el juego directamente mientras se está desarrollando, hay que activar las opciones de desarrollador en los dispositivos móviles.

En el menú de ajustes del teléfono o *tablet*, buscar la opción “*Acerca del teléfono*” y pulsar 7 veces sobre el “*Número de Compilación*”. Una vez hecho este paso, en la parte de ajustes se habilitará una nueva opción “*Opciones de desarrollador*” y activar “*Depuración USB*”.

Conecta el dispositivo y accede al terminal de comandos “*cmd*”. Teclear “*adb devices*” (Ilustración 18). En el dispositivo móvil marcar la opción de “*Permitir siempre en este*

ordenador” y ya estaría preparado el dispositivo para el desarrollo de videojuegos desde *Unreal Engine*.



```
Microsoft Windows [Versión 10.0.18363.900]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\adb>adb devices
List of devices attached
* daemon not running. starting it now at tcp:5037 *
* daemon started successfully *
BH90H4T14D    unauthorized

C:\Users\adb>adb devices
List of devices attached
BH90H4T14D    device
```

Ilustración 18. Terminal de comandos. Fuente: Elaboración propia

III.3.2.1.3 Creación de un proyecto en Unreal Engine

Al seleccionar un proyecto nuevo se abre el configurador del proyecto (Ilustración 19). Hay que cambiar el tipo de dispositivo final a *Mobile/Tablet, Scalable 3D or 2D* y sin *Starter Content*. El *Starter Content* es un contenido que viene por defecto en *Unreal* con materiales, partículas, mallas... Pero para dispositivos móviles no está optimizado.

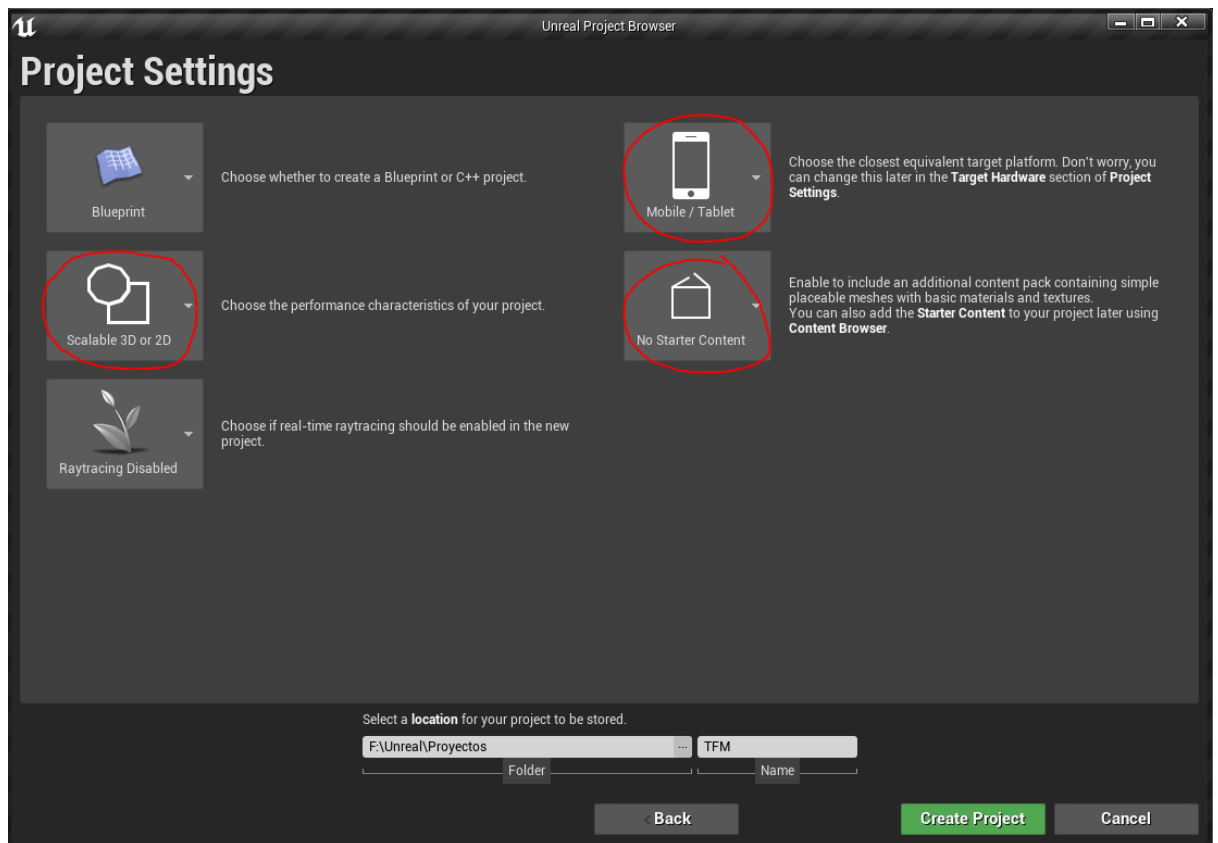


Ilustración 19. Creación de nuevo proyecto en Unreal Engine 4. Fuente: Elaboración propia

Una vez dentro del proyecto, en el menú “Edit > Project Settings > Platforms > Android” es necesario aceptar la licencia del SDK antes de empezar el desarrollo (Ilustración 20).

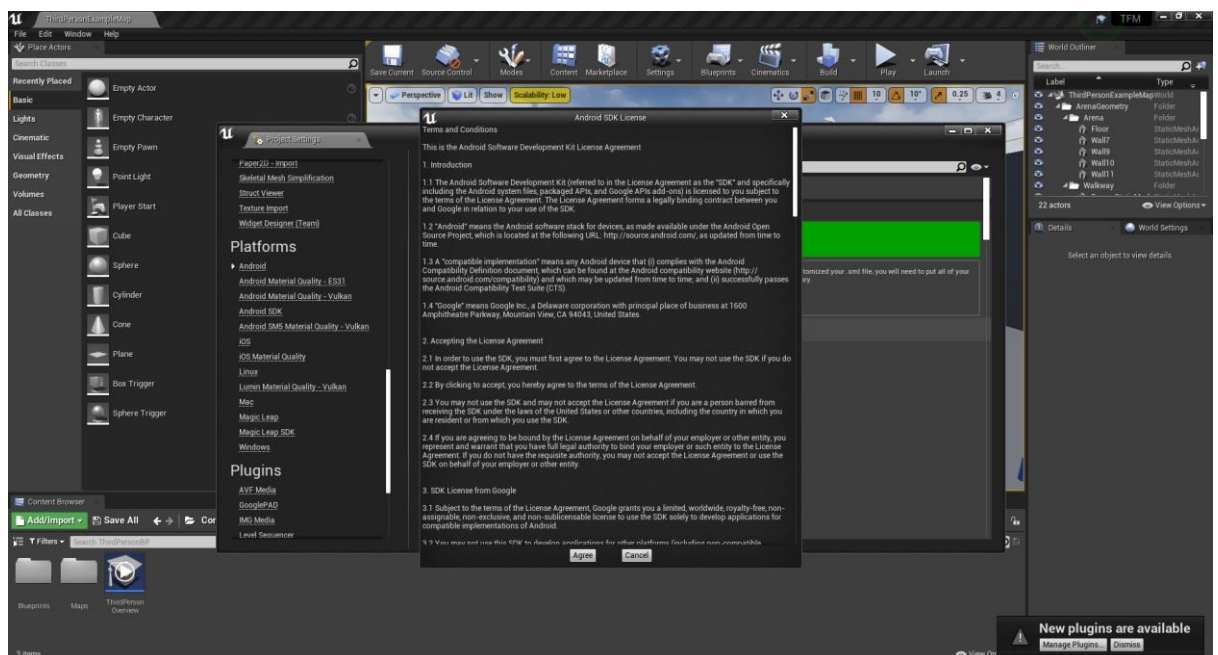


Ilustración 20. Aceptar licencia de SDK. Fuente: Elaboración propia

Para ejecutar el juego directamente en el dispositivo móvil, en la pestaña *Launch* debe aparecer el dispositivo que se ha configurado en la anterior sección. Cuando termine de procesar todos los archivos, el juego aparecerá en el dispositivo *Android* (Ilustración 21).

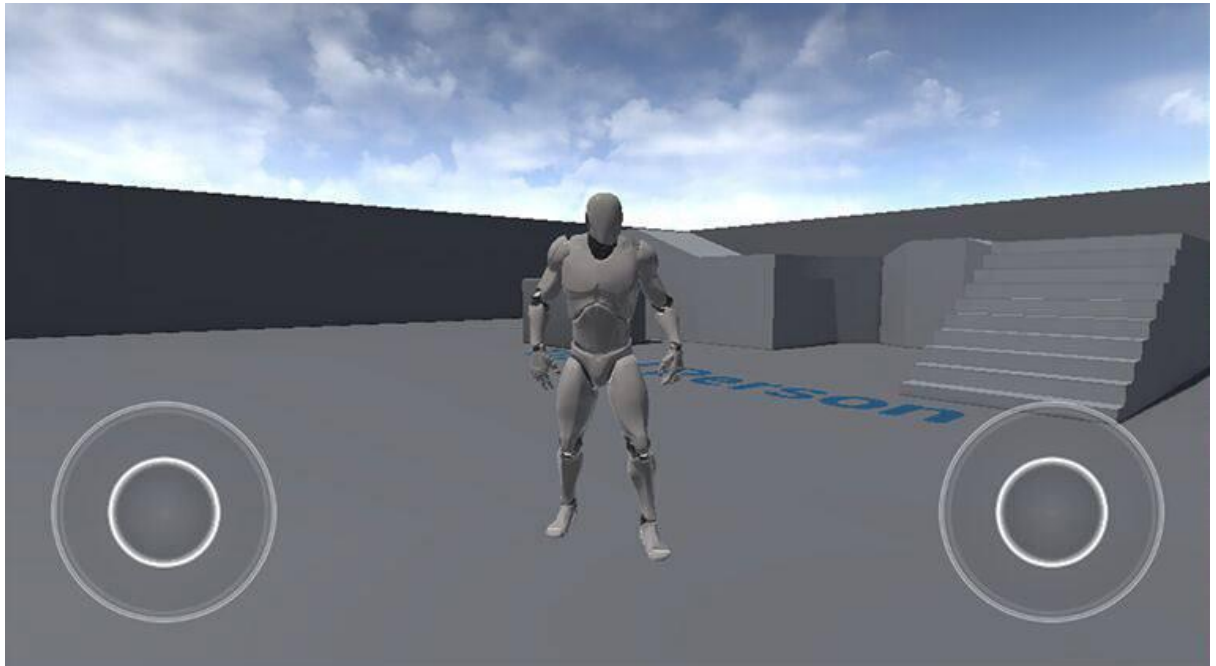


Ilustración 21. Ejecución del juego en dispositivo móvil. Fuente: (Salinas, 2020)

Una vez modificados los parámetros iniciales indicando que el juego va a ser desarrollado para móvil, se puede proseguir con el desarrollo normal de un juego como se indica en la sección Fases de desarrollo de un videojuego

III.3.2.2 Optimización de recursos

En la fase de desarrollo del proyecto hay que tener en cuenta la plataforma de destino. Las consolas o *PCs* por lo general tienen mayor capacidad de procesamiento y de soportar mejor la visualización de gráficos con mayor resolución que las plataformas móviles. También hay diferencias notables en rendimiento dependiendo de la consola; la *Nintendo Switch* no tiene tanta capacidad gráfica como la *PlayStation 4* o la *Xbox One*.

Al proceso de adaptar los recursos a las características de la plataforma objetivo para su mejor funcionamiento se le llama **optimización** (Epic Games, n.d.).

III.3.2.2.1 Complejidad del material.

El material es una herramienta que se le añade a los *mesh* o mallas de polígonos para controlar el aspecto visual de la escena, es decir, la textura que se le aplica a un objeto. El material define el tipo de superficie a partir de la cual su objeto parece estar hecho. Puede definir su color, su brillo, si se puede ver a través del objeto (transparencia) y mucho más. La creación de materiales también se realiza a través del lenguaje visual *Blueprints*.

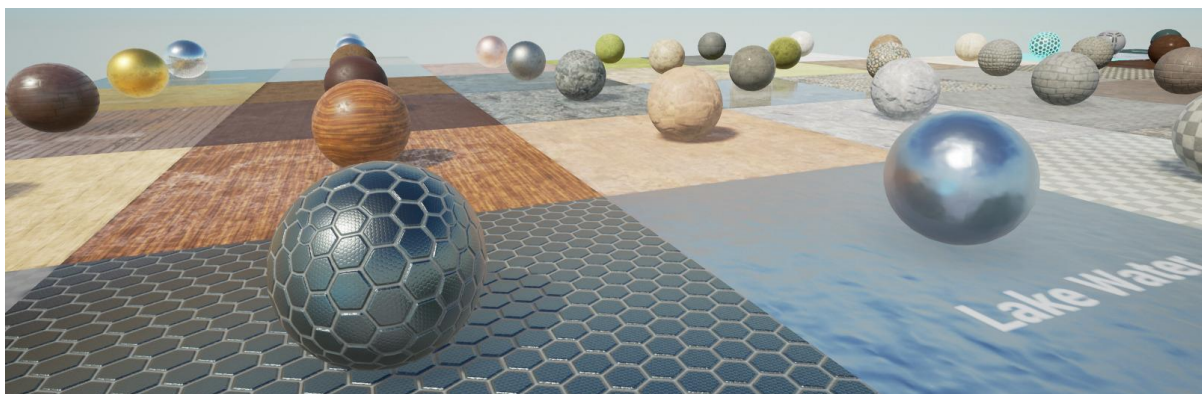


Ilustración 22. Materiales Unreal Engine 4. Fuente: Documentación oficial Unreal

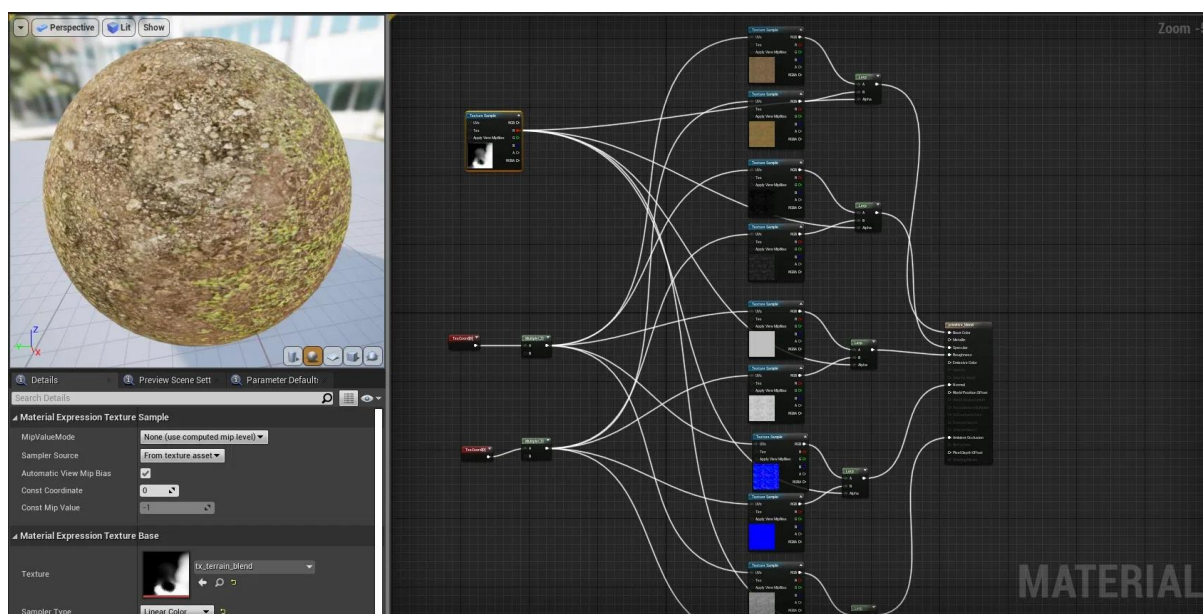


Ilustración 23. Blueprint materiales Unreal Engine 4. Fuente: (Erell, 2019)

En el editor de *Unreal* hay un modo de visualización que muestra al desarrollador dependiendo de unos códigos de colores con los que podemos discriminar el coste de visualización o procesamiento para las plataformas a los que se dirige: verde significa buen rendimiento, rojo brillante significa pesado rendimiento y blanco o rosa significa que el material es muy difícil de procesar en esa plataforma y el juego no va a rendir bien.

Para usar este modo de visualización, en la pantalla principal hay que hacer *click* en *View Modes*, se despliega una lista y buscar *Optimization Viewmodes* y escoger *Shader Complexity* o usar el atajo de teclado *Alt + 8*



Ilustración 24. Modos de optimización. Fuente: Elaboración propia

Estos son algunos ejemplos del modo de visualización:

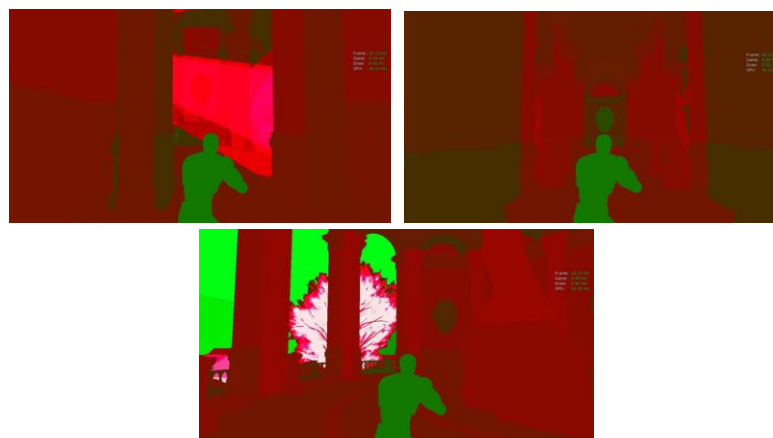


Ilustración 25. Modos de visualización: Fuente: Documentación oficial Unreal

III.3.2.2.2 Configuración de la calidad del material

Al crear contenidos para un proyecto que se ejecutará en dispositivos móviles de gama baja y alta, a menudo nos podemos encontrar que una característica u objeto funciona en un conjunto de dispositivos pero en otros no. Para solucionar este problema, *Unreal* cuenta con un sistema de configuración de la calidad del material. Este sistema permite al desarrollador crear un solo material que luego se puede ajustar al dispositivo sin tener en consideración la gama a la que pertenece.

Hay tres niveles de calidad: bajo, medio y alto. Y se pueden configurar de las siguientes maneras:

1. Desde consola o terminal

Abrir la consola del editor de *Unreal* presionando la ` (tilde invertida) y escribiendo *r.MaterialQualityLevel* seguido de uno de los siguientes valores (Tabla 1. Calidad del material UE4. Fuente: Elaboración propiaTabla 1):

Número de calidad	Configuración
0	Bajo
1	Alto
2	Medio

Tabla 1. Calidad del material UE4. Fuente: Elaboración propia

2. Dependiendo del dispositivo objetivo.

Usando un archivo de configuración (.ini) que será incluido en el juego, se puede establecer la calidad del material por defecto para la lista de dispositivos que está incluida.

En la barra de herramientas principal seleccionar *Windows > Developer Tools > Device Profiles*

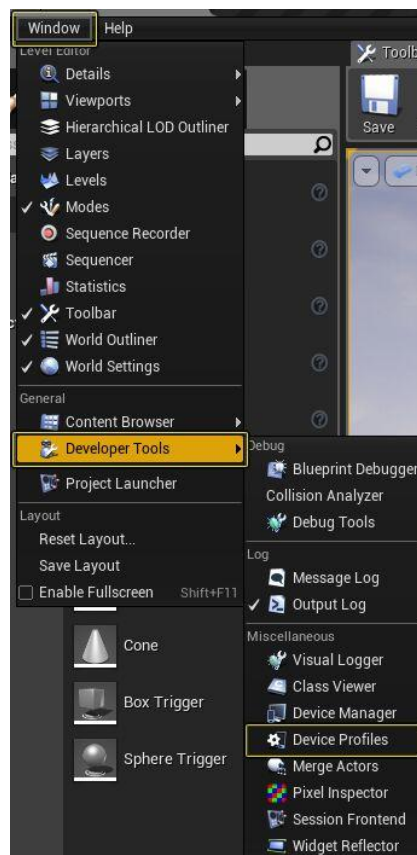


Ilustración 26. Device Profiles (1) UE4. Fuente: Elaboración propia

En los distintos perfiles de dispositivos, están los dispositivos Android:

- *Android_Low* para gama baja
- *Android_Mid* para gama media
- *Android_High* para gama alta

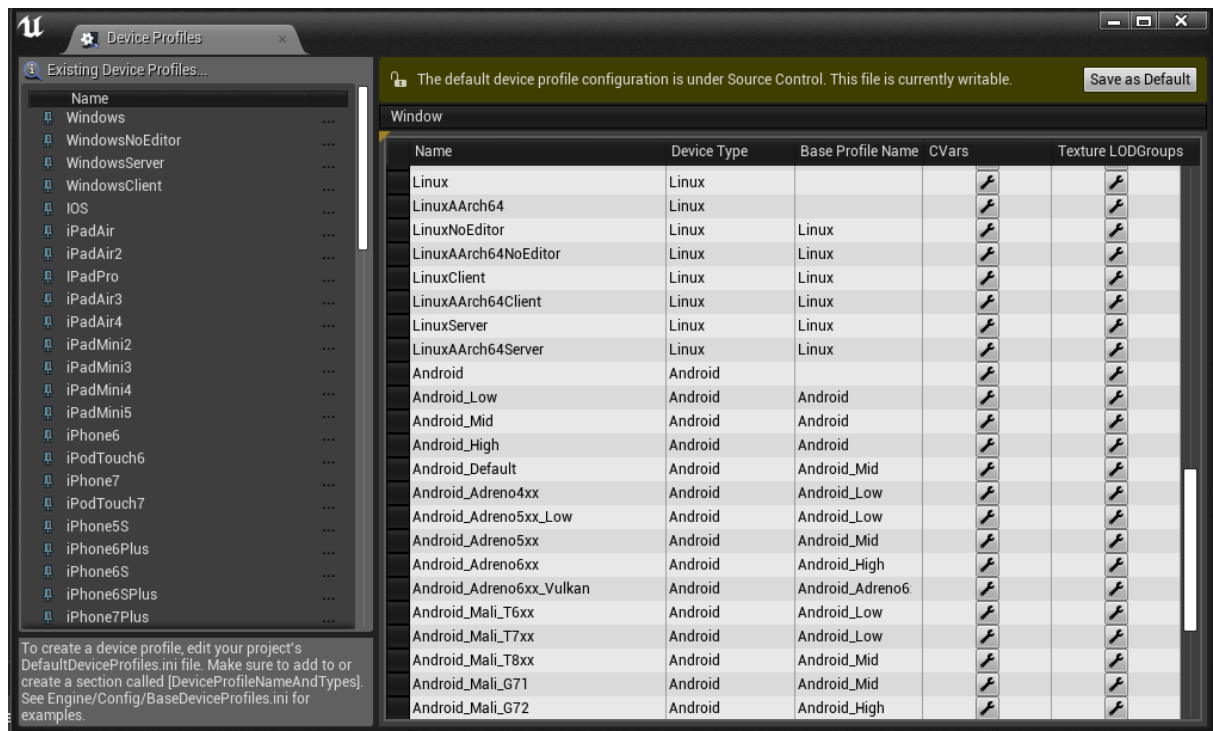


Ilustración 27. Device Profiles (2) UE4. Fuente: Elaboración propia

Para cada gama, se puede abrir la configuración pulsando en el icono de la llave inglesa.

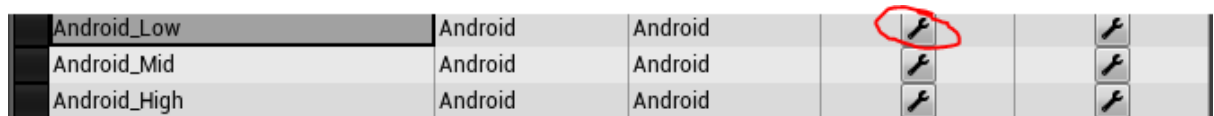


Ilustración 28. Device Profiles (3) UE4. Fuente: Elaboración propia

En la sección de variables de consola, *Rendering*, añadir *r.MaterialQualityLevel*.

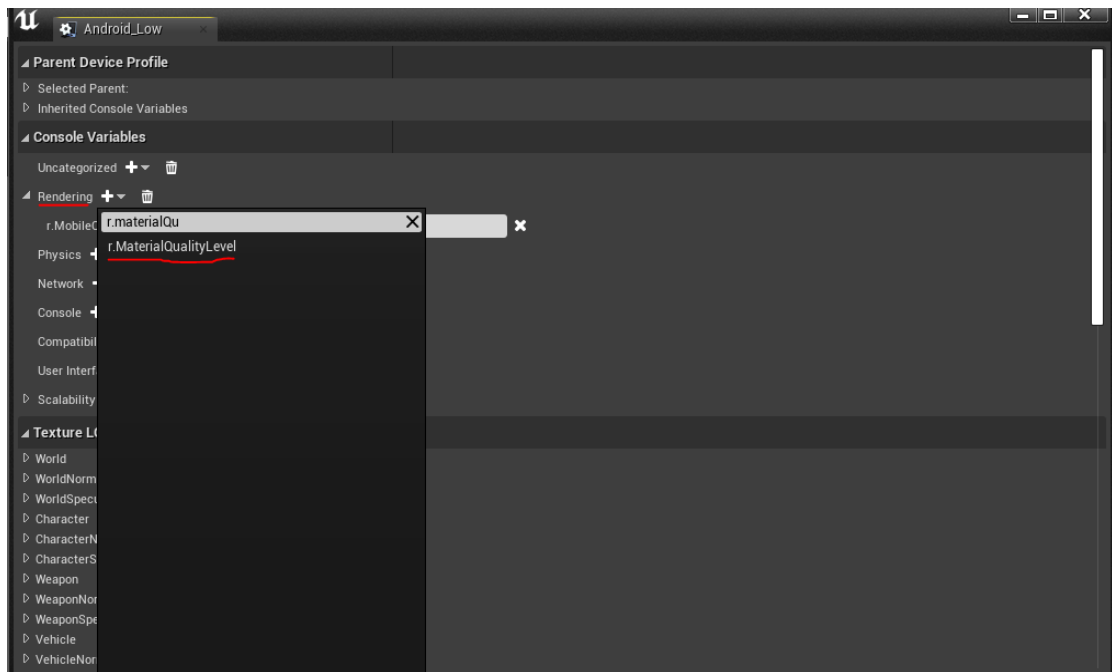


Ilustración 29. Device Profiles (4) UE4. Fuente: Elaboración propia

Una vez añadida esta variable, es necesario cambiar el valor predeterminado para que se ajuste a las características de los dispositivos de las distintas gamas.

3. En el blueprint del material

Existe un nodo que se puede incluir en el *blueprint*, llamado *Quality Switch* (Ilustración 30), que permite definir diferentes niveles de complejidad dentro de un material.



Ilustración 30. Nodo Quality Switch. Fuente: Documentación oficial Unreal

Por ejemplo, si el material contiene una operación matemática compleja, funcionará en dispositivos móviles de alta gama o consolas, pero no así en los de gama baja. Al usar este nodo se pueden definir otras operaciones simplificadas que no requieran de tanta capacidad de cómputo.

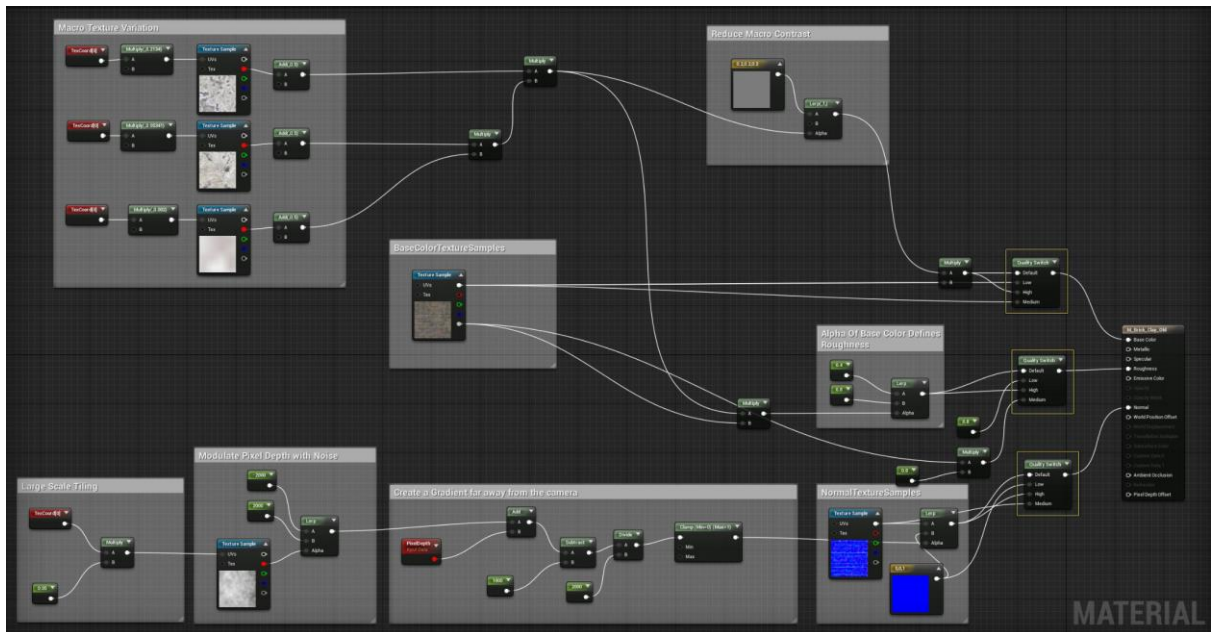


Ilustración 31. Blueprint material. Fuente: Documentación oficial Unreal

III.3.2.2.3 Mallas

Las mallas o *mesh* son piezas de geometría que están formadas por un conjunto de polígonos que se visualizan mediante la tarjeta gráfica. Cuantos más polígonos tenga la malla, mayor nivel de detalle tendrá el objeto y por tanto, necesitará una mayor capacidad de cómputo para visualizarla. Adaptar la cantidad de mallas o reducir la cantidad de polígonos que se visualizan al mismo tiempo por pantalla es una de las optimizaciones más potentes para dispositivos móviles. Otra de las opciones que se pueden utilizar para la optimización de mallas es el uso de niveles de detalle o también llamado **LOD** (*Level of Detail*) (Epic Games, n.d.).

El nivel de detalle es un “truco” de renderizado que ayuda a reducir el coste de visualización de los objetos que están lejos de la cámara del jugador, consiguiendo engañar al jugador haciendo creer que los objetos están más detallados de lo que realmente están. Los *LODs* funcionan almacenando múltiples versiones de menor resolución de un objeto, y en función de la distancia entre la cámara del jugador y el objeto se alternan de manera automática esas múltiples versiones.



Ilustración 32. LODs Unreal Engine 4. Fuente: Documentación oficial Unreal

En la Ilustración 32 se pueden observar el objeto base (en naranja) y las distintas versiones del objeto con menor resolución, y por tanto menor número de polígonos, definidas como *LOD 1*, *LOD 2* y *LOD 3*. El objeto Base también recibe el nombre de *LOD 0*.

A continuación se muestra un ejemplo de *LODs* en un árbol (Elaboración propia), en el que se le ha aplicado 3 niveles de *LODs*:



Ilustración 33. LOD 0. Fuente: Elaboración propia

Con *LOD 0* o base, la malla original tiene 13.390 triángulos (Ilustración 33).

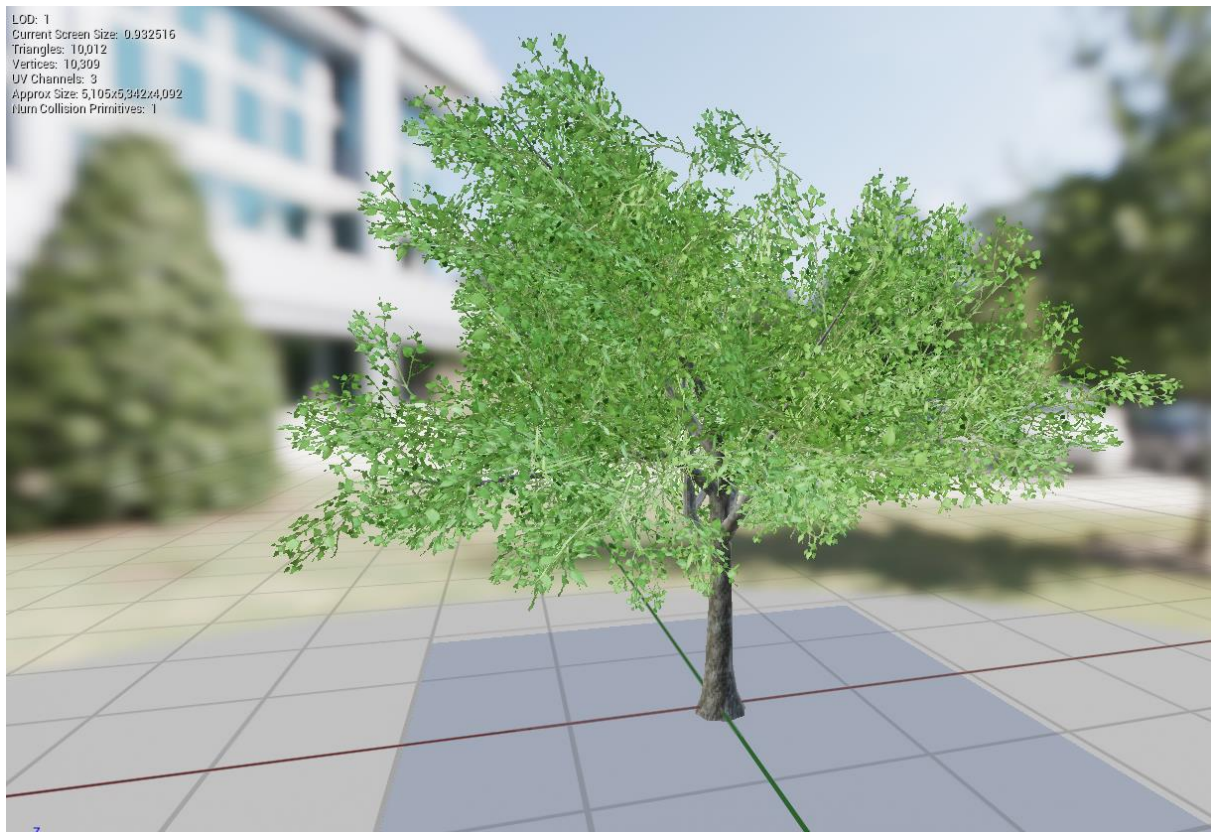


Ilustración 34. LOD 1. Fuente: Elaboración propia

Con el primer *LOD* que se le aplica, *LOD 1* (Ilustración 34), la cantidad de triángulos desciende hasta los 10.012 triángulos.



Ilustración 35. LOD 2. Fuente: Elaboración propia

En el segundo nivel de *LOD*, *LOD 2* (Ilustración 35), disminuye la cantidad de triángulos a 2.516 triángulos. Se puede observar que se ha reducido considerablemente la cantidad de triángulos mostrados en pantalla y aunque ha disminuido la calidad de la malla, sigue siendo una buena aproximación óptima a la malla original.

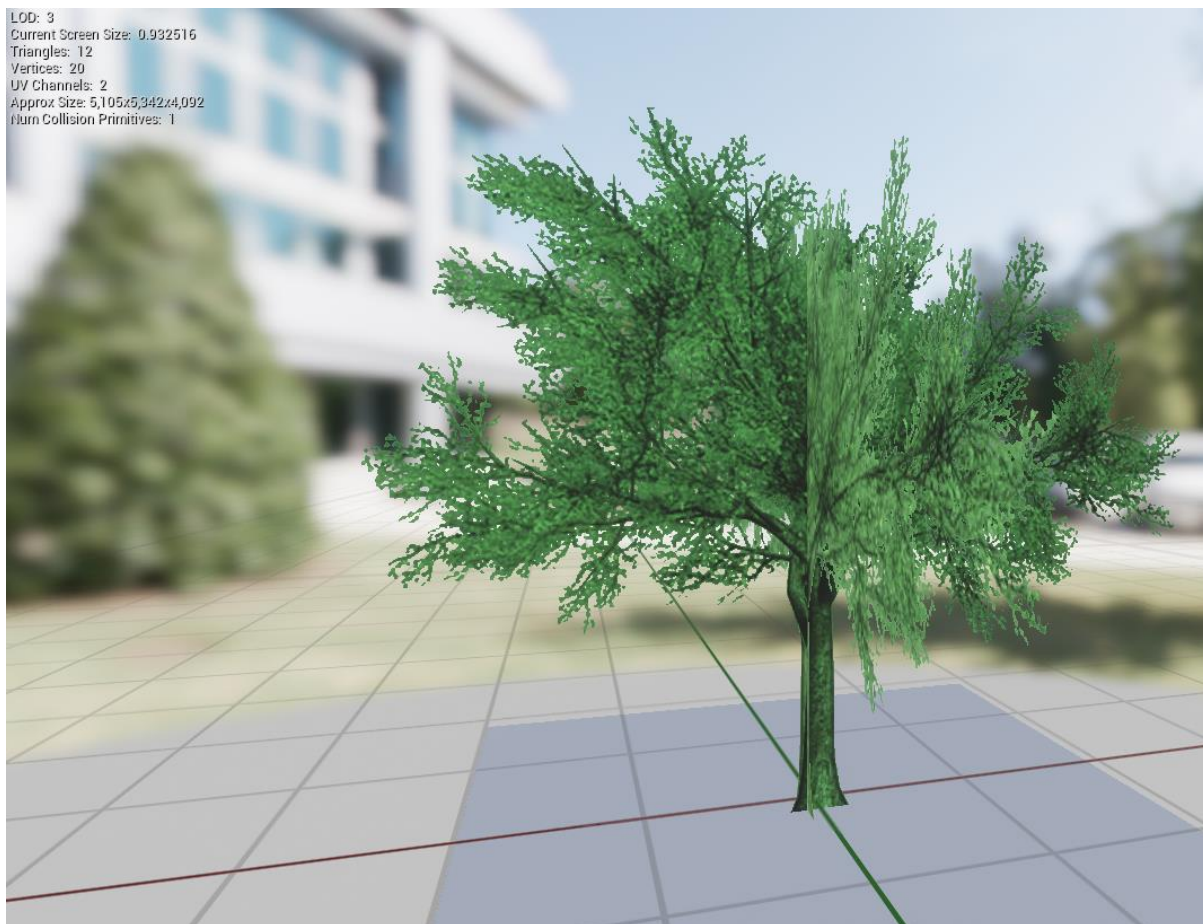


Ilustración 36. LOD 3. Fuente: Elaboración propia

En el último nivel de detalle, *LOD 3* (Ilustración 36), la cantidad de triángulos es de 12. Por lo que renderizar dicho árbol no supondrá ningún exceso de coste para cualquier dispositivo en el que se está visualizando.

III.3.2.3 Exportación a móvil/web

Una vez desarrollado, el videojuego deberá ser exportado a un formato específico para su posterior publicación y distribución. Para ello se ha de seleccionar la plataforma objetivo a la que se desee exportar. *Unreal Engine* cuenta con múltiples plataformas en las que el desarrollador puede exportar el producto, *Android*, *iOS*, *Linux*, *Windows*, *PS4*, *Nintendo Switch*...

Al realizar esta exportación, el motor de videojuegos realizará un empaquetado de todos los archivos necesarios para su correcta visualización, obteniendo como resultado uno o varios archivos dependiendo de la plataforma objetivo (Epic Games, n.d.). En *Android*, el formato de exportación es *APK*, y en *Windows*, el formato es *EXE*.

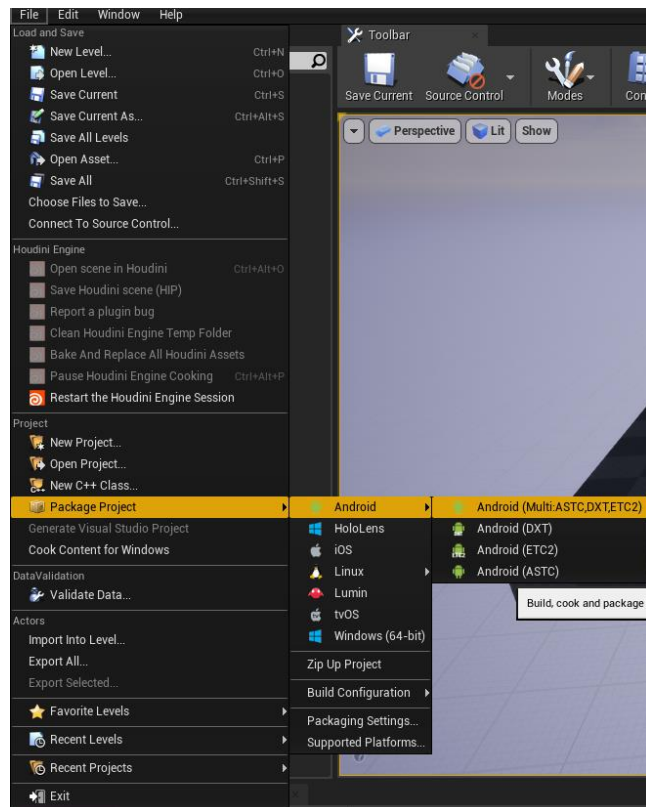


Ilustración 37. Exportación a dispositivos móviles. Fuente: Elaboración propia

III.3.3 Publicación y distribución

Una vez finalizado el desarrollo del juego, tal y como se mencionó en la Submission, última etapa de las fase de desarrollo, se puede publicar y distribuir entre las distintas tiendas existentes. En este TFM no se ha valorado la publicación física de juegos por lo que toda la publicación y distribución se centrará en tiendas digitales (Zarzecki, 2018) (Sabán, 2019).

III.3.3.1 Steam

Es una de las principales tiendas digitales de publicación de juegos centrada en videojuegos de escritorio. Esta plataforma tiene mucho éxito entre los usuarios, lo que significará que existe gran cantidad potencial de compradores del juego, aunque, en su contra al tener un acceso sencillo, multitud de desarrolladores publican a diario sus juegos en *Steam* haciendo que algunos juegos puedan pasar desapercibidos. Los principales problemas de *Steam* para desarrolladores noveles son las altas comisiones y el precio de entrada. Dar de alta un juego cuesta 100\$ y las comisiones suponen para *Steam* un 30% por venta.

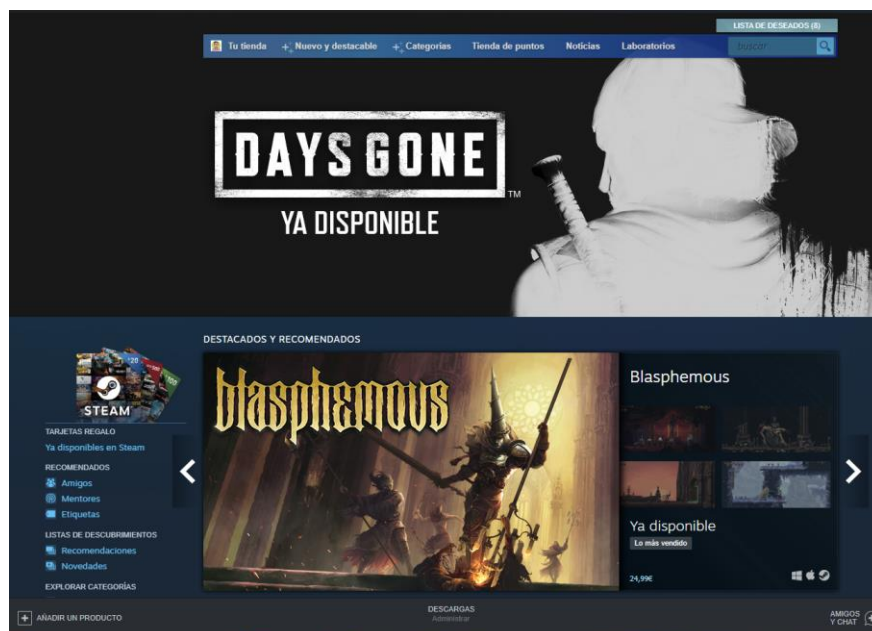


Ilustración 38: Steam. Fuente: Elaboración propia

III.3.3.2 Itch.io

Plataforma orientada a la distribución de juegos *indies* o independientes, estudios de pequeña envergadura y sin financiación. La tienda cuenta con más de 180.000 juegos, de los cuales, el 95% son gratuitos. A diferencia con *Steam*, cualquier desarrollador puede publicar sin costo alguno, lo que puede suponer que la calidad de los juegos no sea la más óptima. Se pueden distribuir juegos tanto para PC como para móviles.

Aunque la mayoría de los juegos son gratuitos, dispone de una opción para que el usuario pueda donar dinero al desarrollador.

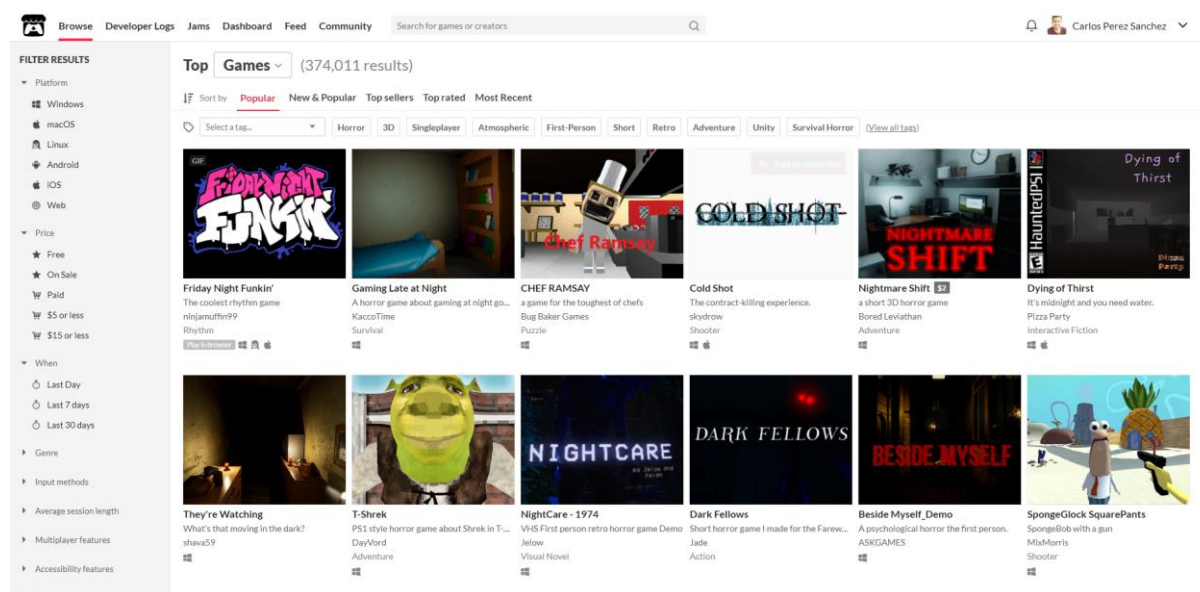


Ilustración 39: Itch.io. Fuente: Elaboración propia

III.3.3.3 App Store

Tienda exclusiva para aplicaciones y juegos desarrollados para dispositivos *Apple*. Uno de los principales obstáculos del desarrollo para dispositivos *Apple* es que todo lo que se desarrolle, debe realizarse desde un ordenador de marca *Apple*. La cuota de entrada a la plataforma para poder publicar es de 100\$ anuales y antes de poder publicar cualquier aplicación (o actualización), debe pasar por un proceso de aprobación y control de calidad por parte de la propia compañía de *Apple*.

Aunque presenta varias dificultades iniciales, el hecho de que *Apple* garantice que las aplicaciones pasan el filtro, puede hacer suponer que la calidad de estos productos es adecuada para poder venderse, y eso puede repercutir en que no exista una gran masificación en el número de aplicaciones de la tienda, a diferencia con *Steam* o *Itch io*.

III.3.3.4 Google Play Store

Es la tienda para dispositivos *Android*. La cuota de entrada a la plataforma para poder publicar es de 25\$. Después de eso, las aplicaciones pueden ser publicadas pasadas unas horas. Tener una entrada asequible y no disponer de un proceso de filtrado, hace que puedan existir en la tienda una gran cantidad de aplicaciones de dudosa calidad.

III.4 Tendencias actuales y futuras

Mucho han cambiado las cosas en el campo de los videojuegos desde 1950, cuando unos pocos pudieron disfrutar de juegos como *OXO* o *Tennis for Two*. El avance de la tecnología y la profesionalización del sector han producido que la industria del videojuego haya evolucionado a un ritmo vertiginoso durante los últimos años. Actualmente esta industria es una de las más potentes que existen. El número de ventas que producen estas empresas es muy elevado y por lo general, la mayoría de personas han jugado a algún juego alguna vez en su vida. Esta es una de las razones por la que esta industria es muy atractiva a la hora de innovar, y que constantemente está en evolución. La tendencia actual se centra en el jugador, el usuario, que sea el protagonista y personaje principal de la trama.

III.4.1 Realidad Virtual (VR)

Actualmente, la visualización gráfica en los videojuegos está llegando a la cima de su evolución. Los juegos se parecen tanto a la realidad que se hace difícil diferenciar juego de realidad (Pure Electronic Games iOnline, 2019).



Ilustración 40. Flight Simulator. Fuente: (Venckunas, 2020)

Por esta razón, tecnologías como la realidad virtual (VR), en la que el jugador se sumerge por completo en el mundo virtual creando una experiencia sensorial dentro de un ambiente artificial, haciendo uso de gafas especiales y auriculares, están en auge. Este tipo de gafas te cubren los ojos de manera que solo puedes ver lo que se proyecta en la pantalla de las propias gafas y disponen de unos sensores que reconocen el movimiento de la cabeza, de manera que cuando se gira la cabeza, el movimiento se realizará también en el mundo virtual (Fernández, 2018). Actualmente hay dos tipos de gafas, las que tienen su propia pantalla incorporada como la *Oculus Rift*, o las que necesitan de un *smartphone* para hacer de pantalla como por ejemplo las *GearVR* de *Samsung*.

Que el juego se haga mucho más inmersivo, es algo que todo jugador espera probar algún día, ya que los juegos de realidad virtual ofrecen una experiencia de juego diferente y completamente nueva donde al final, el jugador puede sentirse realmente protagonista del videojuego.

III.4.2 Realidad Aumentada (AR)

Otro de los campos en los que se está avanzando es en la llamada Realidad Aumentada. Esta tecnología permite a los usuarios superponer objetos digitales en entornos reales. No solo la están utilizando videojuegos como *Pokemon Go* (Ilustración 41), sino que muchas empresas la están utilizando para que el usuario pueda observar sus productos en 3D, mostrando los mismos desde todos los ángulos, lo que permitiría por ejemplo,

contemplar cómo puede quedar una cocina o un mueble en una habitación antes de comprarlo (Ilustración 42).



Ilustración 41. Pokemon Go. Fuente: (Peru.com, 2016)

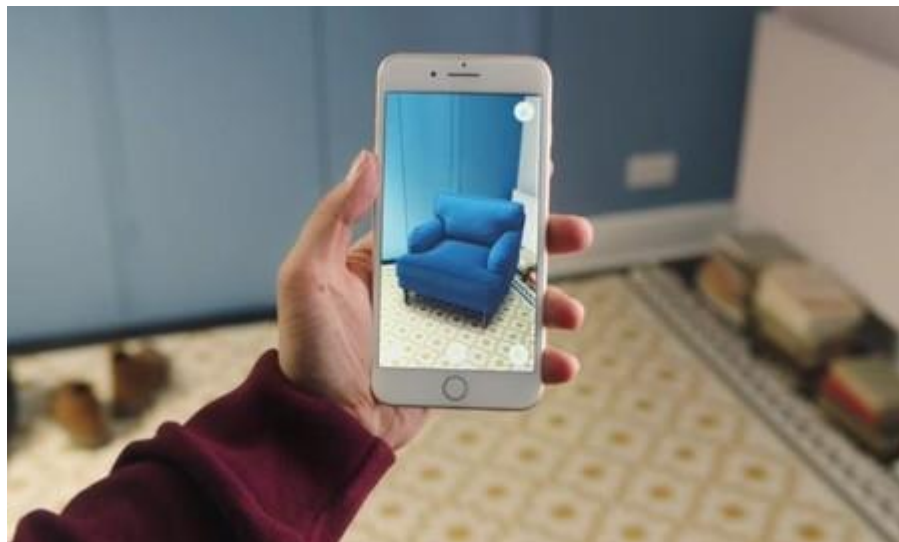


Ilustración 42. Visualización de un mueble con RA. Fuente: (Pastor, 2017)

También abre una puerta a su uso en el sistema educativo puesto que podemos visualizar elementos que de otra manera no podrían visualizarse tales como ciudades, ciudades de la antigüedad, animales, dinosaurios, plantas, cuerpo humano...



Ilustración 43. Dinosaurio 3D con RA. Fuente: (Cantudo, 2020)

III.4.3 Juegos en la nube

Gracias al aumento de la velocidad en las conexiones a Internet, la interacción con la “nube” está abriéndose hueco en la industria de los videojuegos.

El juego en la nube o jugar en la nube, es un modo de juego que permite al usuario interactuar en línea a través de su PC, consola o dispositivo móvil, mientras que el juego se ejecuta en el servidor de la compañía desarrolladora y se transmite directamente a dispositivos que tengan acceso a este servidor. Esto permite el acceso a los juegos sin necesidad de que el usuario tenga dispositivos potentes, ya que es el servidor de la compañía el que ejecuta el juego con todas las necesidades de procesamiento que se requieran.

Empresas como *Google* han desarrollado *Google Stadia*, que definen como: *“Stadia es la plataforma de videojuegos de Google que te permite jugar al instante en las pantallas que ya tienes. Juega directamente en tus dispositivos compatibles favoritos. Están siempre listos allí donde tengas conexión a internet, sin esperas, descargas ni actualizaciones.”*, y que según su web, *“Stadia emite en streaming tus juegos directamente en un ordenador, un teléfono o tableta compatibles, o un televisor con Chromecast Ultra, de forma parecida a como reproduces en streaming música o series. Con Stadia, puedes empezar a jugar en cuestión de segundos (Google Stadia, s.f.).”*



Ilustración 44. Videojuego español exclusivo de Google Stadia, Gylt. Fuente: Tequila Works

III.4.4 Otros sectores

Actualmente, los motores de videojuegos no solo se están utilizando para desarrollar videojuegos. *Unreal Engine*, *Epic Games* –empresa desarrolladora del motor – proporciona a diferentes sectores distintas herramientas para trabajar. Por ejemplo, *Unreal* proporciona herramientas para crear entornos 3D que pueden cambiar en tiempo real, usado por arquitectos, se pueden realizar diseños de gran resolución de automóviles, entrenamiento y simulación, desarrollo de cine y televisión (*Epic Games*, s.f.).

Podemos encontrarnos series como *The Mandalorian* (Ilustración 45), que utilizan *Unreal* para modificar escenarios y añadir efectos especiales en tiempo real, haciendo uso de paneles gigantes *LED* donde se pueden ver escenarios creados en *Unreal Engine 4*.

Una de las ventajas de utilizar esta tecnología es que los actores utilizan referencias reales y no escenarios de fondos verdes–*croma*–, lo que les ayuda a tener una sensación más inmersiva, para realizar una mejor interpretación.

De cara a la producción no solo supone ventajas al crear contenido, también un ahorro de tiempo y dinero al no tener que desplazarse a distintas localizaciones de grabación, pudiéndose realizar íntegramente en un solo plató. Además, cualquier modificación del contenido se puede realizar al momento. Por ejemplo si el equipo de producción decide que se incluyan nubes, se activan; si se quiere cambiar la iluminación de la escena, sobre la marcha se modifica y en tiempo real se cambian los paneles *LED*. Con *The Mandalorian* no se ha estrenado esta tecnología, directores como *Spielberg* o *Denis Villeneuve* han

hecho uso de esta tecnología en *Ready Player One* o *Blade Runner 2049* (Santamaría, 2019).

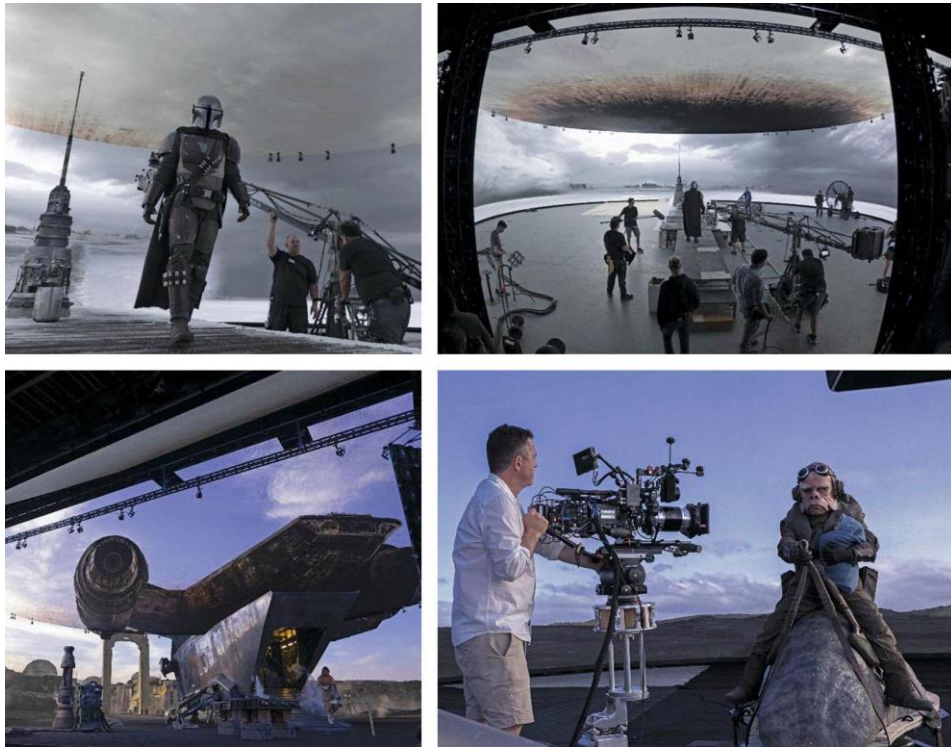


Ilustración 45. Detrás de las escenas de *The Mandalorian*. Fuente: Reddit

III.5 Conclusiones

Para una gran cantidad de la población, videojuego puede ser sinónimo de pasatiempo, diversión y ocio. Sin embargo, en los últimos años, los videojuegos pueden usarse también como herramienta educativa que permite a los estudiantes desarrollar competencias en sus procesos de aprendizaje. Captar la atención de los estudiantes es el reto principal de cualquier docente y a través de esta herramienta, puede ayudar en esta tarea.

En este sentido, cabe destacar que existen videojuegos creados para su uso docente, ejemplo de esto puede ser el videojuego *MinecraftEDU*, una adaptación del popular videojuego *Minecraft* (Bourne & Salgado, 2016).

Creado por programadores y educadores de Estados Unidos y Finlandia, *MinecraftEDU* es la versión educativa del popular videojuego *Minecraft*. Consiste en interactuar en un mundo virtual generado por bloques (similares al *Lego*) donde un jugador puede realizar construcciones libres con la ayuda de otros jugadores y resolver situaciones de supervivencia propias del videojuego. En la versión educativa, es el profesor quien tiene el control del entorno virtual, y los jugadores (alumnos) tienen que resolver problemas de carácter educativo; algunos de ellos pueden solucionarse individualmente o en grupo. Estas actividades ayudan al estudiante a desarrollar destrezas de aprendizaje

motivadas a continuar en el videojuego. Actualmente *MinecraftEDU* se puede jugar en línea y es utilizada por más de 5000 escuelas y colegios entre Estados Unidos y Europa.

La utilización de *MinecraftEDU* y programas como *Scratch* (mencionado anteriormente), o en general el mundo de los videojuegos, aumenta la motivación en el aprendizaje, ayuda al alumno a adquirir conocimientos de una manera atractiva y contribuye al desarrollo de competencias.

Creo que el uso de la tecnología del videojuego en el ámbito educativo presenta y presentará **más ventajas que desventajas**, y en los años venideros estoy convencido que su uso en las escuelas será cada mayor.

BLOQUE 2

PROYECCIÓN

DIDÁCTICA

IV. Bloque 2. Proyección Didáctica

IV.1 Introducción

En esta sección se elabora la Unidad Didáctica *Desarrollo de juegos para dispositivos móviles*, correspondiente al módulo profesional *Programación multimedia y dispositivos móviles (0489)* del *Ciclo Formativo de Grado Superior en Desarrollo de Aplicaciones Multiplataforma*. Dicho módulo está encuadrado dentro del segundo curso del ciclo, con una distribución horaria de 4 horas semanales, haciendo un total de 84h en el curso. El personal docente que puede impartir el módulo en el que se encuadra la Unidad Didáctica es de la especialidad de Informática pudiendo ser Profesor de Enseñanza Secundaria o Catedrático de Enseñanza Secundaria.

Esta Unidad Didáctica se basa en la normativa vigente que se especificará a lo largo de este capítulo. Asimismo, se ha realizado una contextualización en el centro donde se va a impartir el Ciclo Formativo, aportando datos como la situación geográfica del instituto, aspectos socioculturales y económicos, la descripción del centro y características del alumnado

A lo largo de este capítulo, se elaborarán los siguientes apartados incluidos en la Unidad Didáctica *Desarrollo de juegos para dispositivos móviles*: Normativa actual en el que se basa esta sección del TFM, Competencias, Orientaciones pedagógicas, Líneas de actuación, Resultados de Aprendizaje, Contenidos, Objetivos didácticos, la Metodología seleccionada y la Transversalidad. Asimismo, esta Proyección Didáctica también incluye aspectos relevantes como son la Evaluación y Recuperación y la Atención a la Diversidad.

Durante el curso 2019/2020 la pandemia causada por el COVID-19 supuso un cambio en la vida diaria de muchas personas, incluidos profesores y alumnos. Mantener la distancia social o la implementación de metodologías completamente digitales ha supuesto un reto para la mayoría de sectores profesionales, puesto que se han tenido que adaptar lo más rápidamente posible a la nueva situación que se presentaba. Clases *online*, clases semipresenciales, clases presenciales afectadas por ausencias debidas a confinamientos de alumnos, uso generalizado de herramientas digitales (ordenadores, *tablets*, móviles...) que cada docente debe acostumbrarse a usar, han supuesto un cambio demasiado radical en un corto espacio de tiempo y que no todos los docentes han asumido a la misma velocidad. No obstante, se ha generalizado el uso de plataformas tales como *Google Meet*, *Google Classroom*..., así como el uso de presentaciones tipo *Power Point*, *Prezi* o *Google Presentaciones* en las aulas.

Esta Unidad Didáctica, basándose en la normativa del COVID-19, se ha adaptado para prevenir cualquier situación excepcional de confinamiento y no presencialidad en el aula por parte del alumnado.

IV.2 Contextualización del centro

IV.2.1 Plan de centro

La Unidad Didáctica que se va a desarrollar en este TFM está enmarcada en el IES Jabalcuz, situado en Jaén.



Ilustración 46. Entrada del centro IES Jabalcuz. Fuente: Elaboración propia

El IES Jabalcuz es un Centro público dependiente de la Consejería de Educación de la Junta de Andalucía que viene desarrollando su labor educativa y formativa desde hace ya casi cuarenta años, se trata de un centro educativo público donde se cursan enseñanzas de Educación Secundaria Obligatoria (ESO), de Bachillerato (en sus modalidades de Humanidades y Ciencias Sociales y de Ciencias de la Naturaleza y la Salud) y, desde el curso 2003-04, de un Ciclo Formativo de Grado Superior en la especialidad de Animación Sociocultural y Turística (IES Jabalcuz, 2008) y de un Ciclo Formativo de Grado Superior en Desarrollo de Aplicaciones Multiplataforma.

El IES Jabalcuz se ubica en la zona noreste de la ciudad de Jaén (Ilustración 47), en un barrio, el del Polígono del Valle, situado junto a una de las áreas de expansión de mayor proyección de nuestra capital, el sector norte, y a un tiro de piedra de la ciudad universitaria de Jaén. Se trata de un barrio de gran tradición y raigambre cuya población, eminentemente obrera, goza de unos estándares de renta económica medio-bajos, con un índice de paro ciertamente significativo, especialmente entre la juventud, comparado con otros barrios anejos, ya instalados en la zona centro, y cuyo bagaje cultural puede calificarse, asimismo, de medio-bajo, dadas la actividad laboral genérica de la zona y la ausencia casi absoluta de servicios públicos que permitan paliar en cierta medida tal carencia: biblioteca de barrio, centro de ocio, centro cívico, asociaciones culturales...

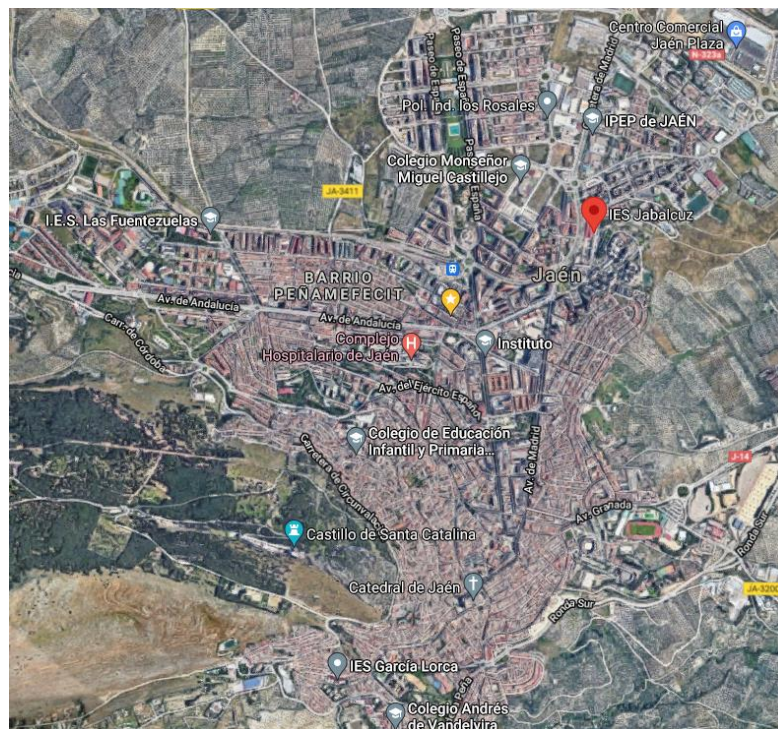


Ilustración 47. Localización del IES Jabalcuz (Jaén). Fuente: Google Maps

El centro acoge al alumnado del barrio del Valle, pero también y gracias al servicio de transporte escolar, tiene asignado a los alumnos que provienen de la periferia de Jaén, así como de la localidad de Villatorres para los alumnos de Bachillerato.

Sobre la base de las nuevas exigencias y demandas que la sociedad plantea a los centros de enseñanza en su conjunto y a los profesionales que allí desempeñan su labor de manera particular, el instituto en el que he realizado las prácticas del máster, contempla un compromiso firme de buscar espacios comunes de actuación Centro-Comunidad, esto es, de establecer vías de colaboración con instituciones, organizaciones y servicios que favorezcan la apertura del centro conectándolo con su entorno y que hagan de éste un verdadero motor, una dinamizador, de la realidad educativa y cultural de la zona.

El Centro cuenta con, aproximadamente, un total de 63 profesores, de los cuales más de 50 cuentan con destino definitivo en el IES Jabalcuz y son miembros de la plantilla orgánica del mismo. A ellos hay que añadir un pequeño núcleo del profesorado en situación administrativa de comisión de servicios, prácticas e interinidad. Los cargos unipersonales son: Director, un Jefe de Estudios, una Vicedirectora, un Secretario y una Jefa de Estudios Adjunta. El total de tutores/as es aproximadamente de 25.

Asimismo, el Centro cuenta con servicios y desarrolla programas y proyectos educativos innovadores que completan su oferta educativa:

- A partir del curso escolar 2008-2009, el IES Jabalcuz oferta enseñanza bilingüe a su alumnado. En la actualidad, el Centro es totalmente bilingüe para la etapa de ESO, y parcialmente en Bachillerato.

- Plan de Autoprotección y Prevención de Riesgos Laborales.
- Servicio de Transporte Escolar.
- Igualmente, desde el curso escolar 05/06, el Instituto forma parte de la red de Centros que incorporan las nuevas Tecnologías de la Información y la Comunicación a la práctica docente y que hoy se constituyen en Escuela TIC 2.0.
- Proyecto Escuela: Espacio de Paz.
- Proyecto de “Coeducación”.
- Escuelas Deportivas.

El IES Jabalcuz cuenta con una Asociación de Madres y Padres que, especialmente en los últimos años lleva trabajando con ilusión y rigor, en la organización de actividades para el Día del Libro, San Antón, etc. y auspiciando Mercadillos de Libros Solidario y jornadas de convivencia en Navidad, Día de Andalucía y Fin de Curso, etc.

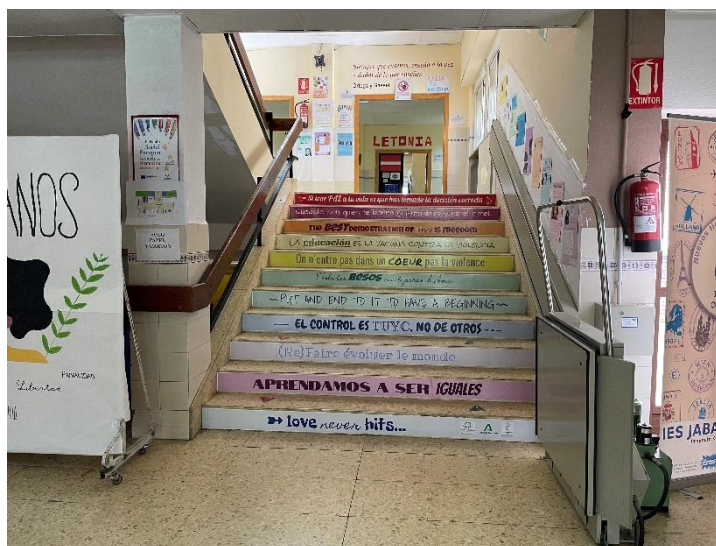


Ilustración 48. Hall del centro IES Jabalcuz. Fuente: Elaboración propia

IV.2.2 Características del alumnado

Las edades del alumnado están comprendidas entre los 12 y los 21 años, si bien cabe decir que hay algunos alumnos/as de mayor edad. Por sexos, el número de alumnas es prácticamente equivalente al de alumnos. Se aprecia que la mayoría son nacidos en Jaén capital.

Como corresponde a un centro público, se trata de un alumnado heterogéneo en sus etapas obligatorias.

El alumnado de la ESO proviene, en su inmensa mayoría, de los centros de primaria de la capital adscritos al IES Jabalcuz (Cándido Nogales, Santo Domingo, San José de Calasanz). A ellos hay que sumar el alumnado usuario de transporte escolar proveniente, principalmente, del colegio Alfredo Cazabán.

En Bachillerato el centro recibe, además, alumnado de los IES Santa Teresa y, en mayor medida, Llanos de la Viña, de la vecina localidad de Villatorres, quienes, al igual que muchos otros compañeros que residen en la periferia de la ciudad, hace uso habitual del servicio de transporte escolar con el que cuenta el Centro.

También hay que señalar que el centro tiene alumnos de diversas nacionalidades, generalmente, procedentes de familias sudamericanas que se han asentado en la localidad.

IV.3 Unidad Didáctica

IV.3.1 Introducción

Esta Unidad Didáctica se encuadra dentro de la programación del módulo *“Programación multimedia y dispositivos móviles”*, perteneciente al Ciclo Formativo de Grado Superior en Desarrollo de aplicaciones multiplataforma, que a su vez pertenece a la familia profesional de Informática y Comunicaciones.

En la ORDEN de 16 de junio de 2011, se desarrolla el currículo correspondiente al título de Técnico Superior en Desarrollo de Aplicaciones Multiplataforma, y concretamente el módulo al que pertenece esta Unidad Didáctica está asignado al código 0489 y puede ser impartido por profesionales de la especialidad de Informática pertenecientes al cuerpo de Profesores de Enseñanza Secundaria (PES) o por Catedráticos de Enseñanza Secundaria.

Este módulo se imparte en el primer trimestre del segundo curso y tiene asignadas 4 horas semanales y 84 horas totales.

IV.3.2 Descripción de la Unidad Didáctica

A continuación se muestra un resumen con los datos más relevantes de la Unidad Didáctica que se desarrolla en este capítulo.

- **Título:** Desarrollo de juegos para dispositivos móviles
- **Etapas:** CFGS Desarrollo de Aplicaciones Multiplataforma
- **Módulo:** Programación multimedia y dispositivos móviles
- **Curso:** Segundo
- **Trimestre:** Primero
- **Número de sesiones:** 5 sesiones de 2 horas cada una
- **Temporalización:** 10 horas

IV.3.3 Normativa

La normativa por la que se rige la Unidad Didáctica perteneciente a este TFM es la que sigue. A causa de la pandemia del COVID-19, se hace una distinción entre la normativa general y la normativa específica relativa a la situación excepcional de pandemia.

IV.3.3.1 Normativa General

Real Decreto 450/2010, de 16 de abril, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Multiplataforma y se fijan sus enseñanzas mínimas.

ORDEN de 16 de junio de 2011, por la que se desarrolla el currículo correspondiente al título de Técnico Superior en Desarrollo de Aplicaciones Multiplataforma.

IV.3.3.2 Normativa COVID

RD 21/2020, de 9 de junio, de medidas urgentes de prevención, contención y coordinación para hacer frente a la crisis sanitaria ocasionada por el COVID-19.

Instrucciones de 6 de julio de 2020, de la viceconsejería de educación y deporte, relativas a la organización de los centros docentes para el curso escolar 2020/2021, motivada por la crisis sanitaria del covid19.

IV.3.4 Competencia General

La competencia general de este título consiste en desarrollar, implantar, documentar y mantener aplicaciones informáticas multiplataforma, utilizando tecnologías y entornos de desarrollo específicos, garantizando el acceso a los datos de forma segura y cumpliendo los criterios de «usabilidad» y calidad exigidas en los estándares establecidos.

IV.3.5 Orientaciones pedagógicas

Este módulo profesional contiene la formación necesaria para desempeñar la función de desarrollo de aplicaciones multimedia, juegos y aplicaciones adaptadas para su explotación en dispositivos móviles.

La función de desarrollo de aplicaciones multimedia, juegos y aplicaciones adaptadas para su explotación en dispositivos móviles incluye aspectos como:

- c) El desarrollo de juegos 2D y 3D utilizando las funcionalidades que ofrecen los motores de juegos, así como su puesta a punto e implantación en dispositivos móviles.

IV.3.6 Competencias profesionales, personales y sociales

La formación del módulo contribuye a alcanzar las competencias profesionales, personales y sociales de este título que se relacionan a continuación:

- d) Gestionar entornos de desarrollo adaptando su configuración en cada caso para permitir el desarrollo y despliegue de aplicaciones.
- g) Integrar contenidos gráficos y componentes multimedia en aplicaciones multiplataforma, empleando herramientas específicas y cumpliendo los requerimientos establecidos.
- h) Desarrollar interfaces gráficos de usuario interactivos y con la usabilidad adecuada, empleando componentes visuales estándar o implementando componentes visuales específicos.
- i) Participar en el desarrollo de juegos y aplicaciones en el ámbito del entretenimiento y la educación empleando técnicas, motores y entornos de desarrollo específicos.
- j) Desarrollar aplicaciones para teléfonos, PDA y otros dispositivos móviles empleando técnicas y entornos de desarrollo específicos.
- m) Empaquetar aplicaciones para su distribución preparando paquetes auto instalables con asistentes incorporados.

IV.3.7 Líneas de actuación

Las líneas de actuación en el proceso de enseñanza-aprendizaje que permiten alcanzar los objetivos del módulo versarán sobre:

- d) El desarrollo de aplicaciones que integran objetos multimedia.
- f) El desarrollo de juegos 2D y 3D aplicando técnicas específicas y utilizando instrucciones gráficas para establecer efectos sobre objetos o imágenes.

IV.3.8 Objetivos Generales

La formación del módulo contribuye a alcanzar los objetivos generales de este ciclo formativo que se relacionan a continuación:

- h) Emplear herramientas de desarrollo, lenguajes y componentes visuales, siguiendo las especificaciones y verificando interactividad y usabilidad, para desarrollar interfaces gráficos de usuario en aplicaciones multiplataforma.
- i) Seleccionar y emplear técnicas, motores y entornos de desarrollo, evaluando sus posibilidades, para participar en el desarrollo de juegos y aplicaciones en el ámbito del entretenimiento.

j) Seleccionar y emplear técnicas, lenguajes y entornos de desarrollo, evaluando sus posibilidades, para desarrollar aplicaciones en teléfonos, PDA y otros dispositivos móviles

s) Establecer procedimientos, verificando su funcionalidad, para desplegar y distribuir aplicaciones.

IV.3.9 Resultados de aprendizaje

El resultado de aprendizaje asociado a esta Unidad Didáctica es el siguiente:

5. Desarrolla juegos 2D y 3D sencillos utilizando motores de juegos.

Los criterios de evaluación asociados a este resultado de aprendizaje se detallarán en la posterior sección Criterios de evaluación.

IV.3.10 Contenidos

Se detallan los contenidos básicos que contendrá la Unidad Didáctica *“Desarrollo de juegos para dispositivos móviles”*. Estos contenidos básicos van a permitir desarrollar las capacidades y competencias del alumnado.

Estos contenidos están relacionados con el resultado de aprendizaje mencionado en el apartado anterior, así como con los criterios de evaluación que se detallaran en posteriores apartados.

- Entornos de desarrollo para juegos.
- Integración del motor de juegos en entornos de desarrollo.
- Conceptos avanzados de programación 3D.
- Fases de desarrollo.
- Propiedades de los objetos, luz, texturas, reflejos, sombras.
- Aplicación de las funciones del motor gráfico. Renderización.
- Aplicación de las funciones del grafo de escena. Tipos de nodos y su utilización.
- Análisis de ejecución. Optimización del código.

IV.3.11 Objetivos didácticos

Los objetivos didácticos que debe alcanzar el alumno son:

- Conocer los entornos de desarrollo para juegos
- Integrar el motor de juegos en entornos de desarrollo
- Utilizar conceptos avanzados de programación 3D
- Conocer las fases del desarrollo de los videojuegos.
- Aplicar propiedades a los objetos, luz, texturas, reflejos, sombras
- Aplicar las funciones del motor gráfico

- Aplicar las funciones del grafo de escena
- Analizar y optimizar el código

IV.3.12 Metodología

La metodología didáctica es el conjunto de iniciativas que se llevan a cabo con el objetivo de guiar en el proceso de enseñanza-aprendizaje en el aula, siempre con el objetivo de alcanzar los Resultados de Aprendizaje de la Unidad Didáctica *“Desarrollo de juegos para dispositivos móviles”*

La metodología que se menciona para esta Unidad Didáctica tiene como finalidad preparar al alumnado para la actividad profesional, personal y social. Es por ello, por lo que, la metodología a seguir tiene principalmente un enfoque práctico con el objetivo de motivar al alumnado, partiendo de sus intereses, necesidades y expectativas, así como de sus conocimientos previos.

La metodología debe partir de la perspectiva del profesorado como orientador, promotor y facilitador del desarrollo competencial en el alumnado. Uno de los elementos fundamentales en la enseñanza por competencias es despertar y mantener la motivación hacia el aprendizaje en el alumnado, lo que implica un nuevo planteamiento de su papel, más activo y autónomo, consciente de ser el responsable de su aprendizaje, a tal fin, el profesorado ha de ser capaz de generar en él la curiosidad y la necesidad por adquirir los conocimientos, las destrezas y las actitudes y valores presentes en las competencias

En resumen, desde un enfoque basado en la adquisición de las competencias clave cuyo objetivo no es solo saber, sino saber aplicar lo que se sabe y hacerlo en diferentes contextos y situaciones, se precisan distintas estrategias metodológicas, entre las que resaltaremos las siguientes:

- Plantear diferentes situaciones de aprendizaje que permitan al alumnado el desarrollo de distintos procesos cognitivos: analizar, identificar, establecer diferencias y semejanzas, reconocer, localizar, aplicar, resolver, etc.
- Potenciar en el alumnado la autonomía, la creatividad, la reflexión y el espíritu crítico.
- Contextualizar los aprendizajes de tal forma que el alumnado aplique sus conocimientos, habilidades, destrezas o actitudes más allá de los contenidos propios del módulo y sea capaz de transferir sus aprendizajes a contextos distintos del escolar.
- Potenciar en el alumnado procesos de aprendizaje autónomo, en los que sea capaz, desde el conocimiento de las características de su propio aprendizaje, de fijarse sus propios objetivos, plantearse interrogantes. organizar y planificar su

trabajo, buscar y seleccionar la información necesaria, ejecutar el desarrollo, comprobar y contrastar los resultados y evaluar con rigor su propio proceso de aprendizaje.

- Fomentar una metodología experiencial e investigativa, en la que el alumnado desde el conocimiento adquirido se formule hipótesis en relación con los problemas planteados e incluso compruebe los resultados de las mismas.
- Utilizar distintas fuentes de información (directas, bibliográficas, de Internet, etc.) así como diversificar los materiales y los recursos didácticos que utilicemos para el desarrollo y la adquisición de los aprendizajes del alumnado.
- Promover el trabajo colaborativo, la aceptación mutua y la empatía como elementos que enriquecen el aprendizaje y nos forman como futuros ciudadanos de una sociedad cuya característica principal es la pluralidad y la heterogeneidad. Además, nos ayudará a ver que se puede aprender no solo del profesorado, sino también de quienes nos rodean, para lo que se deben fomentar las tutorías entre iguales, así como procesos colaborativos, de interacción y deliberativos, basados siempre en el respeto y la solidaridad.

IV.3.12.1 Descripción del aula

Las clases serán impartidas en su totalidad en el aula de informática. Cada alumno dispondrá de un ordenador con los programas necesarios para el desarrollo de las clases instalado en el ordenador, así como de cuadernos si los necesitan. El aula tendrá dos puestos de ordenadores por cada mesa, orientadas hacia la pizarra y la mesa del profesor tal y como se muestra en la Ilustración 49.

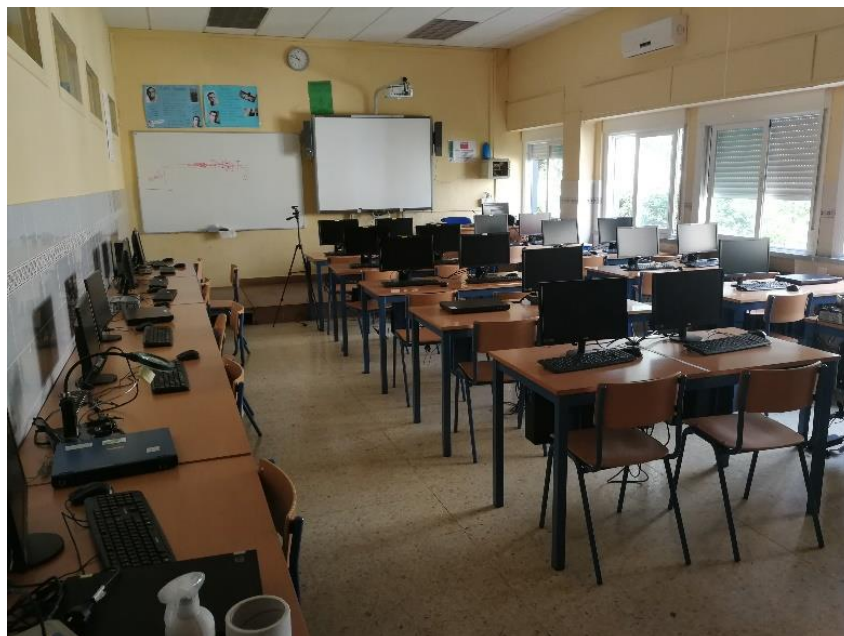


Ilustración 49. Aula informática I.E.S. Jabalcuz (Jaén). Fuente: Elaboración propia

IV.3.12.2 Tipo de Agrupación

En esta Unidad Didáctica se realizarán los siguientes agrupamientos para la realización de las actividades programadas, el número de alumnos máximo para el desarrollo de este módulo será de **18-20 alumnos**:

- **Grupo –Clase:** Se requerirá la participación y colaboración de todo el grupo para la puesta en común de ideas y opiniones.
- **Individual:** Cada estudiante trabajará de forma autónoma
- **Grupos de 2:** Este tipo de agrupamiento es adecuado para llevar a cabo las prácticas en el aula de informática. El alumnado tiene la posibilidad de comunicar, compartir y realizar los trabajos simultáneamente, contando en todo momento con la participación activa de todos los miembros del grupo. Los grupos se organizarán por parte del alumnado mayoritariamente, pudiendo el profesor organizar los grupos el profesor, en caso de ser necesario.

IV.3.12.3 Materiales y Recursos Didácticos

Para el desarrollo adecuado de esta Unidad Didáctica se requerirá el uso de una serie de recursos para cada alumno:

- **Material Clase:**
 - Ordenador con el software necesario instalado.
 - Proyector y pantalla de proyección
 - Pizarra de rotulador
- **Software**
 - *Unreal Engine 4*
 - *Draw.io*
- **Software apoyo a la docencia**
 - *Google Classroom*
 - *Google Presentaciones*
- **Manuales y Guiones**
 - Temario y enlaces disponibles en *Google Classroom*
 - Guion de prácticas
 - Manuales técnicos de *Unreal Engine 4*
 - Actividades de refuerzo y ampliación proporcionadas por el profesorado si fuese necesario.

IV.3.12.4 Tipos de actividades enseñanza-aprendizaje

Se muestran las diferentes actividades de enseñanza-aprendizaje utilizadas en la metodología para el desarrollo de las sesiones correspondientes a esta Unidad Didáctica.

- **Conocimientos previos.** Realizadas para obtener información sobre lo que saben y qué procedimientos, destrezas y habilidades tienen desarrolladas los alumnos sobre un tema concreto. La estrategia más utilizada es la tormenta de ideas y preguntas.
- **Desarrollo de contenidos.** Tienen por finalidad desarrollar los contenidos propuestos para la consecución de los objetivos y la adquisición de las competencias profesionales. Se realizarán dos tipos de actividades:
 - **Clases magistrales:** El profesor desarrollará teóricamente los contenidos.
 - **Clases prácticas:** El profesor, haciendo uso de los recursos del aula y del software anteriormente mencionado, desarrollará de forma práctica los contenidos.
- **Actividades de resolución de problemas.** Siguiendo un guion de prácticas, cada grupo formado por 2 alumnos, deberá desarrollar los contenidos de la Unidad Didáctica.
- **Diagnóstico.** Orientadas a comprobar el proceso de aprendizaje del alumnado durante el proceso, con el fin de contrastar si está asimilando los contenidos trabajados de manera correcta o si es necesario que trabaje más en profundidad algunos aspectos.
- **Evaluación.** Tienen por objeto la valoración del proceso de aprendizaje del alumno a través de preguntas orales/escritas, trabajos individuales o en grupo,... sobre los contenidos y actividades trabajadas a lo largo de la Unidad Didáctica. Tendrán como referentes los criterios de evaluación y se realizarán con los procedimientos de evaluación.
- **Recuperación.** Son las que se programan para el alumnado que no haya alcanzado los objetivos planteados a través de los contenidos trabajados.

IV.3.12.5 Metodología General

Se detalla la metodología a seguir en situaciones normales sin la situación excepcional de la pandemia de COVID-19, suponiendo un escenario de total presencialidad por parte del alumnado y del profesor.

Se presenta una tabla resumen (Tabla 2) con el tipo de actividades de enseñanza-aprendizaje utilizadas así como un tiempo estimado, tipo de agrupamiento y recursos utilizados. El desarrollo de todas las actividades se realizará en el aula asignada.

Actividad E-A	Tiempo estimado (min)	Agrupamiento	Recursos
Conocimientos previos. Tormenta de ideas y preguntas sobre desarrollo de juegos móviles	15	Grupo-Clase	Exposición oral
Clases magistrales. Desarrollo teórico de los contenidos. - Fases de desarrollo de un videojuego - Optimización de código y recursos artísticos para juegos móviles	105	Individual	Google Presentaciones Pizarra Ordenador
Clases prácticas. Repaso práctico del motor e introducción de nuevos conceptos. - Configuración del proyecto para desarrollo de juegos móviles - Introducción al desarrollo de juegos móviles - Navegación por Unreal Engine 4 - Manipulación de objetos - Blueprints - Interacción usuario-juego: Mando, teclado y móvil (Tap y Swipe) - Introducción de sonido y efectos especiales básicos - Cambio de nivel - Introducción a interfaces gráficas (Widgets) - Exportación de juegos a distintas plataformas (Windows, Web, Android)	100	Individual	Unreal Engine 4 Google Presentaciones Pizarra Ordenador
Actividades de resolución de problemas. Casos prácticos para aplicar la teoría impartida por el profesor. A realizar por el alumnado. Guion de prácticas para desarrollarlo en Unreal Engine 4. Desarrollo de un videojuego para plataformas móviles	300	Grupos de 2	Unreal Engine 4 Draw.io Papel Ordenador
Diagnóstico. Resolución de dudas. Realización de ejercicios con dudas	30	Grupo-Clase	Unreal Engine 4 Pizarra Ordenador
Evaluación. Defensa oral de las prácticas	50	Grupos de 2	Oral

Tabla 2. Tabla resumen. Tipos de actividades enseñanza-aprendizaje. Fuente: Elaboración propia

En la tabla (Tabla 3) se presentan las actividades que se han programado para esta Unidad Didáctica y su temporalización distribuidas en sesiones. Leyenda: *Numero de sesión_Orden de realización de la actividad.*

Actividad E-A	Tiempo estimado (min)	Tipo de agrupamiento
1_A Conocimientos previos. Tormenta de ideas y preguntas sobre desarrollo de juegos móviles.	15	Grupo-Clase
1_B Clases magistrales. Fases de desarrollo de un videojuego.	45	Individual
1_C Actividades de resolución de problemas. Introducción al guion de prácticas por parte del alumnado. - Descomposición del problema planteado (resumen del juego y sus diferentes niveles) y realización de un diagrama visual con Draw.io para su correcta comprensión. - Inicio con el planteamiento de los niveles en el motor así como de los blueprints necesarios a desarrollar.	60	Grupos de 2
2_A Clases prácticas Repaso práctico del motor e introducción de nuevos conceptos. - Configuración del proyecto para desarrollo de juegos móviles - Introducción al desarrollo de juegos móviles - Navegación por Unreal Engine 4 - Manipulación de objetos - Blueprints - Interacción usuario-juego: Mando, teclado y móvil (Tap y Swipe) - Introducción de sonido y efectos especiales básicos - Cambio de nivel - Introducción a interfaces gráficas (Widgets) - Exportación de juegos a distintas plataformas (Windows, Web, Android).	100	Individual
2_B Diagnóstico. Resolución de dudas. Realización de ejercicios con dudas	20	Grupo-Clase
3_A Clases magistrales. Optimización de código y recursos artísticos para juegos móviles En esta actividad se tratará el tema transversal relativo al Patrimonio cultural y artístico de Andalucía, utilizando recursos como imágenes de los monumentos más emblemáticos, así como reproducciones en 3D (Ilustración 50).	60	Individual
3_B Actividades de resolución de problemas Continuación del guion de prácticas. - Implementación de los primeros niveles del videojuego, introducción de recursos artísticos y musicales. - Continuación del desarrollo de los primeros blueprints para la creación del funcionamiento de los primeros niveles.	60	Grupos de 2
4_A Actividades de resolución de problemas	120	Grupos de 2

Continuación del guion de prácticas. Desarrollo de los niveles finales del videojuego así como la creación del archivo ejecutable del juego en formato .APK		
5_A Actividades de resolución de problemas Continuación del guion de prácticas. Toma de imágenes del desarrollo del juego, así como de los Blueprints más destacados.	60	Grupos de 2
5_B Diagnóstico. Resolución de dudas. Realización de ejercicios con dudas	10	Grupo-Clase
5_C Evaluación Defensa oral de las prácticas.	50	Grupos de 2

Tabla 3. Tabla temporalización de actividades. Fuente: Elaboración propia

IV.3.12.6 Adaptación al COVID

Precisiones a tener en cuenta por COVID-19 (y en caso de confinamiento):

- Asistencia obligatoria a clases online (*Google Meet*)
- Cada alumno/a deberá realizar la instalación del programa gratuito que se va a utilizar para la realización de las prácticas (*Unreal Engine 4*), proporcionándoles una guía de dicha instalación.
- La defensa oral de los grupos deberá realizarse a través de *Google Meet* utilizando una presentación de Google.

IV.3.13 Evaluación y Recuperación

La evaluación proporciona al profesor datos para conocer el grado de progreso alcanzado por los estudiantes en relación a los objetivos propuestos, así como determinar si la enseñanza ha sido adecuada o no.

IV.3.13.1 Evaluación y Recuperación General

Se presenta una tabla resumen (Tabla 4) con las tareas y sus correspondientes criterios de calificación, así como el instrumento utilizado para evaluarlo y el momento temporal dentro de la UD en la que se evalúa.

Tabla resumen de evaluación			
Tarea	Observación	Ejercicios prácticos	Defensa oral
Criterios de calificación	5%	75%	20%
Instrumento de evaluación	Cuaderno del profesor	Guion de prácticas	Rúbrica
Temporalización	Durante todo el periodo docente	Intercalado entre clases magistrales y clases prácticas.	Al finalizar la UD

Tabla 4. Tabla resumen de la evaluación. Fuente: Elaboración propia

IV.3.13.1.1 Criterios de evaluación

Los criterios de evaluación de esta Unidad Didáctica para los Resultados de aprendizaje descritos en la sección Resultados de aprendizaje son:

- Se ha establecido la lógica de un nuevo juego.
- Se han creado objetos y definido los fondos
- Se han instalado y utilizado extensiones para el manejo de escenas.
- Se han utilizado instrucciones gráficas para determinar las propiedades finales de la superficie de un objeto o imagen.
- Se ha incorporado sonido a los diferentes eventos del juego.
- Se han desarrollado e implantado juegos para dispositivos móviles.
- Se han realizado pruebas de funcionamiento y optimización de los juegos desarrollados.
- Se han documentado las fases de diseño y desarrollo de los juegos creados.

IV.3.13.1.2 Procedimientos e instrumentos

Las técnicas utilizadas serán las siguientes:

1. Observación directa: A través de esta técnica el profesor percibirá las habilidades conceptuales, procedimentales y actitudinales del estudiante de forma detallada y permanente, con el propósito de orientarlos cuando lo requiera el proceso de enseñanza aprendizaje. El profesor tomará información de la participación en clase del alumnado, su actitud respecto al trabajo de grupo, cuidado del material, uso del material para lo que está destinado, etc.

2. Ejercicios prácticos: Se emplearán para evaluar los conocimientos que el estudiante va adquiriendo. Utilizando el motor de videojuegos *Unreal Engine 4*, los alumnos, en grupos de dos, deberán realizar un guion de prácticas para el desarrollo de un juego para dispositivos móviles, que ambos entregar de manera individual. Dicho guion constará de tres apartados:

- Diagrama visual donde se resuma el juego y sus diferentes niveles.
- Documento memoria con imágenes del desarrollo del juego (*Blueprints* más destacados e imágenes de los niveles en el editor de *Unreal Engine*.).
- Archivo del juego en formato *APK*

3. Defensa oral: Se emplea para evaluar el conocimiento adquirido a lo largo de la UD, necesitando el alumno de los conocimientos adquiridos en la parte práctica. Se realizará por ambos miembros del grupo, utilizando los documentos entregados en los ejercicios prácticos: Diagrama visual, documento memoria o archivo del juego. La defensa oral de cada grupo durará como máximo 5 min.

Se muestra una tabla resumen (Tabla 5) con las técnicas e instrumentos utilizados

Técnicas	Instrumentos
Observación	Cuaderno del profesor
Ejercicios prácticos	Guion de prácticas
Defensa oral	Rúbrica

Tabla 5. Tabla resumen técnicas e instrumentos. Fuente: Elaboración propia

Los instrumentos utilizados serán los siguientes:

- **Cuaderno del profesor:** En este cuaderno el profesor anotará todo lo que estime oportuno para realizar una evaluación objetiva (realización de trabajos en clase, seguimiento del trabajo en casa, atención y participación en clase...). Las anotaciones, serán usadas para aumentar en 0.2 puntos cada petición que el profesor haga en clase, hasta alcanzar la máxima nota. Igualmente, los comportamientos no adecuados restarán la misma calificación
- **Rúbrica:** Rúbrica elaborada para la evaluación de la defensa oral de la parte práctica por parte del alumnado. Nota: 0.1-10 puntos.
- **Guion de prácticas:** Se incluirán los siguientes contenidos:
 - Realización del diagrama visual. 20%
 - Documento memoria. 60%
 - Archivo del juego en formato *APK*. 20%

IV.3.13.1.3 Criterios de Calificación

Tomando de referencia la tabla resumen mencionada anteriormente (Tabla 4), la nota de la evaluación de la UD se realizará realizando la **media ponderada** de cada una de las actividades realizadas de acuerdo con los criterios de calificación en base a tareas anteriormente descritas. Para realizar dicha media, el alumno deberá **entregar todos los documentos requeridos en el guion de prácticas para su posterior defensa oral**. La UD se dará por superada si se obtiene una puntuación ≥ 5 , sobre 10. En caso contrario, se

considera negativa y por tanto no superada. En este caso el alumno/a debe recuperar la UD.

Nota = 75% ejercicios prácticos + 20% defensa oral + 5% observación

IV.3.13.1.4 Recuperación

La **recuperación** de la UD, está dirigida al alumnado que no haya alcanzado los objetivos mínimos y constará de la entrega del guion de prácticas, en caso de no tener superada esa actividad, y de la defensa oral de la misma. En caso de no tener superada la defensa oral, deberá volver a defender la parte práctica. La observación no se podrá tener en cuenta en esta fase por imposibilidad de realización, manteniendo la nota obtenida durante la evaluación ordinaria. Para superar la recuperación, como en la evaluación ordinaria, el alumno/a **deberá entregar todos los documentos requeridos en el guion de prácticas y realizar la defensa oral.**

IV.3.13.2 Evaluación y Recuperación COVID

Se mantiene el mismo sistema de evaluación y calificación que en situaciones de presencialidad.

La defensa oral se realizará en caso necesario, vía *Google Meet*, si la situación así lo requiriese.

IV.3.14 Transversalidad

En general, las Tecnologías de Información y Comunicación, marco general de la familia profesional Informática y Comunicaciones, tienen un ámbito de aplicación multidisciplinar que permite contextualizar el proceso de enseñanza-aprendizaje a contenidos de otras materias, a temáticas relativas al patrimonio de Andalucía (Ilustración 50), a los elementos transversales del currículo, o a la especialización del alumnado, propia del Grado Superior de la Formación Profesional, mediante el uso de aplicaciones y herramientas informáticas.



Ilustración 50: Catedral de Jaén en 3D Unreal Engine. Fuente: Elaboración propia

Por último, desde el módulo de Programación multimedia y dispositivos móviles, se debe promover un clima de respeto, convivencia y tolerancia en el ámbito de la comunicación digital, prestando especial atención a cualquier forma de acoso, rechazo o violencia; fomentar una utilización crítica, responsable, segura y autocontrolada en su uso; incentivar la utilización de herramientas de software libre; minimizar el riesgo de brecha digital debida tanto a cuestiones geográficas como socioeconómicas o de género; y a perfeccionar las habilidades para la comunicación interpersonal.

IV.3.15 Atención a la Diversidad

Según la legislación vigente, se debe atender la diversidad en el aula y, por tanto, hay que tener en cuenta los distintos tiempos de aprendizaje, así como, la diversidad de intereses, motivaciones y capacidades de todo el alumnado.

Esta propuesta promueve la inclusión educativa y el principio de no discriminación, procurando educar al alumnado en la solidaridad, sentido crítico y responsabilidad. Para ello, el alumnado requiere un currículo flexible que permita atender la diversidad del alumnado, en muchas ocasiones marcado por desventajas culturales, sociales, económicas y/o familiares.

Actualmente, en las escuelas y centros educativos nos encontramos con alumnos/as que proceden de otros países. Es muy habitual encontrar en nuestras zonas alumnado procedente de las zonas de Marruecos y norte de África, cuyas familias y ellos mismos, se encuentran en España desarrollando trabajos agrarios. El alumno puede hablar tanto árabe, francés y algunos poseer un nivel básico de inglés o castellano.

Para atender esta diversidad se tomarán las siguientes medidas:

1. La guía de las Prácticas Evaluables será redactadas tanto en español como en inglés o inclusive en su idioma nativo si las herramientas tecnológicas así lo permitieran. Asimismo, se le proporcionarán los manuales y temario en este idioma.
2. El alumno puede utilizar un traductor si lo considerase necesario.
3. En un primer momento, se cambiará la configuración del idioma en el ordenador, así como de las herramientas y aplicaciones que se puedan utilizar. La herramienta principal para el desarrollo de las prácticas es *Unreal Engine* y cuenta con un sistema completo de traducciones a múltiples idiomas.
4. Cuando se realicen los trabajos grupales, el alumno será incluido en un grupo con alumnos que tengan un buen nivel de inglés o francés para fomentar la fluidez en la comunicación en aquellos momentos en los que no pueda hacerse en castellano.
5. Se animará al alumno a realizar la defensa oral en español, siempre y cuando no suponga una situación comprometida. En tal caso, se le indicará al alumno que

realice la defensa en inglés o francés. El objetivo es que se sienta integrado en el grupo y que al expresarse en su lengua materna se sienta motivado.

IV.3.16 Anexos

IV.3.16.1 Guion de prácticas: Desarrollo de juego móvil

Realizar el desarrollo de un juego móvil usando el motor de videojuegos *Unreal Engine*.

Se realizará un juego en primera persona en el que el jugador deberá desactivar un explosivo. Cada vez que se alcanza un nivel, el método de desactivación del explosivo irá variando.

Los niveles de los que estará compuesto el juego se detallarán a continuación.

En todos los niveles existirán las siguientes condiciones (menos en el último nivel):

- **Condición de victoria:** El jugador logra desactivar el explosivo con el método de desactivación correcto.
- **Condición de derrota:** El explosivo explota. En caso de derrota el juego mostrará un menú en el cual el jugador deberá elegir entre volver al menú principal o reiniciar el nivel.

Nota: La sonorización, efectos visuales, ambientación y los recursos artísticos empleados dependerán de cada alumno.

IV.3.16.1.1 Nivel 0. Menú principal

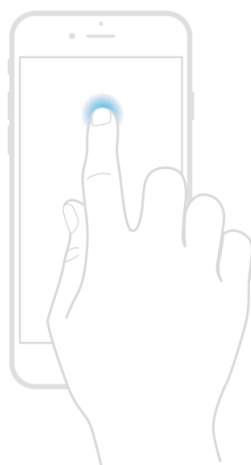
Menú del juego en el que aparecerán los distintos subniveles para que el jugador pueda escoger el nivel correspondiente.

Sugerencia de mejora: Hasta que no se completa un nivel, el jugador no puede avanzar.

IV.3.16.1.2 Nivel 1. Tap

El jugador dispondrá de un tiempo T para desactivar el explosivo. Un contador debe mostrar por pantalla el tiempo que le queda al jugador antes de la explosión.

Método para la desactivación: El jugador debe tocar (*Tap*) un número de veces la pantalla (x veces). El número de toques restantes para realizar la desactivación debe aparecer en pantalla.



Tap
(~0.1 second)

Ilustración 51. Gesto de Tap. Fuente: (Apple, n.d.)

IV.3.16.1.3 Nivel 2. Swipe

El jugador dispondrá de un tiempo T para desactivar el explosivo. Un contador debe mostrar por pantalla el tiempo que le queda al jugador antes de la explosión.

Método para la desactivación: El jugador debe hacer *swipe* un número de veces (x veces). El número de *swipes* restantes para realizar la desactivación debe aparecer en pantalla.

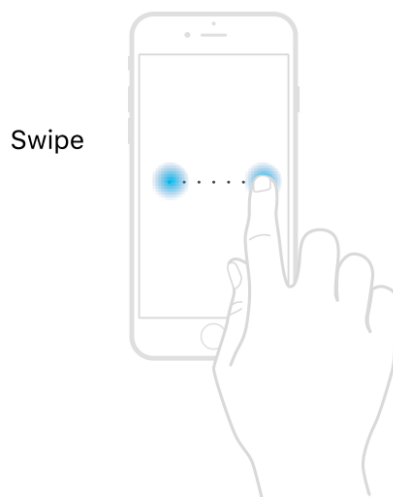


Ilustración 52. Gesto de Swipe. Fuente: (Apple, n.d.)

IV.3.16.1.4 Nivel 3. Patrones

El jugador dispondrá de un tiempo T para desactivar el explosivo. Un contador debe mostrar por pantalla el tiempo que le queda al jugador antes de la explosión.

Método para la desactivación: Aleatoriamente aparecen por pantalla distintos patrones o gestos (*Tap o Swipe*). Hasta que el jugador no realiza la acción correctamente el patrón no cambia. El jugador debe realizar un número de patrones (x patrones) antes de la

explosión. El número de patrones restantes para realizar la desactivación debe aparecer en pantalla.

Si el jugador no realiza todos los patrones antes de que el tiempo se acabe, el jugador será derrotado.

Sugerencia de mejora: Los patrones *swipe* pueden realizarse siguiendo varias direcciones:

- De derecha a izquierda
- De izquierda a derecha
- De arriba hacia abajo
- De abajo hacia arriba

IV.3.16.1.5 Nivel 4. Nivel infinito

No hay ningún tiempo de desactivación del explosivo

Por pantalla mostrar un contador del número de patrones correctos que ha realizado el jugador. También mostrar el número de intentos restante.

El objetivo de este nivel es intentar conseguir el máximo número de patrones correctos (*Tap o Swipe*) antes de que los intentos lleguen a 0.

Por pantalla deberá aparecer un patrón que debe realizar el jugador. Dicho patrón se mantendrá por pantalla durante t tiempo. Pasado dicho tiempo si el jugador no ha realizado la acción, se descontará un intento. Si el jugador ha realizado la acción correctamente, el contador de patrones correctos incrementará en uno su valor y aparecerá otro patrón (de manera aleatoria).

IV.3.16.1.6 Distribución y exportación a dispositivo móvil

Creación de un paquete con formato *APK* para la visualización de dispositivos móviles.

Nota: La exportación se realiza en *File > Package Project > Android*

IV.3.16.1.7 Entrega:

Se deberá entregar una carpeta con el siguiente contenido:

- Archivo del juego *APK*
- Documento memoria con imágenes del desarrollo del juego (*Blueprints* más destacados e imágenes de los niveles en el editor de Unreal Engine.). Si se han utilizado recursos externos (arte o sonidos) deberá aparecer el nombre de su creador para reconocer la autoría de los mismos.
- **Diagrama visual** donde se resuma el juego y sus diferentes niveles. (Nota: Se puede realizar dicho diagrama utilizando la extensión de Google Draw.io (Draw io, s.f.)

La carpeta deberá ser comprimida en formato **ZIP**.

Nombre de la carpeta: *Apellido1Apellido2Nombre.zip*

IV.3.16.2 Rubrica

Indicador	Excelente (2 p)	Muy bien (1.5 p)	Bien (1 p)	Regular (0.5 p)	Mal (0.1 p)	No presentado
Asimilación o análisis del problema	El alumno/a ha comprendido completamente el problema, relacionándolo con otros problemas vistos en clase.	Entiende el problema de manera literal, sin ser capaz de relacionarlo con otros contenidos del temario	Entiende el problema con dificultad	Entiende un problema que tiene poco que ver con el ejercicio planteado	No se ha comprendido el problema	No ha realizado la evaluación
Desarrollo de la solución	El alumno/a desarrolla una solución al problema óptima, eficiente, y más allá de lo que se exige	Se resuelve el problema. La solución no es la mejor, pero es una Solución funcional.	Desarrolla una solución con errores leves	Se desarrolla una solución que no funciona pero relativa al problema planteado, o desarrolla una solución funcional pero del problema analizado incorrectamente	No se ha desarrollado ninguna solución funcional al problema planteado	No ha realizado la evaluación
Exposición del resultado	El alumno/a es capaz de presentar el resultado de manera clara, y concisa. Además, es capaz de comentar y criticar posibles soluciones alternativas.	Se presenta la solución de manera clara, aunque con algunas deficiencias en la explicación.	Presenta el resultado de manera confusa.	Se presenta el resultado de manera confusa, y poco clara, con indicios de copia, sin analizar la solución.	No es capaz de transmitir una opinión sobre el problema. Es una copia.	No ha realizado la evaluación
Creatividad	El alumno/a ha presentado una solución creativa fuera de la explicada por el profesor	El alumno/a es capaz de presentar una solución dentro de las esperadas	No hay creatividad en la solución, pero hay un intento de resolución correcta.	No hay creatividad en la solución, pero hay un intento de resolución aunque no es la adecuada.	No se ha dado solución al problema	No ha realizado la evaluación
Preguntas	Puede contestar con precisión a casi todas las preguntas planteadas por el profesor.	Puede contestar con precisión a la mayoría de las preguntas planteadas por el profesor.	Puede contestar con precisión a unas pocas preguntas planteadas por el profesor.	Puede contestar con poca precisión a preguntas planteadas por el profesor.	No puede contestar las preguntas planteadas por el profesor.	No ha realizado la evaluación

Tabla 6: Rubrica. Fuente: Elaboración propia

V. Índice de Tablas

Tabla 1. Calidad del material UE4. Fuente: Elaboración propia	44
Tabla 2. Tabla resumen. Tipos de actividades enseñanza-aprendizaje. Fuente: Elaboración propia	74
Tabla 3. Tabla temporalización de actividades. Fuente: Elaboración propia	76
Tabla 4. Tabla resumen de la evaluación. Fuente: Elaboración propia	77
Tabla 5. Tabla resumen técnicas e instrumentos. Fuente: Elaboración propia	78
Tabla 6: Rubrica. Fuente: Elaboración propia.....	84

VI. Índice de ilustraciones

Ilustración 1: Tennis for Two. Fuente: (The History of Video Games, s.f.)	11
Ilustración 2: Pac-Man. Fuente: (Espacio Psicofamiliar, s.f.)	12
Ilustración 3: Consolas de Sony hasta la PS4. Fuente: (Alvarez, s.f.)	15
Ilustración 4. Clasificación PEGI. Fuente: (Fantasía Gamer, 2016)	17
Ilustración 5. Etiquetas PEGI. Fuente: (Fantasía Gamer, 2016)	17
Ilustración 6. Ejemplo de aplicación del PEGI. Fuente: (García Gigante, s.f.)	18
Ilustración 7: Graybox de Uncharted, primer prototipo. Fuente: (Schreier, s.f.)	22
Ilustración 8. Diagrama DAFO. Fuente: (Gestión de proveedores, s.f.)	26
Ilustración 9. Feria del videojuego Madrid Games Week. Fuente: Madrid Games Week	28
Ilustración 10. Consola portátil Nintendo Switch. Fuente: (Nintendo, s.f.)	31
Ilustración 11: Logo Unity. Fuente: Unity	34
Ilustración 12: Logo Unreal Engine 4. Fuente: Epic Games	34
Ilustración 13. Juegos desarrollados con Unreal Engine (de izquierda a derecha: Batman Arkham Knight, Street Fighter V, Fortnite). Fuente: (González Soto, 2018)	35
Ilustración 14. Bolt. Fuente: Unity.com	36
Ilustración 15. Blueprints. Fuente: Elaboración propia	37
Ilustración 16. Comprobación de instalación Android (1). Fuente: (Salinas, 2020)	38
Ilustración 17. Comprobación de instalación Android (2). Fuente: (Salinas, 2020)	38

Ilustración 18. Terminal de comandos. Fuente: Elaboración propia.....	39
Ilustración 19. Creación de nuevo proyecto en Unreal Engine 4. Fuente: Elaboración propia.....	40
Ilustración 20. Aceptar licencia de SDK. Fuente: Elaboración propia.....	40
Ilustración 21. Ejecución del juego en dispositivo móvil. Fuente: (Salinas, 2020)	41
Ilustración 22. Materiales Unreal Engine 4. Fuente: Documentación oficial Unreal	42
Ilustración 23. Blueprint materiales Unreal Engine 4. Fuente: (Erell, 2019).....	42
Ilustración 24. Modos de optimización. Fuente: Elaboración propia.....	43
Ilustración 25. Modos de visualización: Fuente: Documentación oficial Unreal	43
Ilustración 26. Device Profiles (1) UE4. Fuente: Elaboración propia	44
Ilustración 27. Device Profiles (2) UE4. Fuente: Elaboración propia	45
Ilustración 28. Device Profiles (3) UE4. Fuente: Elaboración propia	45
Ilustración 29. Device Profiles (4) UE4. Fuente: Elaboración propia	46
Ilustración 30. Nodo Quality Switch. Fuente: Documentación oficial Unreal	46
Ilustración 31. Blueprint material. Fuente: Documentación oficial Unreal.....	47
Ilustración 32. LODs Unreal Engine 4. Fuente: Documentación oficial Unreal	48
Ilustración 33. LOD 0. Fuente: Elaboración propia	48
Ilustración 34. LOD 1. Fuente: Elaboración propia	49
Ilustración 35. LOD 2. Fuente: Elaboración propia	50
Ilustración 36. LOD 3. Fuente: Elaboración propia	51
Ilustración 37. Exportación a dispositivos móviles. Fuente: Elaboración propia	52
Ilustración 38: Steam. Fuente: Elaboración propia	53
Ilustración 39. Itch.io. Fuente: Elaboración propia.....	53
Ilustración 40. Flight Simulator. Fuente: (Venckunas, 2020).....	55
Ilustración 41. Pokemon Go. Fuente: (Peru.com, 2016)	56
Ilustración 42. Visualización de un mueble con RA. Fuente: (Pastor, 2017)	56
Ilustración 43. Dinosaurio 3D con RA. Fuente: (Cantudo, 2020).....	57
Ilustración 44. Videojuego español exclusivo de Google Stadia, Gylt. Fuente: Tequila Works	58
Ilustración 45. Detrás de las escenas de The Mandalorian. Fuente: Reddit	59

Ilustración 46. Entrada del centro IES Jabalcuz. Fuente: Elaboración propia	63
Ilustración 47. Localización del IES Jabalcuz (Jaén). Fuente: Google Maps.....	64
Ilustración 48. Hall del centro IES Jabalcuz. Fuente: Elaboración propia	65
Ilustración 49. Aula informática I.E.S Jabalcuz (Jaén). Fuente: Elaboración propia	71
Ilustración 50: Catedral de Jaén en 3D Unreal Engine. Fuente: Elaboración propia.....	79
Ilustración 51. Gesto de Tap. Fuente: (Apple, n.d.)	82
Ilustración 52. Gesto de Swipe. Fuente: (Apple, n.d.)	82

VII. Bibliografía

- Alvarez, I. (n.d.). *Comprar.com*. Retrieved from <https://www.comprar.com/cuantas-consolas-playstation-ha-vendido-sony-entre-todas-sus-variantes/>
- Android Studio. (n.d.). *Android Studio*. Retrieved from <https://developer.android.com/studio>
- Apple. (n.d.). *Handling Swipe Gestures*. Retrieved from https://developer.apple.com/documentation/uikit/touches_presses_and_gestures/handling_uikit_gestures/handling_swipe_gestures
- BaboonLab. (n.d.). *Blueprints en Unreal Engine 4, funciones y tipos*. Retrieved from https://baboonlab.odoo.com/en_US/blog/news-1/post/blueprints-en-unreal-engine-4-funciones-y-tipos-21
- Bourne, C., & Salgado, V. (2016, Diciembre 22). *Aika Educacion*. Retrieved from <http://www.aikaeducacion.com/tendencias/los-videojuegos-transforman-aula/>
- Burón García, D., Manzano Vázquez, M., & Paredes Velasco, M. (2020). *Desarrollo de videojuegos para dispositivos móviles*. Síntesis.
- Cantudo, C. (2020, Noviembre 13). *Ser Padres*. Retrieved from <https://www.serpadres.es/familia/tiempo-libre/articulo/como-ver-dinosaurios-en-3d-en-tu-casa-el-truco-de-google-que-fascina-a-los-ninos-441605273662>
- Desarrollo Español de Videojuegos. (2020). *Libro Blanco del Desarrollo Español de Videojuegos*. Obtenido de <https://dev.org.es/images/stories/docs/libro%20blanco%20del%20desarrollo%20espanol%20de%20videojuegos%202020.pdf>
- Draw Io. (n.d.). *Extensión Draw Io*. Retrieved from <https://app.diagrams.net/>

- Epic Games. (n.d.). *Android Quick Start*. Retrieved from <https://docs.unrealengine.com/en-US/SharingAndReleasing/Mobile/Android/GettingStarted/index.html>
- Epic Games. (n.d.). *Epic Games*. Retrieved from <https://www.unrealengine.com/en-US/>
- Epic Games. (n.d.). *Mobile Performance Tips and Tricks*. Retrieved from <https://docs.unrealengine.com/en-US/SharingAndReleasing/Mobile/Performance/TipsAndTricks/index.html>
- Epic Games. (n.d.). *Performance Guidelines for Mobile Devices*. Retrieved from <https://docs.unrealengine.com/en-US/SharingAndReleasing/Mobile/Performance/index.html>
- Erell, O. (2019, Diciembre 26). *UE4 - Basic Material Blending*. Retrieved from <https://odederell3d.blog/2019/12/26/ue4-simple-blending-of-materials/>
- Espacio Psicofamiliar. (n.d.). *Pac-Man*. Retrieved from <https://espaciopsicofamiliar.es/pac-man-estimula-nuestro-cerebro-desde-1980/>
- Estallo, J. (1995). *Los videojuegos. Juicios y prejuicios*. Grupo Planeta.
- Fantasia Gamer. (2016, Diciembre 12). *¿Qué es el PEGI?* Retrieved from <https://fantasiagamer.com/2016/12/05/que-es-el-peg/>
- Fernández, Y. (2018, Julio 6). *Xataka*. Retrieved from <https://www.xataka.com/basics/diferencias-entre-realidad-aumentada-realidad-virtual-y-realidad-mixta>
- GamerDic. (n.d.). *Diccionario online de términos sobre videojuegos y cultura gamer*. Retrieved from <https://www.gamerdic.es/>
- García Gigante, B. (n.d.). *Videojuegos: Medio de ocio, cultura popular y recurso didáctico para la enseñanza y aprendizaje de las matemáticas escolares*. Retrieved from https://repositorio.uam.es/bitstream/handle/10486/3722/25737_garcia_gigante_benjamin.pdf
- Gestión de proveedores. (n.d.). *Análisis DAFO*. Retrieved from <https://sites.google.com/site/gestiondeproveedoras/analisis-dafo>
- González Soto, N. (2018). *Motores de Videojuegos*. Retrieved from <http://jeuazaru.com/wp-content/uploads/2018/11/Motores-de-Videojuegos.pdf>
- Google Stadia. (n.d.). *Google Stadia*. Retrieved from <https://stadia.google.com/?hl=es>

- Hobby Consolas. (2020, 05 13). *Hobby Consolas*. Retrieved from <https://www.hobbyconsolas.com/noticias/epic-solo-cobrara-royalties-desarrolladores-usen-unreal-engine-cuando-superen-millon-ingresos-640203>
- IES Jabalcuz. (2008). Proyecto Educativo del IES Jabalcuz. *Proyecto Educativo del IES Jabalcuz*. Jaén, Jaén, España.
- Industria Animación. (2020). *¿Qué son los Juegos AAA?* Retrieved from <https://www.industriaanimacion.com/2020/10/que-son-los-juegos-aaa/>
- Levis, D. (1997). *Los videojuegos, un fenómeno de masas*. Buenos Aires: Paidós.
- Makuch, E. (2017, Octubre 12). *Gamespot*. Retrieved from <https://www.gamespot.com/articles/fornite-battle-royale-has-hit-10-million-players-/1100-6454008/>
- Martínez Borda, R. (2006). Tesis Doctoral: El papel de los videojuegos en el desarrollo del pensamiento narrativo.
- Ministerio de Educación. (2021). *Borrador Curso de Especialización en Desarrollo de videojuegos y realidad virtual*. Retrieved from <https://www.todofp.es/dam/jcr:f971f93c-4494-4014-888d-701cbfcea29d/prd-gsuperior-videojuegos.pdf>
- Navarro Remesal, V. (2016). Tesis Doctoral: Libertad dirigida: Análisis formal del videojuego como sistema, su estructura y su avataridad.
- Ninja Theory. (n.d.). *Hellblade*. Retrieved from <https://www.hellblade.com/vertical-slice/>
- Nintendo. (s.f.). *Nintendo Switch*. Obtenido de https://www.nintendo.com/es_LA/switch/system/
- Pastor, J. (2017, Noviembre 22). *Xataka*. Retrieved from <https://www.xataka.com/realidad-virtual-aumentada/la-guerra-por-ganar-la-realidad-aumentada>
- Peru.com. (2016, Agosto 9). *Peru*. Retrieved from <https://peru.com/epic/epic-mobile/pokemon-go-que-mi-smartphone-no-tiene-realidad-aumentada-fotos-noticia-468480>
- Provenzo, E. (1991). *Video Kids: Making Sense of Nintendo*. Harvard University Press.
- Pure Electronic Games iOnline. (2019). *3 Tendencias que cambiarán el futuro de los videojuegos*. Retrieved from <https://www.pegionline.eu/es/index/id/3-tendencias-que-cambiaran-el-futuro-de-los-videojuegos/>
- Real Academia Española. (n.d.). *Definición de juego*. Retrieved from <https://dle.rae.es/juego?m=form>

- Real Academia Española. (n.d.). *Definición de videojuego*. Retrieved from <https://dle.rae.es/videojuego>
- Rogers, S. (2014). *Level Up: Guía para ser un gran diseñador de videojuegos*. Parramon.
- Sabán, A. (21 de Agosto de 2019). *Xataka*. Obtenido de <https://www.xataka.com/videojuegos/porque-no-todo-steam-nueve-plataformas-donde-tambien-puedes-comprar-juegos-para-pc-digitales>
- Salinas, J. (2020, Junio 15). *Como instalar Android en Unreal Engine*. Retrieved from <https://javiersalinas.es/instalar-android-en-unreal-engine/>
- Santamaría, P. (2019, Diciembre 16). *El Output*. Retrieved from <https://eloutput.com/noticias/cultura-geek/the-mandalorian-vfx-unreal-engine/>
- Schell, J. (2019). *The Art of Game Design*. A K Peters/CRC Press.
- Schreier, J. (n.d.). *Gizmodo*. Retrieved from <https://es.gizmodo.com/este-es-el-aspecto-que-tienen-los-niveles-de-un-videoju-1819132773>
- Statt, N. (2018, Junio 12). *The Verge*. Retrieved from <https://www.theverge.com/2018/6/12/17456814/fortnite-battle-royale-epic-games-125-million-players-first-year-e3-2018>
- TechTerms. (n.d.). *TechTerms*. Retrieved from <https://techterms.com/definition/goldenmaster>
- The History of Video Games. (n.d.). *Tennis for Two*. Retrieved from <https://videogamehistorydevelopment.weebly.com/tennis-for-two.html>
- Thier, D. (2018, Mayo 24). *Forbes*. Retrieved from <https://www.forbes.com/sites/davidthier/2018/05/24/report-fortnite-battle-royale-made-nearly-300-million-last-month/#1cbf8f63113e>
- Todo FP. (n.d.). *Ciclos Formativos de Grado Superior*. Retrieved from <https://www.todofp.es/que-como-y-donde-estudiar/que-estudiar/ciclos/grado-superior.html>
- Unity. (2020, Julio 22). *Unity Blog*. Retrieved from <https://blogs.unity3d.com/es/2020/07/22/bolt-visual-scripting-is-now-included-in-all-unity-plans/>
- Unity. (n.d.). *Bolt Visual Scripting*. Retrieved from <https://unity.com/es/products/unity-visual-scripting>
- Unity. (n.d.). *Made with Unity*. Retrieved from <https://unity.com/es/madewith>

Venckunas, V. (2020, Diciembre 27). *Aerotime*. Retrieved from <https://www.aerotime.aero/25717-microsoft-flight-simulator-is-pretty-but-how-realistic-is-it>

Vigueras, E. (2001). *Etcétera*.

Zarzecki, M. (2018, Junio 13). *Envato Tuts+*. Retrieved from https://gamedevelopment.tutsplus.com/articles/which-digital-game-distribution-platform-is-the-right-for-my-game--cms-30289?ec_unit=translation-info-language