



UNIVERSIDAD DE EXTREMADURA
FACULTAD DE CIENCIAS DE LA DOCUMENTACIÓN Y LA COMUNICACIÓN
GRADO EN COMUNICACIÓN AUDIOVISUAL

Trabajo Fin de Grado

**EVOLUCIÓN HISTÓRICA Y CONTEXTO SOCIAL DEL
VIDEOJUEGO EN EL SECTOR AUDIOVISUAL.
PREPRODUCCIÓN, PRODUCCIÓN Y POSTPRODUCCIÓN
DE UN VIDEOJUEGO A TRAVÉS DE SOFTWARE LIBRE
PROFESIONAL.**

David Sánchez Vega

REALIZADO BAJO LA TUTELA DEL PROFESOR/A:
José Luis Garralón Velasco

SEPTIEMBRE 2015

Sumario

- Introducción
- Objetivos y metodología
- Análisis
 - Historia y evolución
 - Géneros de videojuegos
 - Estudios sobre videojuegos
- Desarrollo de un videojuego
 - Preproducción
 - Producción
 - Postproducción
 - Memoria de producción
- Conclusiones
- Bibliografía
- Índice
- Anexos

1. INTRODUCCIÓN

¿Es el videojuego un nuevo medio de comunicación? La creciente presencia de los videojuegos en la sociedad es una realidad patente hoy en día. Tal es así que, a nivel económico, el sector de los videojuegos es un mercado en alza. Sin embargo, todavía es un concepto ajeno a algunos sectores de la población, y hay mucha gente a la que le resulta extraña la idea de que los videojuegos puedan considerarse un medio de comunicación, e incluso llegan a negar esta posibilidad. Técnicamente, un videojuego no es un medio de comunicación, pero sí es una obra audiovisual. El título de medio de comunicación le correspondería a las consolas, ordenadores, móviles, tablets o cualquier dispositivo mediante el cual se pueda jugar a un videojuego. Sin embargo, de la misma forma que hablamos del medio “cine” para referirnos a las películas (otro tipo de obra audiovisual), hablaremos de medio cuando tratemos los videojuegos, ya que no presentan un único medio de comunicación en el que se ejecuten, y sea cual sea, aunque presenten diferencias entre sí, comparten los mismo elementos esenciales a la hora de ejecutar el juego, elementos que a su vez definen a los videojuegos como obra audiovisual.

Todos los medios de comunicación tienen una definición común que lo caracteriza como tal y les otorga esa categoría. Transmiten un mensaje elaborado por un emisor, plasmado en el medio a través de un determinado código y dirigido a uno o más receptores. Emisor, código, receptor y mensaje, todos estos elementos están presentes en cualquier medio de comunicación: televisión, radio, prensa, Internet, cine... y, por supuesto, también el videojuego. A estos elementos hay que sumar otro factor que cada vez está más extendido en los mass media: la interactividad. Actualmente todos, o casi todos los medios, presentan interactividad de una forma u otra. Se ha extendido como un valor añadido, una forma de competir con las empresas rivales. Sin embargo, sólo hay dos medios cuya interactividad es innata: Internet y los videojuegos.

Este trabajo servirá para adentrarnos en el contexto de la cultura del videojuego, así como iniciarnos en el proceso de producción de un videojuego.

El trabajo presenta una doble naturaleza, según la normativa sobre el Trabajo de Fin de Grado: por un lado, se trata de un proyecto de naturaleza investigativa en el ámbito de la titulación; por otro lado, es un trabajo artístico.

2. OBJETIVOS Y METODOLOGÍA

Objetivos:

- Sistematizar un criterio de catalogación de tipos de videojuegos.
- Objetivar el desarrollo del videojuego desde los puntos de vista histórico y tecnológico.
- Establecer la metodología necesaria para desarrollar un videojuego, así como los conocimientos mínimos necesarios de programación y de manejo de software orientado a este fin.
- Facilitar la inserción de los videojuegos dentro del sector de la comunicación audiovisual.

Metodología

- Búsqueda de información a través de fuentes bibliográficas y webgráficas acerca de la historia, de los distintos géneros y de diversos estudios sobre videojuegos, así como del proceso de producción de un videojuego.
- Formación de un equipo de trabajo para el desarrollo de un videojuego.
- Empleo del motor gráfico Unity.
- Iniciación en la labor de programación a través de diversos tutoriales y guías.
- Desarrollo completo de un videojuego (demo).

3. ANÁLISIS TEÓRICO

3.1. Historia y evolución

3.1.1. Los orígenes

A la hora de hablar del origen de los videojuegos la mayoría de la gente piensa en el “Pong” como el primero de todos, pero hubo algunos antes de éste que, si bien presentan polémica sobre si son realmente videojuegos o no, es indudable que fueron el origen de todo lo que vino después. En una relación no exhaustiva, indico los siguientes:

- **Lanzamiento de misiles (1947):** Consiste en la adaptación de una pantalla de radar con válvulas proyectadas sobre una pantalla de rayos catódicos que era capaz de calcular una curva de lanzamiento de misiles hacia objetivos virtuales. No presentaba movimiento y los objetivos estaban sobreimpresionados, por lo que no se considera un videojuego, sino, más bien, el “Primer Experimento Electrónico de simulación en Pantalla”.
- **OXO (1952):** Versión del “Tres en Raya” creada para la computadora EDSAC. El jugador transmitía órdenes a través de un dial telefónico y en función de sus movimientos OXO tomaba decisiones. Hay quienes consideran a éste el primer videojuego, pero otros lo niegan debido a la ausencia de movimiento.
- **Tennis For Two (1958):** Creado por William Higinbotham a partir de un osciloscopio. El juego consiste en una línea horizontal que representaba el suelo, una pequeña línea vertical en el centro para representar la red y un punto de oscilación (la bola) a la que los jugadores debían dar un ángulo y golpearla. Este es el primer videojuego con animación y que permite la interacción entre dos personas (multijugador). A pesar del enorme éxito que tuvo el juego, Higinbotham nunca llegó a patentarlo, lo que le hubiera convertido en una de las personas más ricas del mundo.
- **Spacewar (1961):** Fue programado por Stephen Russell para el ordenador PDP-1 del Instituto Tecnológico de Massachusetts (MIT) y consistía en un duelo entre dos naves (cada una controlada por un jugador) que se disparaban mutuamente. Al mismo tiempo sus movimientos se veían afectados por la gravedad de una estrella que destruía las naves si se acercaban demasiado. Tuvo mucho éxito en el ámbito de las universidades, pero fuera de éste era desconocido.

- **Galaxy Game (1971):** Creado por Bill Pitts y Hugh Tuck mediante una reprogramación de “Spacewar”, supone el primer videojuego comercial de la historia, ya que contaba con “coin-door”, el mecanismo por el que se introducen las monedas que permiten al jugador comenzar una partida.
- **Computer Space (1971):** Fue la primera máquina en ser producida en serie, lo que permitió impulsar la comercialización masiva de videojuegos y con ello abandonar la carga experimental que tenía hasta entonces el desarrollo de estas máquinas. “Computer Space” fue creado por Nolan Bushnell, quien posteriormente fundaría Atari y daría vida al famoso “Pong”.

En mi opinión personal, el primer videojuego fue el OXO, ya que cumple con el concepto básico de un videojuego: la interactividad entre jugador y máquina. Sin embargo, al no existir un acuerdo en cuanto a cuál es el primer videojuego, ésta es sólo una opinión basada en mi propio criterio y cualquiera que lea este documento es libre de estar en desacuerdo.

En el Anexo 1 se incluye una colección de imágenes sobre las consolas y los videojuegos más relevantes a lo largo de cada una de las etapas de la historia que veremos a continuación (incluida la anterior). Esto servirá como complemento a lo que se explique a continuación, y permitirá ver la evolución de los videojuegos desde sus orígenes hasta la actualidad.

3.1.2. Década de los 70

En 1972 nace la primera consola, “Magnavox Odyssey” (también conocida como “Brown Box” por su aspecto simulado de madera), de la mano de quien para muchos fue el padre de los videojuegos: Ralph Baer. A pesar del éxito que tuvo la consola, no se llegó a vender muy bien debido a diversos errores de marketing, como su alto precio o la errónea interpretación que se dio a la publicidad, entendiéndolo el público que la consola sólo se podía usar en televisores Magnavox cuando en realidad funcionaba en cualquier televisor.

A Odyssey le surgió también otro problema: en el mismo año Nolan Bushnell creó la empresa Atari y junto a Al Alcorn desarrolló el videojuego “Pong”. Consistía en dos palas (cada una controlada por un jugador) y una bola que se movía de un lado a otro. La bola aumentaba su velocidad progresivamente y rebotaba con un ángulo distinto en función del punto en que diera en las palas, lo cual lo hacía más interesante. El objetivo del juego

era llevar la bola al extremo de la pantalla del lado del rival para marcar puntos (similar a una partida de air hockey).

Fue entonces cuando Atari recibió una denuncia de Magnavox, pues el Pong era muy similar a uno de los juegos de Odyssey, y Ralph Baer tenía la patente de la consola y del juego desde 1968. Magnavox y Atari llegaron a un acuerdo, pero fue Atari quien salió ganando, tanto económicamente como en el aspecto del prestigio.

En 1976 surgió por primera vez la polémica de la violencia en los videojuegos con la aparición de Death Race, de Exidy. Originalmente su nombre fue Destruction Derby y en él los jugadores debían chocarse contra los otros coches del circuito para destruirlos. Se realizaron cambios, y finalmente, en Death Race el objetivo era chocar contra esqueletos que se escapaban del cementerio. Muchos salones recreativos se negaron a exponer el juego y Exidy sólo vendió 1000 máquinas.

Ese mismo año aparecen otras compañías, como Fairchild, que lanzó la consola Channel F. Aunque no tuvo mucho éxito, fue la primera consola en usar juegos en cartucho, y desde ese momento las demás compañías empezaron a lanzar consolas con este sistema.

Atari quiso aprovechar esto y desarrollaron una nueva consola, la VCS (Video Computer System), más potente que Channel F, que salió al mercado en octubre de 1977 y cuyos mandos, además de conservar los sensores para jugar a los juegos tipo Pong, presentaban un nuevo dispositivo, el joystick, una palanca con varios botones. El negocio de mantener los juegos en cartuchos daba mucha más libertad a la hora de crear juegos diferentes y no tan limitados como los de la competencia. Sin embargo, Bushnell tenía problemas económicos, así que se vio obligado a vender la empresa a Warner Communications. En 1978 Bushnell se va de Atari por discrepancias con los nuevos directivos. En este año Taito lanza "Space Invaders".

También en 1978 aparece el primer "easter egg" (o "huevo de pascua"), consistente en una sala oculta en la que aparecía el nombre del programador del juego. Los easter eggs son curiosidades o elementos ocultos dentro de un juego. En este caso se trata del juego "Adventure" de Atari.

En 1979 Atari lanza “Asteroids”, uno de los más importantes éxitos de la historia de los videojuegos en Estados Unidos. Este año también apareció “Pac-Man” de la mano de Namco, y se fundaron las empresas Capcom y Activision.

3.1.3. Década de los 80

Atari comienza la década de los 80 licenciando “Space Invaders”, lo que le granjeó unos 100 millones de dólares y una gran ventaja sobre la competencia. Hicieron una conversión que permitía 2 jugadores e incluía 112 modos de juego.

En 1980 Nintendo crea en Nueva York “Nintendo of America Inc.”, y comienza a distribuir sus consolas portátiles “Game & Watch”, con juegos como Flagman, Vermin o Judge (más adelante aparecerían más juegos, siendo destacables “Donkey Kong”, The Legend of Zelda”, “Mario Bros.”, “Mickey Mouse” y “Balloon Fight”, entre otros).

En este año llega a Estados Unidos la consola “Intellivision” de Mattel (a Japón y Europa llegaría dos años más tarde). Era una consola de 16 bits, y fue la primera en sustituir los joysticks por mandos con botones. También tenía por separado un sintetizador de voz. Durante este año vendió 175.000 unidades y tuvo 19 juegos.

En 1982 se lanzaron al mercado algunas consolas: “Commodore 64” (de Commodore), “Arcadia 2001” (de Emerson), “Colecovision” (de Coleco), “Atari 5200” (de Atari), “ZX Spectrum” (de Sinclair) y “Vectrex”. También se crean las empresas “Electronic Arts” (con el nombre “Amazin’ Software”) y “Ocean Software”.

En 1983 Nintendo lanza en Japón la consola “Famicom”, y unos años después en el resto del mundo con el nombre “NES” (Nintendo Entertainment System). También aparece el juego “Mario Bros.”, en el que el personaje anteriormente conocido como Jumpman recibe el nombre por el que es mundialmente conocido hoy en día.

Aparece en este año la primera empresa española dedicada al negocio de los videojuegos: “Indescomp”. Comenzó distribuyendo en España las consolas “ZX Spectrum” y “Amstrad CPC”, y posteriormente fue la primera en distribuir videojuegos de origen español (como “La pulga”).

En las navidades de este año se desencadena la “Crisis de 1983” en el sector del videojuego debido a que muchas compañías se lanzaron a la producción de videojuegos, apareciendo en el mercado una ingente cantidad de títulos clónicos y desconocidos (actualmente gracias a Internet podemos informarnos sobre los juegos que salen al

mercado, pero en aquella época era difícil distinguir a un juego bueno de uno malo). Esto provocó que los comercios tuvieran una gran cantidad de juegos que no podían vender, por lo que empiezan a rebajar los precios. A esto hay que sumarle el éxito de los ordenadores personales, que ofrecían mucho más que las consolas. Todo esto provocó grandes pérdidas a las empresas: Coleco y Magnavox abandonan el negocio, y Atari se encuentra cerca de la bancarrota (fue vendida por Warner a Commodore por 240 millones de dólares). A pesar de esta crisis aparece una nueva compañía, “Squaresoft”, que llegaría a ser una de las más importantes en la historia de los videojuegos.

En 1984 el sector de los videojuegos se encuentra sumido en una gran crisis en todo el mundo salvo en España, pues en este año comienza la “Edad de Oro del Videojuego Español”, apareciendo las primeras empresas españolas dedicadas al software de entretenimiento. En noviembre aparece “MicroHobby”, una de las primeras revistas españolas dedicadas a los videojuegos, aunque, al igual que “ZX”, estaba dedicada sólo a la Spectrum.

También es destacable en este año la aparición de la primera aventura gráfica: “King’s Quest I: Quest for the Crown” (implicó más de 18 meses de trabajo y un presupuesto de más de 700.000 dólares, y supuso una revolución). Existen 8 entregas del juego y varios remakes.

En 1985 comienza a recuperarse la industria y Nintendo lanza en EEUU la NES y un nuevo juego: “Super Mario Bros.”, uno de los juegos más famosos y el que convirtió a Mario en la mascota de la empresa. Destacan también en este año la aparición del juego “Tetris” y el lanzamiento por parte de Sega de la consola “SG-1000 Mark III” en Japón, presente un año más tarde en el mercado americano con el nombre “Master System”.

Sega lanza en 1988 una consola de 16 bits con la idea de que fuera tan potente como los mejores ordenadores del momento. Se lanzó en Japón con el nombre “Sega Mega Drive”, y un año más tarde en EE.UU. con el nombre “Sega Genesis” (en Europa aparece en 1990 con este mismo nombre). También en 1988 aparece el que para muchos fue, durante mucho tiempo, el mejor videojuego de creación española: “La Abadía del Crimen”, un juego basado en la novela “El nombre de la rosa”.

En 1989 Nintendo lanza la “Game Boy”. Buscaban una consola pequeña, ligera, barata y con muchos juegos, y como resultado apareció esta consola, sucesora de las Game

& Watch (con la diferencia de que la Game Boy permitía intercambiar los juegos, que estaban grabados en cartuchos).

3.1.4. Década de los 90

A comienzos de los 90 aparece la nueva consola de Nintendo, la “Super NES” (o “Super Famicom”). Por su parte, SNK comercializa “Neo Geo”, un sistema de cartuchos para máquinas recreativas, y también una versión doméstica del mismo con la que los jugadores podían experimentar un juego similar al de las recreativas. Tuvo mucho éxito, pero su elevado precio la convirtió en una consola al alcance de pocos.

España pone fin en 1991 a su edad de oro y cae en una crisis provocada por el estancamiento en los 8 bits, mientras en Europa se extendía el mercado de los 16 bits. En consecuencia, muchas desarrolladoras españolas fueron desapareciendo.

Entre 1994 y 1995 Sony lanza la consola que sentaría las bases de los videojuegos tal y como los conocemos hoy en día, la “PlayStation”. Esta consola lideró el cambio de gráficos 2D a gráficos 3D. Poco antes, en estos mismos años, Sega lanzó una nueva consola: la “Sega Saturn”, pero, a pesar de ser una plataforma prometedora, su precipitada comercialización y la aparición de la PlayStation (cuyo éxito ensombreció a la Saturn) hicieron que esta consola recibiera una escasa acogida.

En 1996 se lanza la “Nintendo 64”, con la que Nintendo da también el salto a las 3D. Nintendo también estrena este año la “Game Boy Pocket”, una versión más compacta de la Game Boy y con una pantalla más grande (en 1998 aparecería la “Game Boy Color”, con mejores gráficos, hardware y velocidad, pero sus principales atractivos eran la presencia de color en los juegos y la compatibilidad con los juegos de la Game Boy original).

En 1998 sale a la venta en Japón la última consola de Sega, “Dreamcast” (en Estados Unidos y Europa aparece en 1999). Aunque no están muy claros los motivos, esta consola fue un fracaso, y Sega dejó de fabricarla en 2001. Sin embargo, a día de hoy es considerada una de las mejores consolas de la historia.

3.1.5. Desde el año 2000 hasta la actualidad

Sony inicia el nuevo siglo con su “PlayStation 2” (o “PS2”), que se distribuyó por todo el mundo en el año 2000 y arrasó con la competencia. Incluía un lector de DVD (para juegos y películas) y puertos USB, y era retrocompatible con los juegos de PlayStation 1. Se convirtió en una de las consolas más vendidas de la época, en parte debido al éxito de su predecesora, e incluso se siguió vendiendo durante bastante tiempo tras la aparición de la “PlayStation 3”.

En 2001 Nintendo comercializa la “GameCube”. Anunciada en 1999 como “Project Dolphin”, su presentación fue espectacular y tuvo muy buena acogida. Sin embargo, el retraso del lanzamiento al 2001 permitió a Sony coger ventaja con su PS2, y GameCube quedó relegada a un 2º lugar. Nintendo también lanza la “Game Boy Advance” (“GBA”), sucesora de las anteriores Game Boy.

En Estados Unidos también aparece este año la “Xbox” por parte de Microsoft (en Europa y Japón saldrían al año siguiente). Fue la primera consola en incorporar un disco duro. Tuvo bastante éxito en EE.UU., sin embargo, al igual que la GameCube, esta consola también quedó a la sombra de la PlayStation 2.

En 2003 aparece el sistema PEGI (Pan European Game Information) que clasifica los videojuegos según la edad mínima recomendada y su contenido en Europa.

Nintendo lanza la “Game Boy Advance SP” (“GBA SP”), una nueva versión de la GBA, con batería de Ion Litio (en sustitución de las pilas), formato plegable y pantalla con iluminación frontal. En 2004 vuelven a estrenar consola con su “Nintendo DS”, con el que establecieron una nueva forma de juego gracias a su pantalla táctil (“DS” son las siglas de “Dual Screen”). Además de la pantalla táctil (para la que llevaba un puntero), incluía un micrófono, conexión Wi-Fi para jugar online, y era compatible con los cartuchos de Game Boy Advance. Al igual que la GBA SP, esta consola es plegable. Más adelante saldrían otras versiones de esta consola (las más actuales permiten jugar en 3D sin necesidad de gafas). Por su parte, Sony decide introducirse también en el mercado de las consolas portátiles con su “PSP” (“PlayStation Portable”). Emplea el formato UMD (Universal Media Disk), que es similar a un CD pero de tamaño muy reducido y con una capacidad de 1,8 GB. Aparte de juegos, la consola también permite reproducir películas, música y fotografías.

En 2005 Microsoft lanza su nueva consola, la “Xbox 360”, de forma simultánea en todo el mundo (es la primera que lo hace), y un año más tarde aparecen las consolas que competirían con ella en la que sería la séptima generación de consolas: “PlayStation 3” (de Sony) y “Wii” (de Nintendo).

En la actualidad, la última generación de consolas comprende la “PlayStation 4” (2013-2014) de Sony, la “Xbox One” (2013-2014) de Microsoft y la “Wii U” (2012) de Nintendo (sucesoras de las mencionadas en el párrafo anterior), y están en marcha interesantes proyectos, como el visor de realidad virtual Oculus Rift, creado por la compañía Oculus VR.

3.2. Tipos de videojuegos

Se pueden establecer varias tipologías a la hora de clasificar los videojuegos en categorías. Sin embargo, la más común y amplia es la clasificación por géneros de videojuego. Son los siguientes:

• Acción

Son juegos en los que el jugador controla un personaje u objeto en tiempo real con el objetivo de llegar a una meta, encontrar objetos, ganar puntos o derrotar a los enemigos. Dentro de este género hay una gran variedad de subgéneros:

- **Beat 'em up:** En estos juegos, el jugador controla a un personaje con el que tendrá que enfrentarse a un gran número de enemigos a medida que avanza por el escenario. En muchos beat 'em up se permite jugar en un modo cooperativo de 2 jugadores. Por lo general, son juegos en 2D y scroll horizontal, pero también existen juegos en 3D. Estos últimos suelen situarse dentro de un subgénero de los beat 'em up, llamado “**Hack n' Slash**”, caracterizado por movimientos y ataques rápidos.



Street of Rage (Beat 'em Up)



Devil May Cry 4 (Hack n' Slash)

- **Lucha:** Presentan combates entre un número limitado de jugadores (por lo general 1vs1, pero se puede aumentar el número de participantes en algunos juegos) en un escenario limitado y desde una vista lateral. El objetivo suele ser reducir a cero la barra de salud del rival.



Tekken 3

- **Infiltración:** El objetivo de estos juegos es avanzar por el escenario y cumplir los objetivos sin ser detectado por el enemigo, con lo que, a diferencia de los géneros anteriores, en estos juegos priman la paciencia y el sigilo.



Metal Gear Solid 2

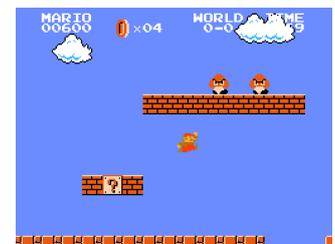
- **Survival horror:** Este género se caracteriza por la influencia del cine de terror. El objetivo es sobrevivir en un entorno oscuro y hostil en el que el protagonista es acechado por enemigos sobrenaturales y



Silent Hill

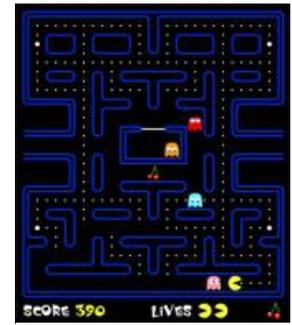
- otros peligros. son habituales los juegos de este género en los que el protagonista va equipado con armas de fuego, y suelen escasear los recursos (munición botiquines, etc.), lo cual sirve para aumentar esa sensación de vulnerabilidad y desamparo que se busca en el género de terror.

- **Plataformas:** Son juegos en los que el jugador avanza por el escenario sorteando obstáculos. En todos los juegos de este género está presente la habilidad de saltar. Además, en función del tipo de juego, el protagonista puede hacer uso de otras habilidades para evadir los obstáculos o vencer enemigos.



Suoper Mario Bros.

- **Arcade:** Reciben este nombre porque mantienen las características de los antiguos juegos de recreativas presentan mecánicas sencillas y niveles cortos, en los que la dificultad va aumentando al progresar (con frecuencia, la dificultad se establece por la velocidad del juego).



Pacman

• Shooter

Este tipo de juegos, como su propio nombre indica, centra la acción en los disparos. Presenta los siguientes subgéneros:

- **Shoot 'em up:** El jugador controla a un personaje o vehículo que puede disparar una cantidad ilimitada de proyectiles (a veces en ráfagas) a los enemigos (que también disparan o atacan cuerpo a cuerpo). En estos juegos es frecuente que la pantalla se llene de balas y el jugador deba esquivarlas constantemente. La vista puede ser cenital (scroll vertical) o lateral (scroll horizontal).



Tyrion 2000

- **First Person Shooter (FPS):** En estos juegos la cámara se sitúa en los ojos del personaje, por lo que lo que éste ve es lo que ve el jugador. Los controles consisten en desplazar al personaje y usar las armas, principalmente de fuego (aunque también suele haber granadas y un cuchillo). Por propia definición del género, todos los FPS son juego con gráficos 3D.



Counter Strike

- **Third Person Shooter (TPS):** Estos juegos mezclan los disparos y el combate con la interacción con el entorno. La cámara se sitúa detrás del personaje o desde arriba, creando una perspectiva isométrica respecto a éste. La precisión al disparar

es menor, pero a cambio, hay mayor libertad de movimiento (muchos juegos ofrecen la posibilidad de usar una vista en primera persona al apuntar con el arma).

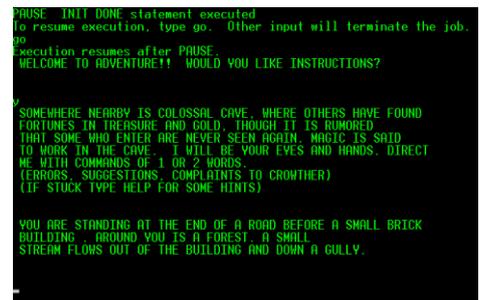


Resident Evil 5

• Aventura

Los juegos de aventura se basan en el relato de una historia y en la resolución de rompecabezas y desafíos mediante el ingenio. Se suelen clasificar también como juegos de aventura aquellos que se basan en recorrer niveles venciendo a enemigos y recogiendo objetos de valor, una definición bastante acorde a la del género de acción, pero es probable que esta clasificación de juegos de “aventura” venga dada por el argumento del mismo y por la inclusión de puzzles a lo largo de la acción. Los principales subgéneros de aventura son:

- **Conversacional:** Estos juegos son los más antiguos del género de aventura y debido a las limitaciones de la tecnología no presentaban imágenes, sólo texto. El jugador interactúa con el juego introduciendo comandos con el teclado, es decir, “hablando” con el juego.



Colossal Cave Adventure

Posteriormente se añadieron imágenes y surgió un subgénero dentro de las aventuras conversacionales: la **novela visual**, en la que el jugador se limitaba a escoger entre las opciones que le ofrecía el juego para avanzar en la trama.

- **Gráfica:** Son bastante parecidas a las novelas conversacionales, pero en las aventuras gráficas, el jugador interactúa con el escenario para recoger objetos, abrir puertas, desplazar al personaje, etc. mediante el uso del ratón (por este motivo son conocidos también como



The Secret of Monkey Island

juegos “point and click”).

• Sandbox

La característica principal de estos juegos es la oferta de un amplio escenario para explorar. Estos juegos tienen una historia principal que se desarrolla mediante misiones, pero también presentan misiones secundarias, ajenas a la historia, a lo largo de todo el mapa.



Red Dead Redemption

• Deportes

Este tipo de juegos simulan deportes reales. Aunque en algunos casos son deportes ficticios o parodias de deportes existentes, en la mayoría de los casos los juegos de deportes intentan recrear de la forma más fiel posible el deporte real, pudiéndose incluir en el género



F1 2015

de los simuladores. Destacan los juegos de carreras y los de fútbol, aunque también hay otros muchos deportes, como baloncesto, golf, tenis, etc.

• RPG

Los RPG (Role Playing Game) o juegos de Rol son aquellos en los que el jugador controla a un personaje o equipo de personajes (es habitual que pueda elegir a los miembros de su equipo de entre un grupo de personajes mayor) con una serie de datos estadísticos (nivel, salud, magia, defensa, etc.) que irán mejorando



Final Fantasy X

al luchar contra los enemigos para ganar puntos de experiencia y subir de nivel. A lo largo de la historia el jugador pasará por distintos pueblos y ciudades donde recuperar energía y abastecerse de armas y objetos, y también deberá enfrentarse a escenarios laberínticos como mazmorras, bosques, etc. En los RPG tradicionales, el jugador debe pensar de forma estratégica y emplear sus recursos de forma estratégica, pues su principal característica es el sistema de combate por turnos. Sin

embargo, en los últimos años han aparecido otros tipos de juegos dentro del género RPG:

- **Tactical RPG:** también conocido como Strategic RPG, es un género que se centra especialmente en la estrategia, pues el jugador controla a un grupo de personajes a los que desplazará por turnos sobre un escenario dividido en casillas (como si de un tablero de juego se tratase), tomando decisiones tácticas en base al terreno, la posición de los enemigos y las habilidades y debilidades de cada personaje.



Fire Emblem: Awakening

- **Action RPG:** En este subgénero, a diferencia del anterior, los combates se desarrollan en tiempo real, por lo que el jugador debe usar su habilidad y reflejos para vencer, más que la estrategia.



Tales of Xillia

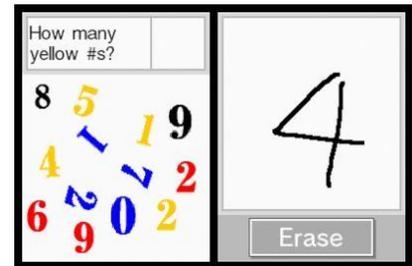
- **MMORPG (Multiplayer Massive Online Rol Playing Game):** Estos son juegos de rol online que a su vez están dentro de otro género: los MMO. Éstos reúnen a jugadores de todo el mundo en partidas multitudinarias. En función del juego, los jugadores pueden competir entre ellos, colaborar para lograr objetivos comunes, o ambas. Cada vez es más frecuente ver torneos de juegos MMO a nivel global en los que participan jugadores profesionales, hasta tal punto que pueden ser considerados como un nuevo deporte.



World of Warcraft

• Educativos

Son aquellos que se desarrollan con el objetivo de educar, impartir conocimientos o ejercitar diversas capacidades físicas o psicológicas. Son habituales los juegos educativos orientados al público infantil, pero también existen para adultos, en función de cual sea el objetivo del mismo.



Brain Training

• Estrategia

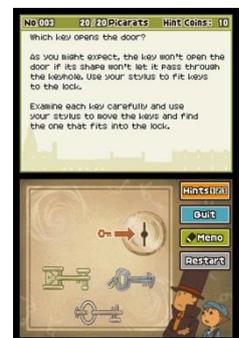
En estos juegos, el jugador debe usar un pensamiento táctico y planificar sus acciones para alcanzar la victoria. Debe administrar los recursos de los que dispone para alcanzar una situación ventajosa ante el objetivo al que se enfrenta.



Age of Empires

• Puzzle

Son juegos cuyo objetivo es resolver puzzles, acertijos o juegos de ingenio. Es frecuente que en estos juegos haya algún límite de tiempo o un número limitado de intentos.



Profesor Layton y la Caja de Pandora

• Casuales

Son juegos con controles sencillos cuyo objetivo es entretener al jugador de forma casual (no tienen una historia que completar ni requieren largas y frecuentes partidas para subir niveles). El público objetivo no precisa tener mucha experiencia con los videojuegos y

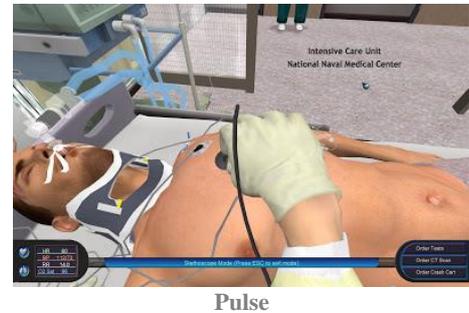


Mario Party 10

son frecuentes los juegos multijugador. Dentro de este tipo de juegos se pueden incluir los de minijuegos, los de tablero, los de cartas, los de música, los de pinball, pachinco, tragaperras...

• Simuladores

Estos juegos sitúan al jugador en un entorno ficticio que pretende imitar un entorno real. Según el objetivo, se pueden distinguir entre simuladores lúdicos, en los que el objetivo no es otro que entretener al jugador (simuladores de construcción, por ejemplo), y simuladores educativos, que pretenden adiestrar al jugador en una determinada disciplina (como los simuladores de vehículos), también conocidos como “serious games” (estos también se emplean en entrevistas de trabajo para poner a prueba al candidato). Sin embargo, no todos los serious games son simuladores, pues su principal característica es su objetivo, consistente en la adquisición, por parte del jugador, de determinadas destrezas, conocimientos o ideas, independientemente del factor lúdico o, incluso, aprovechándolo para potenciar su efecto. Así, los serious games pueden agruparse en 5 categorías diferentes, según exponen Julián Álvarez y Olivier Rampnoux en el artículo “**Serious Game: Just a question of posture?**” (**Artificial & Ambient Intelligence, 2007, P. 420-423**):



- **Edutainment games**, o juegos educativos.
- **Advertgaming**, o juegos destinados a la promoción de una marca o producto.
- **Subergames**, o juegos de denuncia y crítica.
- **Simuladores**.
- **Edumarket games**, o juegos que engloban varias de las categorías anteriores.

Éstos son los principales géneros de videojuegos, pero no son todos. De hecho, al igual que en otros medios, a veces es difícil distinguir entre un género y otro, ya que comparten características, e incluso muchos juegos mezclan propiedades de varios géneros (creando a veces géneros completamente nuevos).

Pero, aunque no se emplean tanto a la hora de distinguirlos, hay otras formas de clasificar videojuegos:

- **Géneros narrativos:** actualmente la mayoría de los videojuegos narran una historia, son “películas interactivas”. Aunque algunos géneros de videojuegos están por lo general ligados a un determinado género narrativo (como los survival horror o los juegos de aventuras), cabe mencionar esta clasificación, pues el género narrativo en que se enmarque el juego determinará su historia, complementando la mecánica del mismo (que es determinada por el género del videojuego). Así, tendremos juegos de acción, drama, comedia, cine negro, etc.
- **Sistema PEGI:** sus siglas se traducen como “Información Paneuropea sobre Juegos” (Pan-European Game Information) y es un código que emplea etiquetas en los videojuegos para clasificarlos por edades, así como advertir acerca de contenidos que puedan ofender a algunos jugadores (a excepción de la etiqueta de juego online, pues más que advertir, ofrece esa opción). Se estrenó en 2003 y sustituyó diversos sistemas de clasificación para establecer una clasificación homogénea en los diferentes países de Europa (actualmente se utiliza en 30 países europeos). El sistema fue creado por la Federación de Software Interactivo de Europa (ISFE), y está respaldado por los principales fabricantes de consolas.



1 Sistema PEGI

- **En función de la plataforma y otros tipos de juegos:** dependiendo de cuál sea el medio a través del cual juguemos, se pueden distinguir entre juegos de PC, de consola o de móvil y Tablet (los cuales están en alza últimamente). Además, se pueden hacer otras distinciones, como entre los juegos de ocio y los ya mencionados serious games, o entre los juegos “triple A” (aquellos realizados mediante grandes equipos, presupuesto y tecnología) y los juegos “indies” (producidos por equipos pequeños y con poco presupuesto, pero con la ventaja de una mayor libertad creativa).

3.3. Estudios sobre videojuegos

Es muy frecuente escuchar comentarios negativos sobre los videojuegos. Como ocurre con cualquier otra cosa, los videojuegos en exceso pueden tener efectos negativos en los jugadores. Sin embargo, el uso de los videojuegos tiene muchos efectos positivos. A continuación se explican algunas ventajas de jugar a videojuegos recogidas en el libro **“Vida Extra (Gina Tost y Oriol Boira; 2015; editorial Grijalbo)”**, así como algunos estudios que corroboran dichas ventajas:

• Reducción del estrés

Algunos estudios demuestran que los videojuegos ayudan a reducir el estrés y la ansiedad, así como a hacer frente a enfermedades mentales, como el realizado por Carmen Russoniello (profesor asociado y director del Centro de Psicofilosofía y biofeedback de la ECU), el cual se realizó en base a juegos casuales de palabras y rompecabezas como “Bejeweled 2”, “Peggle” y “Bookworm Adventures”. El juego “Bejeweled 2” redujo el estrés de los jugadores en torno a un 54%. Otro estudio realizado por Christopher J. Ferguson y la Universidad de Texas demostraba que los videojuegos violentos ayudan a reducir el estrés, la depresión y las actitudes hostiles.

Mark Baldwin (de la McGill University) fue más lejos todavía. Mientras él y su equipo estudiaban los comportamientos de los jugadores, descubrieron que éstos pueden aliviar la presión y el estrés al hacer que el jugador se centre en un objetivo y ofrecerle estímulos positivos. Tras ver los resultados obtenidos en el estudio fundaron la empresa MindHabits Inc. con el objetivo de crear videojuegos centrados en aumentar la autoestima y reducir la tensión social. Uno de sus primeros juegos fue MindHabits Trainer, el cual se puede encontrar en www.mindhabilities.com.

• Coordinación Mano-ojo

Jugar con regularidad a videojuegos también puede mejorar destrezas físicas y mentales en los jugadores (como mejorar los reflejos, las habilidades motrices y la precisión, entre otros ejemplos). A finales de 2012, científicos de la Universidad de Texas Medical Branch realizaron un estudio en el que participaron estudiantes de secundaria (que jugaban una media de 2 horas al día), estudiantes universitarios (que jugaban 4 horas al día) y médicos profesionales: su objetivo era realizar una cirugía virtual. Los

participantes realizaron 32 pruebas diferentes en las que eran puestas a prueba distintas destrezas: coordinación mano-ojo, precisión de controles manuales y control del tiempo, entre otras. El estudio demostró que los estudiantes de secundaria y los universitarios realizaron la cirugía con igual o mayor precisión que los médicos (quienes prácticamente no jugaban).

- **Toma de decisiones**

Como ya se comentó antes, los videojuegos mejoran aspectos tanto físicos como psicológicos de los jugadores. Según un estudio realizado por Daphne Bavelier, doctora de la Universidad de Rochester, los jugadores tienen una mejor capacidad visual y de atención, y poseen una mayor facilidad para desempeñar ejercicios de resolución de conflictos, así como para realizar varias actividades de forma simultánea. Los juegos de acción permiten mejorar nuestra habilidad a la hora de analizar la información que recibimos y tomar rápidamente la decisión que consideremos mejor para resolver un problema.

Por otra parte, la Queen Mary University y la College University de Londres llevaron a cabo un estudio en el cual reunieron a 72 voluntarias para que jugaran a un videojuego durante 40 horas a lo largo de seis u ocho semanas: la mitad de las voluntarias jugó a Starcraft (un juego de estrategia en tiempo real), mientras que la otra mitad jugó a The Sims (un simulador de vida que no requiere de memoria o estrategias complejas). Tras esto se realizaron varias pruebas a las participantes que demostraron que aquellas que habían jugado a Starcraft demostraban una mejor flexibilidad cognitiva, descrita como la habilidad de una persona de adaptar su forma de pensar a distintas situaciones para solucionar problemas o alternar distintas actividades.

- **Trabajo en equipo y socialización**

Se suelen identificar los videojuegos con la idea de aislamiento social, sin embargo esto no es así. Para mucha gente los videojuegos son una afición, al igual que los libros o el cine, lo que supone una amplia fuente de temas de conversación, ya que la gente socializa más fácilmente en grupos en los que los integrantes comparten aficiones e intereses comunes. Además de esto, los videojuegos multijugador que ofrecen un modo cooperativo o por equipos requieren el desempeño de la estrategia y del trabajo en equipo entre los jugadores para poder lograr la victoria. La doctora Beatriz Marcano llevó a cabo

un estudio en 2013 sobre el trabajo en equipo en videojuegos multijugador de acción en el que analizaba esta idea a través de una encuesta realizada a 294 jugadores, distinguiendo entre jugadores que pertenecían a algún equipo o clan (así como los distintos roles dentro de éstos) y aquellos que no. Aunque algunos afirmaban que no jugaban en equipo (sobre todo los jugadores que no pertenecían a un clan), la mayoría de las respuestas apuntan a una forma de juego colaborativa y de trabajo en equipo. Muchos de los encuestados también afirmaron sentirse irresponsables a la hora de jugar, pero esto puede achacarse a las circunstancias del juego (la capacidad de revivir si mueres, el hecho de estar en un juego, o la posibilidad de usar trucos para jugar con ventaja).

La doctora Marcano sugirió en el Congreso Iberoamericano de Investigadores Noveles la posibilidad de aprovechar esta relación entre los videojuegos y el trabajo en equipo en ámbitos profesionales o educativos, centrándose en éste último. Según afirma, a través del proceso de metacognición “el maestro o asesor puede incitar la reflexión acerca de las acciones hechas en el juego y los resultados y consecuencias que tuvieron esas acciones a través de preguntas sencillas”.

• **Mayor Creatividad**

A estas alturas, con tantas ventajas psicológicas que hemos visto en los estudios anteriores, no es raro pensar que los videojuegos mejoren la creatividad de los jugadores. De hecho, la creatividad está muy presente en éstos: por un lado, hay muchos juegos (prácticamente la totalidad de los juegos indies) que denotan de por sí un gran acopio de creatividad en sus mecánicas y en sus historias (ejemplos de esto son “Limbo”, “Bastión”, “Fez”, “Splatoon”, “Ori and the blind forest”, “Portal”, etc.); por otro, existen, y cada vez de forma más frecuente, juegos que buscan explotar la creatividad del jugador, como “Minecraft”, “Little Big Planet”, “Scribblenauts”, etc.

Aprovechando esto, la escuela sueca Viktor Rydberg ha incluido entre sus asignaturas una sobre Minecraft, un exitoso juego basado en la construcción con bloques y en la búsqueda de recursos. Con el uso de este juego, la escuela busca dar a sus alumnos lecciones sobre conciencia ambiental, creatividad y urbanismo. Por su parte, un grupo de profesores estadounidenses y finlandeses han creado un mod de Minecraft que les permite crear un entorno específico en función de la materia a impartir. Los mods son expansiones que permiten modificar elementos de un videojuego como el entorno, el aspecto de los personajes, los objetos, etc.

La mayoría de los estudios descritos en este trabajo pueden encontrarse en el libro **“Vida Extra (Gina Tost y Oriol Boira; 2015; editorial Grijalbo)”**. Sin embargo, hay otro tema de estudio que no hemos mencionado aún y que merece atención: la violencia en los videojuegos. Todos hemos oído alguna que otra vez lo mismo: Los videojuegos hacen violentos a la gente. Sin embargo, esto no es exactamente así. Alguien que de por sí es agresivo puede reaccionar de forma agresiva al jugar un videojuego, de la misma forma que pasaría al ver una película, leer un libro o incluso escuchar una canción violenta. Pero un videojuego no creará este efecto en un público normal. Un artículo interesante sobre este tema es el publicado por Daniel J. en la web “Infoconsolas” en septiembre de 2009. En él, el autor analiza diferentes estudios a favor y en contra de este tema mostrando que, aunque hay estudios en los dos bandos, aquellos que afirman que los videojuegos causan agresividad en los jugadores no lo hacen de forma contundente o no tienen pruebas concluyentes que respalden esta teoría.

4. DESARROLLO DE UN VIDEOJUEGO

Como cualquier otro producto audiovisual, un videojuego requiere de una determinada metodología para su creación. Existen diversos sistemas y métodos de trabajo y, aunque no hay una estricta delimitación o consenso entre ellos, muchos coinciden en la presencia de tres etapas indispensables en las que dividir todo el proceso: preproducción, producción y postproducción. La metodología de desarrollo que explicaremos a continuación corresponde a la presentada en el libro **“Desarrollo de Videojuegos: Técnicas Avanzadas (Francisco Moya Fernández, Carlos González Morcillo, David Villa Alises, Sergio Pérez Camacho, Miguel A. Redondo Duque, César Mora Castro, Félix J. Villanueva Molina, Miguel García Corchero; 2012; Bubok, LibroVirtual.org)”**.

4.1. Preproducción

Esta fase es muy importante, ya que si se producen errores u omisiones, éstas se arrastran a las fases de producción y postproducción. En esta parte se lleva a cabo lo que se conoce como Diseño Conceptual de juego, es decir, aquellos elementos preliminares o “poco tangibles”. Se crea la idea del juego, estableciendo los elementos fundamentales que lo caracterizarán, y se plasma en un documento conocido como “Game Design Document” (GDD) o Documento de Diseño del Juego. Este documento debe ser elaborado por el equipo creativo y debe incluir todo aquello que será necesario abordar en la fase de producción y (en menor medida) en la de postproducción para el desarrollo del videojuego. Debe contener la siguiente información:

- **Género.** Clasificación del juego según su naturaleza. Los géneros son las categorías que permiten diferenciar o agrupar los videojuegos en función de la representación gráfica, el tipo de controles y de interacción jugador-máquina, la ambientación y temática, el objetivo y la plataforma de juego (móvil, consola, PC). En anteriores páginas de este documento hemos mencionado y descrito los principales géneros de videojuegos.
- **Jugadores.** Tipo de juego en función de los jugadores: individual o multijugador, si los jugadores son personas o están controlados por el juego, etc.
- **Historia.** Primera aproximación a la trama a desarrollar durante el juego, destacando qué se quiere contar (storyline) y cómo se quiere contar (storytelling).

- **Bocetos.** Diseños preliminares de los elementos visibles del videojuego: personajes, escenarios, armas, objetos, etc.
- **Look and Feel.** Corresponde al aspecto gráfico y artístico del juego, formado por: colores, temas dominantes, música y sonidos, técnicas de diseño 3D o 2D, posiciones de las cámaras, etc. Se realiza a partir de los bocetos.
- **Interfaz de Usuario.** Se indicará la forma en que el jugador interactuará con el juego y los mecanismos de que dispondrá para ello. Hay que distinguir dos tipos de interfaces: el Head-Up Display (HUD), que se muestra a lo largo del juego, y el conjunto de elementos “externos” al juego (como los menús, selectores de niveles, pantallas de opciones, etc.).
- **Objetivos.** Las metas a alcanzar en el juego en función de la historia o del planteamiento del mismo.
- **Reglas.** Se establece la forma en que el jugador deberá interactuar con el juego para lograr los objetivos, esto es, qué acciones podrá realizar y cómo podrá hacerlo.
- **Características.** La información principal de cada uno de los personajes y de los elementos que intervienen en el juego.
- **Gameplay.** Incluye los aspectos que caracterizan la forma en que se va a jugar el videojuego y la respuesta del mismo, sin detallar en exceso la historia o el apartado gráfico y sonoro. Es un concepto poco preciso, sin embargo, su descripción en el GDD tiene una gran importancia en la calidad final del juego y en su jugabilidad. El libro “Desarrollo de Videojuegos: Técnicas Avanzadas” presenta como ejemplo de descripción de gameplay el caso del popular videojuego “Space Invaders”:

“En este juego indicaríamos que se debe poder mover una nave alrededor del cuadrante inferior de la pantalla y disparar a una serie de enemigos que aparecen por la parte superior de la pantalla y que desaparecen cuando son alcanzados por los disparos. Estos enemigos tratan de atacarnos con sus disparos y presionándonos mediante la reducción de

nuestro espacio de movimientos e intentando chocar contra nuestra nave.”¹

- **Diseño de Niveles.** Descripción de los distintos niveles de dificultad o escenarios (en función del tipo de juego) que se sucederán a medida que el jugador avanza, así como los obstáculos y los objetivos a cumplir en cada uno de ellos.
- **Requerimientos técnicos.** Como su nombre indica, recoge los requisitos técnicos que requerirá el dispositivo o máquina en que se ejecute el juego para su óptimo funcionamiento.
- **Marketing.** Al igual que en los demás productos audiovisuales y, en general, una buena estrategia de marketing es fundamental para el éxito del producto final, por lo que es importante comenzar a plantearlo desde esta fase. Por otra parte, la publicidad en videojuegos, como en el resto de medios, no implica sólo la que se realiza sobre el producto en cuestión para darlo a conocer, sino toda aquella que se presenta dentro del videojuego para anunciar otras marcas que patrocinen el juego (conocido como Product Placement, también se da en películas y series de televisión).
- **Presupuesto.** Cálculo preliminar del presupuesto que implicará el desarrollo del videojuego.

Este es sólo un modelo de un posible GDD, pero hay muchas versiones diferentes, pues no existe una plantilla estandarizada para elaborarlo. Este documento es fundamental para lograr un correcto desarrollo del videojuego que se pretende conseguir. Sin embargo en este trabajo nos centraremos en la fase de producción, por lo que no abordaremos la redacción del mismo.

4.2. Producción

Esta fase supone la creación de los elementos “tangibles”, es decir, los elementos con los que se crea el juego (imágenes, sonido, programación...). Es la parte en la que se concentra la mayor parte del trabajo y en la que participan más trabajadores. Las etapas que se suceden dentro de la fase de producción son las siguientes:

¹ MOYA FERNÁNDEZ, Francisco et al. Desarrollo de videojuegos 3: técnicas avanzadas. Ciudad Real: Universidad de Castilla la Mancha, 2012.

4.2.1. Diseño del juego.

En esta etapa se desarrolla todo lo explicado en el GDD, explicando con mayor profundidad cada uno de los aspectos detallados en él, obteniendo así una versión definitiva del mismo, conocida como “Documento Técnico de Diseño” (“TDD”), además de la “Biblia de la Historia”, la “Biblia del Arte” y la primera versión del Motor del Juego”. El diseño del juego se desarrolla en tres líneas de trabajo diferentes, que son:

4.2.1.1. Diseño Artístico:

- Biblia de la Historia: en ella se narra el argumento completo del juego, así como todas las historias de los personajes y del mundo donde se desarrolla la trama.
- Biblia del Arte: incluye todos los elementos sonoros y gráficos del juego. Estos elementos permiten comenzar a trabajar con el “Motor de Sonido” y el “Motor Gráfico”, respectivamente.

4.2.1.2. Diseño de la Mecánica del Juego:

- Interactuación con el juego: controles, reglas que rigen el juego y el tipo de comunicación, en caso de ser un juego on-line.
- Diseño del comportamiento, habilidades y otros elementos importantes de los personajes y del entorno.
- Se comienza a trabajar en el diseño del “Motor de Inteligencia Artificial” (“IA”) y en todo lo relacionado con esto.
- Diseño del “Motor Físico” para generar los aspectos físicos del juego (explosiones, disparos, etc.).

4.2.1.3. Motor del Juego:

- Agrupa todo lo relacionado con el Motor de Sonido, el Motor, Gráfico, el Motor de IA, el Motor Físico y el resto

de elementos que sean necesarios para representar los elementos funcionales del videojuego.

4.2.2. Diseño técnico.

En esta etapa se describe cómo será implementado el juego. Para ello se emplean notaciones como UML y se decide la metodología de desarrollo de software más apropiada en función de las características y envergadura del proyecto.

Es importante realizar una descripción que permita ver el funcionamiento del software desde las perspectivas estructural, dinámica, de interacción y de despliegue. Además, se debe establecer una planificación de tareas a realizar, la asignación a los miembros del equipo de desarrolladores de dichas tareas, así como la identificación de hitos importantes, fechas de entrega y análisis de riesgos.

4.2.3. Implementación.

Esta es la etapa en la cual se unifican todos los elementos de software y arte elaborados en las dos etapas anteriores. Es muy frecuente encontrar errores del diseño inicial y que sea necesario realizar revisiones, provocando que tanto esta etapa como la anterior se sometan a repeticiones y ciclos iterativos. Esto suele estar marcado por la metodología de trabajo y por factores como los recursos disponibles, la envergadura del proyecto, etc.

En esta fase es habitual realizar demos del videojuego que se publican y suponen un primer contacto de los potenciales jugadores con el producto final, colaborando así en la campaña de marketing.

4.2.4. Pruebas Alpha.

También conocidas como pruebas “Code Complete”, se realizan en el momento en que ya hay partes del producto final completadas. Estas pruebas las realizan pequeños grupos que participaron en el proceso de desarrollo y

diseño del videojuego para buscar errores y perfeccionar algunos detalles, siendo la jugabilidad uno de los elementos que más atención recibe.

4.2.5. Pruebas Beta.

Denominadas también como “Content Complete”, se realizan una vez que el juego ya presenta todos los elementos gráficos y sonoros. Son llevadas a cabo por personal ajeno al equipo de trabajo y, además de asegurar la calidad y buen funcionamiento del videojuego, se analiza la correcta adecuación del juego a las leyes vigentes y a la ética presente en aquellas zonas donde se pretenda publicar el mismo.

4.2.6. Gold Master.

Esta prueba se realiza con la versión definitiva del videojuego. En esta fase, se precisa la mayor difusión posible a nivel publicitario a través de reportajes, artículos en revistas especializadas, etc., entre otros recursos.

4.3. Postproducción

Esta fase incluye la explotación y el mantenimiento del videojuego.

La explotación implica facilitar el producto final al máximo número posible de potenciales jugadores de la forma más eficaz posible. En función del juego y su objetivo, este proceso se llevará a cabo comercializándolo en tiendas físicas y/o virtuales (si el objetivo es económico), publicarlo en Internet de forma gratuita (si el videojuego se ha realizado sin ánimo de lucro), o distribuirlo a las empresas u organizaciones que vayan a hacer uso del producto (ya sea con fines publicitarios, didácticos, etc.). Por supuesto, para que la explotación del videojuego sea exitosa es necesaria una buena campaña de marketing.

El mantenimiento es una etapa del proceso de postproducción que se ha instaurado en los últimos años debido a la proliferación de la conexión a Internet de consolas y videojuegos. Este mantenimiento consiste en el desarrollo de actualizaciones del videojuego original para corregir fallos, mejorar diversas características o, incluso, prolongar la duración del videojuego con contenido extra (dicho contenido se conoce como “DLC” o “Downloadable Content”, y en la mayoría de los casos las compañías cobran un precio adicional por ellos).

Esta metodología de producción de un videojuego que, como ya se mencionó antes, aparece descrita en el libro “Desarrollo de Videojuegos: Técnicas Avanzadas” corresponde a un sistema de trabajo que parte de cero: requiere la creación del motor del juego y, por lo tanto, implica una elevada labor de programación. Sin embargo, en los últimos años algunos programadores han desarrollado diversos motores (conocidos generalmente como motores gráficos) que han puesto a disposición del público, permitiendo así que pequeños estudios puedan desarrollar videojuegos sin necesidad de invertir elevadas sumas de dinero en motores privados o emplear una gran cantidad de tiempo y trabajo en programar uno propio. Esto facilita mucho el trabajo a la hora de desarrollar un videojuego y, en consecuencia, ha facilitado la proliferación de los mismos, especialmente, de los conocidos como “juegos indies” (ya explicados en el apartado dedicado a los distintos géneros de videojuegos). Algunos ejemplos de estos motores gráficos son: “Unity”, “Cry Engine”, “Unreal Engine”, “Game Maker”, etc.

En el desarrollo del videojuego realizado para este trabajo hemos empleado un motor gráfico de los mencionados en el párrafo anterior, concretamente Unity. A continuación explicaremos el proceso de producción derivado del uso de Unity, así como el manejo de esta herramienta.

4.4. Memoria de producción

El videojuego a realizar es un plataformas 2D de puzzles de carácter filosófico-artístico que pretende concienciar sobre la necesidad de cuidar el planeta y de sentirlo como un elemento vivo que puede incluso revolverse contra el propio ser humano. Debido a la ausencia de presupuesto, al reducido equipo de producción y al carácter original del videojuego, se trata de un juego indie.

El protagonista es Gaia, un ente incorpóreo en forma de llama que representa la esencia de la naturaleza y que se encuentra en todos los seres vivos, excepto en el ser humano, que la perdió hace muchos siglos. En el juego, la llama intenta recuperar la relación con el ser humano, unirse a él de nuevo y conseguir que el hombre vuelva a estar en íntimo contacto con la naturaleza.

A través de las distintas fases y niveles, el jugador –que es la propia Llama- avanza con el objetivo de encontrar al ser humano y restablecer su vínculo con él. En el camino se encontrará con dificultades y enemigos. El juego finaliza cuando el jugador ha conseguido que el ser humano se reintegre en la conciencia global del planeta.

A la hora de abordar mi trabajo de creación de videojuego, he seguido las pautas indicadas anteriormente, adaptándolas a mi propia concepción y a mi propio esquema de trabajo.

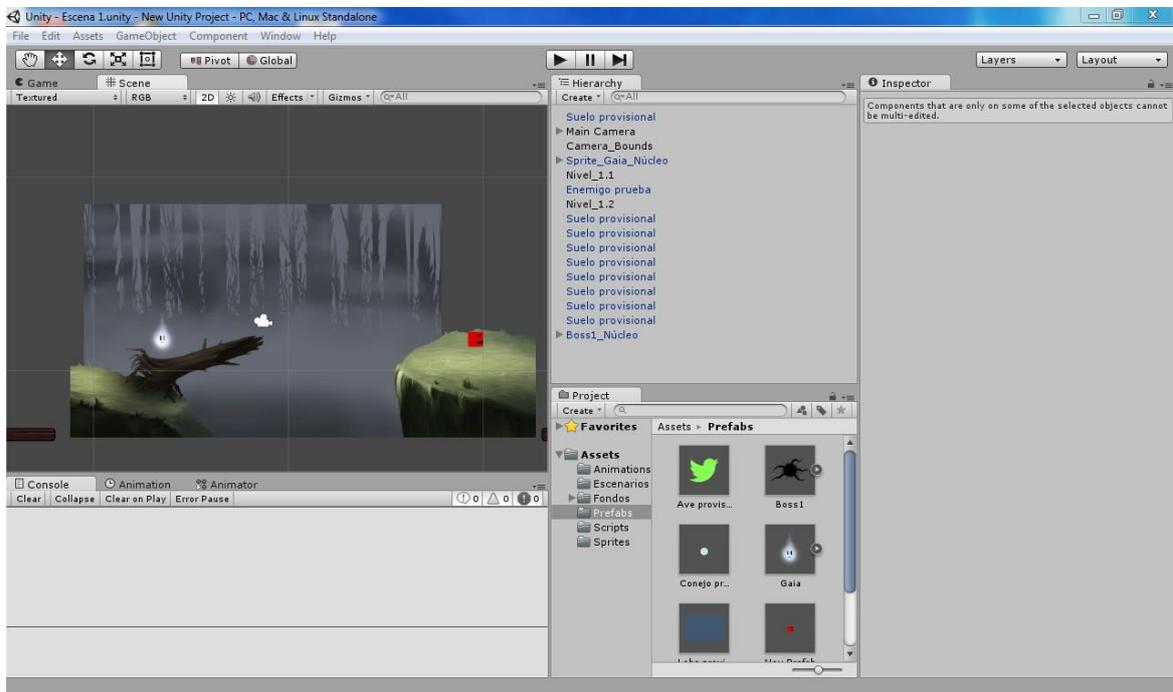
Al mismo tiempo, he tenido que tomar la decisión sobre el motor gráfico a utilizar. Para mis objetivos, entendí que el motor Unity era el más apropiado, ya que es un motor gráfico profesional, gratuito y que permite trabajar tanto en 2D como en 3D.

A continuación explicaremos algunos elementos importantes a la hora de trabajar en la producción del videojuego:

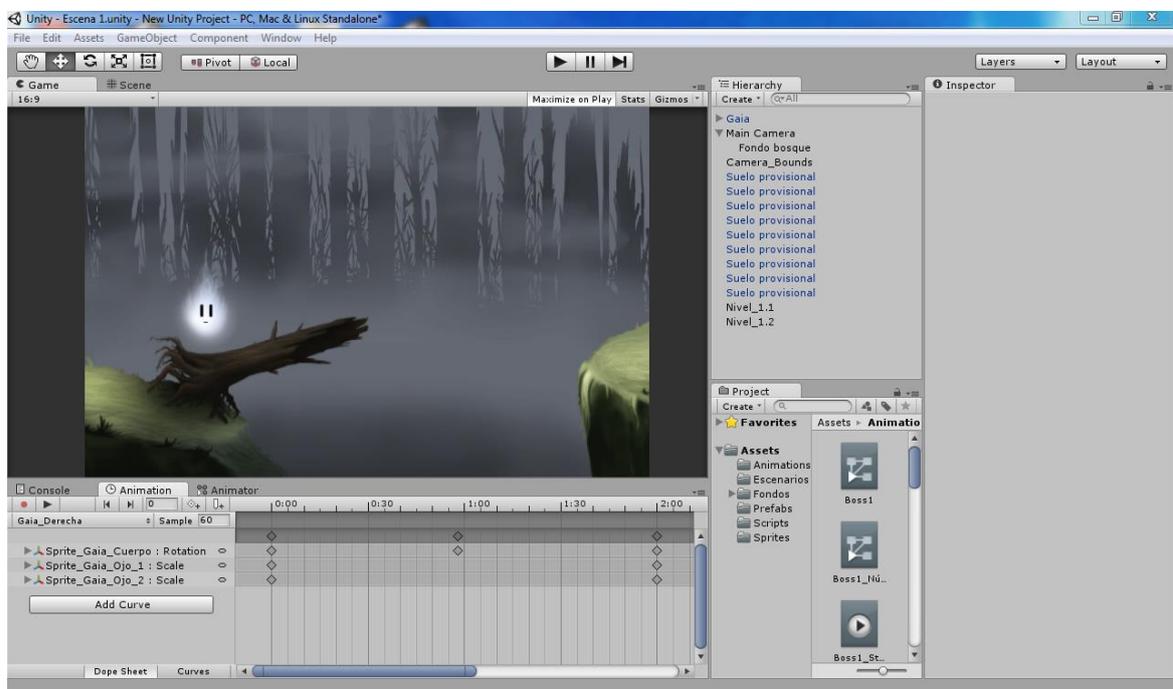
- **Ventanas:** Las siguientes son las ventanas más destacadas a la hora de trabajar con Unity.
 - **Scene (Escena):** Es la ventana con la que más trabajaremos, ya que muestra todo lo que aparece en la escena en la que estamos trabajando, tanto los objetos y sprites, visibles en el juego, como las cámaras, luces, sonidos y demás elementos presentes, así como sus posiciones. Permite desplazar cualquier elemento presente en la escena, así como redimensionarlos, rotarlos, eliminarlos o añadir nuevos elementos.

- **Game (Juego):** Esta ventana muestra una previsualización ejecutable de la escena, tal y como la vería el jugador. Los botones de play y pausa situados en la zona superior central de la interfaz del programa nos permiten ejecutar y detener esta previsualización. Esto es muy útil a la hora de trabajar en el juego para comprobar el correcto funcionamiento de los scripts de programación, de los objetos e imágenes, de los audios, etc.
- **Hierarchy (Jerarquía):** Esta es una ventana que recoge en una columna todos los elementos que están presentes en la escena actual. Los objetos que sean hijos de otros aparecerán dentro de éstos, formando una vista jerárquica en forma de “carpeta” desplegable.
- **Project (Proyecto):** Muestra todos los archivos y las carpetas con los que se trabaja en el proyecto (a diferencia de las ventanas anteriores, que sólo muestran elementos de la escena con la que se trabaja en ese momento).
- **Inspector:** Esta ventana muestra información sobre los sprites, objetos de la escena, scripts, animaciones y cualquier otro elemento del proyecto. Aquí también se pueden editar muchas opciones y valores de dichos elementos, salvo en el caso de los scripts, los cuales se explicarán en el anexo dedicado a la programación.
- **Console (Consola):** Esta ventana muestra el resultado de ejecutar los scripts del proyecto. Generalmente se utiliza para detectar errores, aunque también mostrará valores que se hayan programado para aparecer al ejecutarse, como texto u operaciones matemáticas o lógicas. Si se intenta ejecutar el juego y la consola muestra algún error, Unity impedirá la ejecución. Por otra parte, la consola también mostrará advertencias si detecta alguna incoherencia en el código (por ejemplo, que se cree una variable que el código no utiliza en ningún momento), en este caso el aviso no impide la ejecución del juego, ya que técnicamente no se produce un error.
- **Animation (Animación):** En esta ventana compuesta por una línea de tiempo y un sencillo menú será donde crearemos las animaciones que los distintos objetos mostrarán en el juego.
- **Animator (Animador):** En esta ventana se enlazarán unas animaciones con otras y se establecerán las condiciones necesarias para que se activen.

En las imágenes 2 y 3 se muestra la interfaz de Unity, en la que se pueden apreciar algunas de las ventanas explicadas



2 Interfaz de Unity



3 Interfaz de Unity

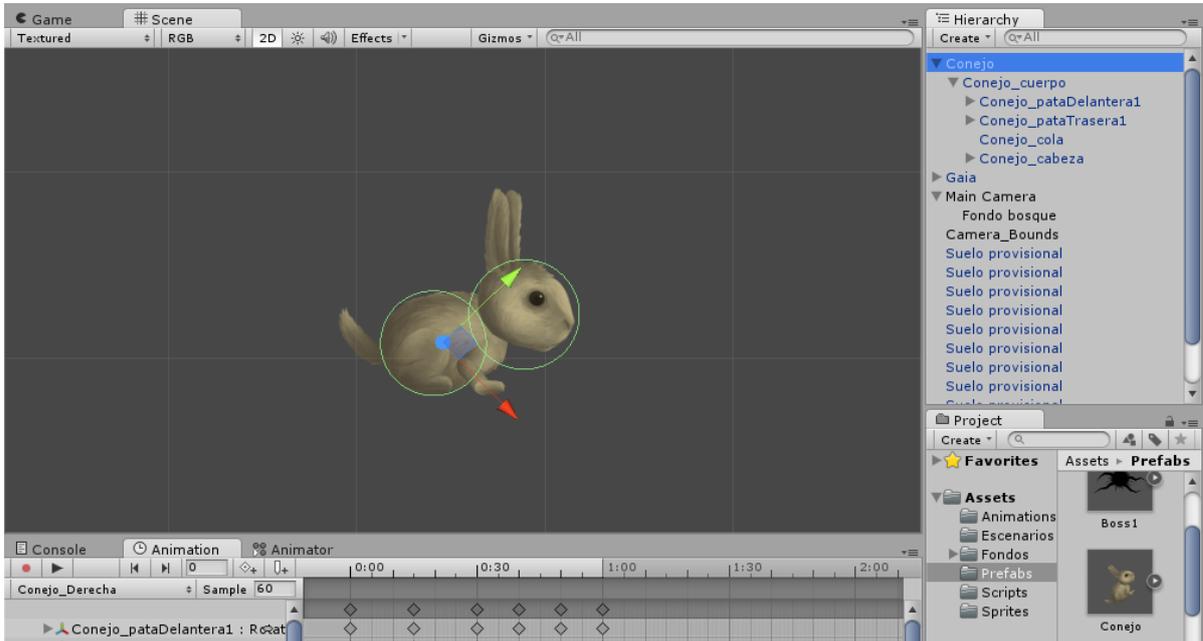
- **Sprites:** Un sprite es la unidad mínima de imagen con la que trabajamos para hacer un videojuego, es decir, es una imagen o conjunto de imágenes que usaremos para representar gráficamente los objetos del proyecto. Cualquier imagen de personajes, ítems o cualquier objeto del escenario, e incluso el mismo fondo de la escena, son sprites. Cabe resaltar que el sprite es sólo la imagen del objeto, no la forma (por ejemplo, una figura tridimensional no es un sprite, sino que sobre él se adhiere el sprite). El siguiente es un ejemplo de los sprites empleados en uno de los personajes de nuestro videojuego.



4 Sprites del conejo

- **Prefabs:** Son objetos “predefinidos” con los que podremos trabajar con varios objetos con las mismas características sin tener que crearlos uno por uno. A ellos se les asigna los sprites, los colliders (o colisionadores, para trabajar con físicas en el juego) y cualquier otra propiedad, efecto o elemento que se le quiera añadir. Para crear un prefab basta con seleccionarlo en el menú desplegable “create” de la ventana de proyecto (aquí se podrán también crear carpetas, scripts y otros elementos). A continuación se arrastra el objeto que queremos establecer como prefab desde la jerarquía hasta el archivo prefab de la ventana de proyecto.

Usaremos el mismo personaje de antes como ejemplo de prefab en la imagen 5.

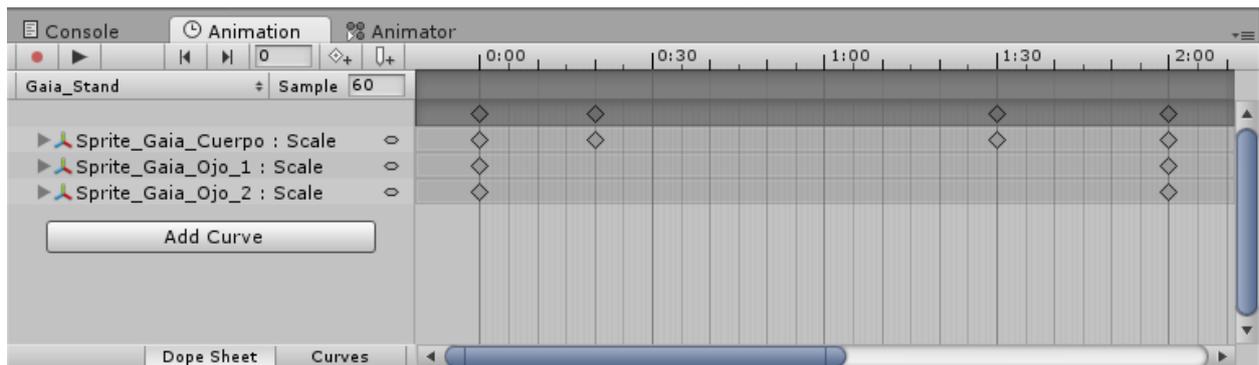


5 Prefab del conejo

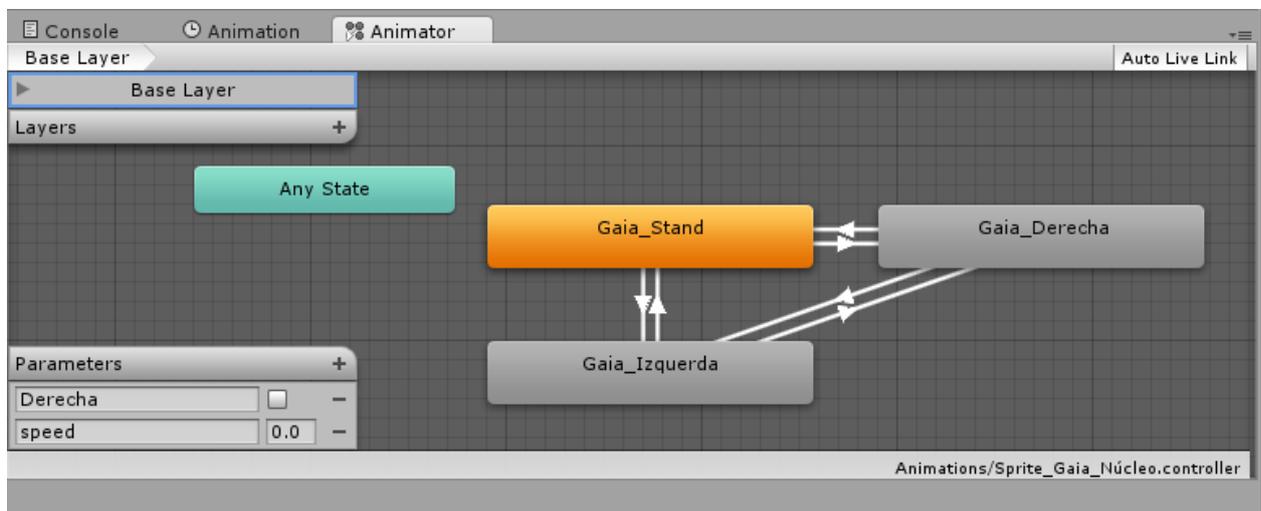
- **Animación:** Se realizará mediante keyframes (o fotogramas clave) que se insertan en la línea de tiempo, estableciendo distintos valores de posición, rotación, tamaño, visibilidad, etc. en cada uno de ellos.

Los keyframes, así como los nuevos valores de esas propiedades, se pueden establecer de 3 formas:

- 1) Modificando el objeto directamente en la ventana de la escena.
- 2) Realizando las modificaciones en la ventana inspector del objeto (esto permitirá modificar más propiedades que la ventana de la escena).
- 3) Realizando los cambios directamente en la ventana animation. Pulsando el botón “Add Curve” (añadir curva) aparecerá una lista de los parámetros del objeto que podemos modificar y que se irán añadiendo al margen izquierdo de la línea de tiempo conforme los vayamos seleccionando junto a los valores que tiene en el momento marcado en la línea de tiempo (si se modifica alguno mediante los 2 procedimientos anteriores, aparecerá automáticamente).



6 Ventana de Animación



7 Ventana del Animator

- Audio:** La implementación del audio es bastante sencilla, pues basta con crear un componente “audio source” en el objeto que vaya a emitir el audio (puede ser un objeto vacío) y asignarle el clip de audio. Después se le ajustan las propiedades (mute, loop, volumen, doppler, reverb, etc.), así como la distancia a la que lo escuchará el “audio listener”. Un audio listener es un objeto que recibe el audio de la escena (literalmente, lo “escucha”) y lo emite, de forma que el audio que recibe el audio listener es el mismo que oye el jugador. Sólo puede haber un audio listener por escena, el cual aparece de forma predeterminada en la cámara principal (aunque es posible cambiarlo). Sin embargo, las opciones que ofrece el audio source son limitadas, y para realizar algunos efectos o

indicar cuándo sonará el audio (lo cual es algo muy importante) es necesario hacer uso de la programación.

- **Programación:** Esta es la parte más extensa y compleja en el desarrollo de un videojuego. Sin embargo, como he explicado anteriormente, se puede rebajar la complejidad de esta tarea trabajando con un motor gráfico, en este caso, Unity. El anexo 4 incluye un tutorial básico de programación en el lenguaje C#, que es el que usaremos en nuestro videojuego. Sin embargo, a continuación explicaremos un par de conceptos importantes: la definición de script y los tipos de errores.
 - **Script:** es el archivo donde se recoge el código a través del cual daremos las instrucciones y las órdenes al programa que lo ejecute. Como cualquier otro tipo de archivo, los scripts presentan en su nombre una terminación en función del formato, en este caso del lenguaje, que posea el archivo: puede ser “.js” (JavaScript), “.py” (Phyton), “.cs” (C#), etc. Nosotros trabajaremos con estos últimos, ya que usaremos el lenguaje C# a la hora de programar el videojuego.

En cuanto a los tipos de errores, hay 3:

- 1) **Errores sintácticos:** son aquellos en los que el programa no entiende el código. la consola te indica este tipo de errores, así como su localización, por lo que son los más sencillos de corregir.
- 2) **Errores de ejecución (runtime errors):** estos errores se dan durante la ejecución del programa (en este caso, del juego), no aparecen antes. Son conocidos también como “excepciones”.
- 3) **Errores semánticos:** en este tipo de errores el código es correcto, es decir, no hay errores en la forma de escribir el código. El error consiste en que lo que queremos decir es diferente de lo que realmente escribimos en el código, por lo que el programa se ejecuta bien, pero no de la forma que nosotros queremos.

La resolución de errores (o debugging) es probablemente la parte más importante de la programación. De hecho, hay gente que entienden la programación en sí misma como un proceso de debug: empiezas con un código cualquiera, y a partir de él vas “corrigiendo errores” hasta lograr tu objetivo.

Como todo trabajo audiovisual, éste también consta de preproducción, producción y postproducción, aunque al ser un proyecto de menor envergadura y que involucra a un equipo pequeño, difiere bastante de la metodología explicada anteriormente para una desarrolladora grande. El proceso de desarrollo llevado a cabo para este videojuego es el siguiente.

Preproducción

- 1 Creación del concepto base del videojuego a través de diversas ideas y del brainstorming.
- 2 Elaboración de la historia sobre la que se sustenta el argumento del videojuego.
- 3 Definición de los objetivos y mecánicas del juego.
- 4 Definición de los personajes.
- 5 Elaboración del documento general de diseño (GDD)
- 6 Elaboración del documento técnico de diseño (TDD)
- 7 Reuniones de trabajo con la artista gráfica.
- 8 Reuniones de trabajo con el compositor musical.
- 9 Reuniones de trabajo con la programadora.

Producción

- 1 Creación del arte gráfico.
 - 1.1 Sprites.
 - 1.2 Fondos.
 - 1.3 Escenarios.
- 2 Composición del fondo sonoro.
 - 2.1 Músicas.
 - 2.2 Efectos sonoros.
- 3 Creación de los scripts de código.
- 4 Implementación. Esta parte de la producción supone la inclusión en el juego del arte gráfico, sonoro y la programación (es habitual realizar la programación de forma paralela a la implementación, pues ambas están interrelacionadas).

Postproducción:

1. Exportación a formato ejecutable.
2. Testeo. Prueba de juego.
3. Rectificación de errores.
4. Documento conceptual (“pitch”).
5. Presentación ante el Tribunal de la Versión Beta del juego.

5. CONCLUSIONES

Como ya vimos al comienzo de este trabajo, los videojuegos son obras audiovisuales, ya que transmiten un mensaje a través de la imagen y el audio, los cuales son tan relevantes en ellos como en una película o serie de televisión, con la peculiaridad de poseer una interactividad innata (a diferencia del resto de medios, salvo Internet). Cada vez están más presentes en la sociedad, y en menos de 40 años de vida han alcanzado el nivel de otros medios más longevos (actualmente, muchos videojuegos, ya sea por presupuesto, guión o calidad gráfica, se podrían calificar como “películas interactivas”), además de realizarse más estudios acerca del sector en los últimos años. Aunque todo aquello de lo que se abusa tiene efectos negativos, un uso moderado de videojuegos puede aportar beneficios al jugador, tanto físicos como psicológicos, e incluso sociales (como mejora de reflejos, memoria, coordinación mano-ojo, reducción de estrés, trabajo en equipo, etc.).

En cuanto al desarrollo de la demo desarrollada para este trabajo, ha supuesto un reto por diversos motivos:

- **Escasez de tiempo:** por diversos motivos, sólo tuvimos un plazo de trabajo de 2 meses, lo cual es un tiempo excesivamente escaso para el desarrollo de un videojuego (un equipo profesional tarda en torno a un año, como mínimo en realizar un videojuego, ascendiendo a más de 2 o 3 años en el caso de los videojuegos triple A).
- **Falta de experiencia:** todos los miembros del equipo de desarrollo, compuesto por cinco personas, carecemos de experiencia previa en el desarrollo de videojuegos.
- **Falta de presupuesto:** el trabajo ha sido realizado sin más recursos que los ordenadores personales de los miembros del equipo y el uso de software gratuito.
- **Programación:** para la realización de este trabajo consulté un par de cursos que encontré en Internet relativos a programación para aprender a trabajar con scripts. Sin embargo, el aprendizaje de un lenguaje de programación es una tarea larga y compleja, por lo que, aun conociendo las bases del lenguaje y contando con la ayuda de dos programadores, la programación del videojuego fue el mayor obstáculo al que tuve que hacer frente.

Como ya he mencionado, el equipo de trabajo ha estado formado por cinco miembros: dos programadores, un artista gráfico, un artista sonoro y un director (encargado también del montaje y la implementación). Dicho esto está claro que el desarrollo de videojuegos es una labor multidisciplinar. Sin embargo, éstos no son los únicos puestos a cubrir en un equipo de desarrollo pues, de forma similar a la televisión o el cine, hay un gran número de puestos, muchos de ellos ligados a la carrera de Comunicación Audiovisual: director, guionistas, animadores, diseñadores, editores de vídeo y audio, músicos, sonidistas, dobladores, publicistas, etc.

En definitiva, este ha sido un trabajo de elevada dificultad. Sin embargo, tanto mi equipo de trabajo como yo hemos adquirido bastantes conocimientos y experiencia, además de haber sentado las bases de lo que, con dedicación y algo de suerte, podría ser una futura carrera profesional orientada al desarrollo de videojuegos.

5. Fuentes bibliográficas y webgráficas

Bibliografía

DOWNEY, Allen. Think Python: How to think like a computer scientist. Needham, Massachusetts: Green Tea Press, 2012.

TOST, Gina y BOIRA, Oriol. Vida extra: los videojuegos como no los has visto nunca. Barcelona: Grijalbo, 2015.

MOYA FERNÁNDEZ, Francisco et al. Desarrollo de videojuegos 3: técnicas avanzadas. Ciudad Real: Universidad de Castilla la Mancha, 2012.

Webgrafía

Historia de los videojuegos.

<http://www.elotrolado.net/wiki/Historia_de_los_videojuegos#Los_inicios_de_los_videojuegos_-__.28Resumen.29> [Consulta: enero 2015].

1952-1972 El origen de los Videojuegos: desde OXO hasta PONG.

<<http://www.scenebeta.com/tutorial/1952-1972-el-origen-de-los-videojuegos-desde-oxo-hasta-pong>> [Consulta: enero 2015].

ARCOS, Eduardo. Historia de la tecnología: Magnavox Odyssey, la primera videoconsola de la historia. <<http://hipertextual.com/2011/08/magnavox-odyssey-primera-videoconsola>> [Consulta: enero 2015].

MORA, Marçal. Magnavox Odyssey.

<<http://www.retromaquinitas.com/index.php/consolas/primera-generacion/odyssey>> [Consulta: enero 2015].

Third-party. <<http://es.wikipedia.org/wiki/Third-party>> [Consulta: febrero 2015].

Documentación en videojuegos:

<<http://eldocumentalistaudiovisual.com/2015/02/06/documentacion-en-videojuegos-documento-de-diseno-gdd/>> [Consulta: marzo 2015]

Géneros de videojuegos.

<http://es.videojuegos.wikia.com/wiki/G%C3%A9neros_de_videojuegos> [Consulta: mayo 2015].

3DJUEGOS S.L. Géneros de los videojuegos.

<<http://www.3djuegos.com/foros/tema/1110597/0/generos-de-los-videojuegos-disponible/>> [Consulta: mayo 2015].

Géneros de los videojuegos.

<http://www.elotrolado.net/wiki/G%C3%A9neros_de_los_videojuegos> [Consulta: mayo 2015].

¿Los videojuegos curan enfermedades?. <<http://www.infotechnology.com/historico/Los-videojuegos-curan-enfermedades-20090813-0001.html>> [Consulta: junio 2015].

El Glosario del Gamer (Semana del 2 de Septiembre de 2013).

<<http://www.chutakoopas.com/post/60117001552/el-glosario-del-gamer-semana-del-2-de-septiembre>> [Consulta: agosto 2015].

¿Qué son los Serious Games? <<http://suite101.net/article/que-son-los-serious-games-a68486#.VeHCplSqqkr>> [Consulta: agosto 2015].

TURNER BROADCASTING SYSTEM, INC. Videojuegos reducen el estrés.

<<http://www.cnnexpansion.com/economia-insolita/comprobado-videojuegos-reducen-estres>> [Consulta: junio 2015].

DIARIO EL CORREO, S.A. Un estudio concluye que los juegos violentos ayudan a eliminar estrés, actitudes hostiles y la depresión. <<http://www.vadejuegos.com/noticias/2010/07/14/un-estudio-concluye-que-los-juegos-violentos-ayudan-a-eliminar-estres-actitudes-hostiles-y-la-depresion-235412.html>> [Consulta: junio 2015].

IBTIMES CO., LTD. Teenage Gamers Beat Medical Professionals When Performing Robotic Surgery. <<http://www.ibtimes.co.uk/teenage-gamers-beat-medical-residents-robotic-surgery-405747>> [Consulta: junio 2015].

UNIVERSITY OF ROCHESTER. Daphne Bavelier: Your Brains on Action Games. <<http://www.rochester.edu/news/video/bavelier-tedx-2012/>> [Consulta: junio 2015].

QUEEN MARY UNIVERSITY OF LONDON. Playing video games can boost brain power. <<http://www.qmul.ac.uk/media/news/items/se/112578.html>> [Consulta: junio 2015].

HOLADOCTOR INC. Videojuegos ¿buenos para el cerebro? <<http://holadoctor.com/es/cerebro-y-sistema-nervioso/videojuegos-buenos-para-el-cerebro>> [Consulta: junio 2015].

MARCANO LÁREZ, BEATRIZ. El Trabajo en equipo en los clanes de Call of duty. <<http://beatrizmarcano.blogspot.com.es/2013/09/el-trabajo-en-equipo-en-los-clanes-de.html#.VY7g9vmqqko>> [Consulta: junio 2015].

JIMÉNEZ, DANIEL. Estudios Psicológicos: Los videojuegos no causan violencia real. <<http://www.infoconsolas.com/secciones/videojuegos-y-sociedad/estudios-psicologicos-los-videojuegos-no-causan-violencia-real>> [Consulta: junio 2015].

PEGI S.A. Información sobre PEGI ¿Qué es PEGI? <<http://www.pegi.info/es/index/id/91/>> [Consulta: mayo 2015].

CODECADEMY. Python. <http://www.codecademy.com/en/tracks/python?jump_to=4fd55e66b280710003045fb5> [Consulta: julio 2014].

“Curso en español de C# para Unity (Nivel 1)”. Youtube <<https://www.youtube.com/playlist?list=PLREdURb87ks31G-kvEoIOYBoUwJMfPkzg>> [Consulta: septiembre 2014].

“Creando un juego "Infinite Runner" en 2D con Unity 4.3 (Tutorial)”. Youtube <<https://www.youtube.com/playlist?list=PLREdURb87ks2qkD9svvIiwYwN35FZ3Afv>> [Consulta: septiembre 2014].

“Curso Unity3d | Interfaz de Unity3d a fondo | Parte 2”. Youtube
<<https://www.youtube.com/watch?v=x-P-H8b0x3A>> [Consulta: noviembre 2014].

“Historia de los Videojuegos (Documental)”. Youtube
<<https://www.youtube.com/watch?v=GQAZIG1mEX0>> [Consulta: noviembre 2014].

“Características 2D en Unity 4.5-10 Programando el Movimiento del Personaje 2D”. Youtube
<<https://www.youtube.com/watch?v=MxmNOLjoJKg>> [Consulta: enero 2015].

“Creating 2D Games in Unity 4.5 #15 - Camera Controller”. Youtube
<<https://www.youtube.com/watch?v=u67fbxe8xxY>> [Consulta: enero 2015].

“Creating 2D Games in Unity 4.5 #16 - Background Parallax”. Youtube
<https://www.youtube.com/watch?v=cpt6MugGloM&index=16&list=PLt_Y3Hw1v3QSFdh-evJbfkxCK_bjUD37n> [Consulta: enero 2015].

“Videojuegos, trabajo en equipos y transferencia de aprendizajes”. Youtube
<<https://www.youtube.com/watch?v=4mRO36pFPyA>> [Consulta: mayo 2015].

6. Sumario

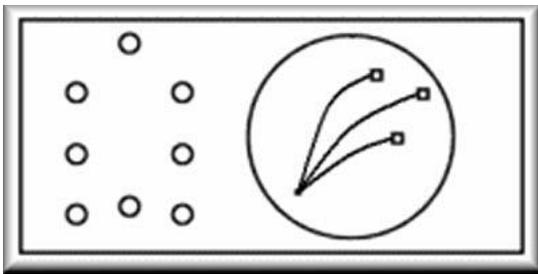
1. INTRODUCCIÓN	3
2. OBJETIVOS Y METODOLOGÍA	4
3. ANÁLISIS TEÓRICO	5
3.1.Historia y evolución	5
3.2.Tipos de videojuegos	13
3.3.Estudios sobre videojuegos	23
4. DESARROLLO DE UN VIDEOJUEGO	27
4.1.Preproducción	27
4.2.....Producción	29
4.3.Postproducción	32
4.4.Memoria de producción	34
5. CONCLUSIONES	43
5. Fuentes bibliográficas y webgráficas	45
6. Sumario	49
7. Anexos	50
Anexo 1: Consolas y videojuegos destacados a lo largo de la historia	50
Anexo 2: Documento de Diseño del Juego (GDD)	59
Anexo 3: Documento Técnico de Diseño (TDD)	63
Anexo 4: Nociones básicas de programación en C#	67

7. Anexos

Anexo 1: Consolas y videojuegos destacados a lo largo de la historia

Las siguientes páginas incluyen una serie de imágenes que muestran y algunos de los videojuegos y las consolas más relevantes de cada década desde la aparición de los videojuegos. Este anexo sirve como complemento al capítulo de este trabajo referente a la historia de los videojuegos.

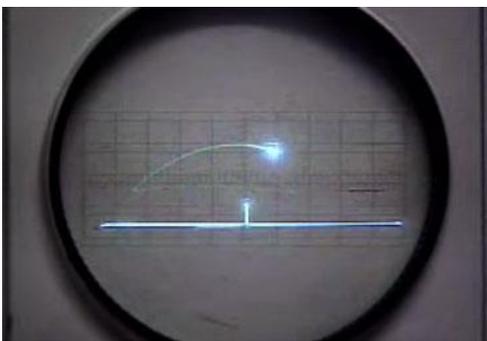
Prehistoria



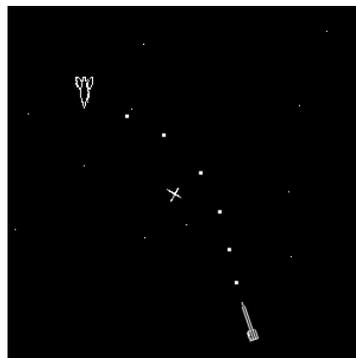
Lanzamiento de Misiles



OXO



Tennis for Two



Spacewar



Galaxy Game



Computer Space

Años 70

Consolas



Magnavox Odyssey



Home Pong



Channel F



VCS

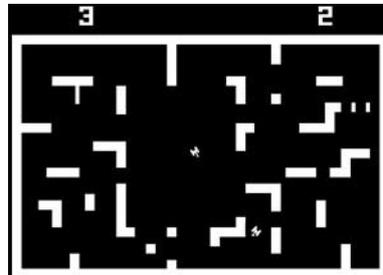


Nintendo Tv Game 6

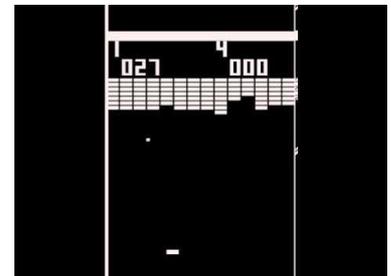
Videojuegos



Pong



Tank



Breakout



Space Invaders



Asteroids



Pacman

Años 80

Consolas



Game & Watch



Intellivision



Commodore 64



Colecovision



ZX Spectrum



NES



Sega Mega Drive/
Genesis

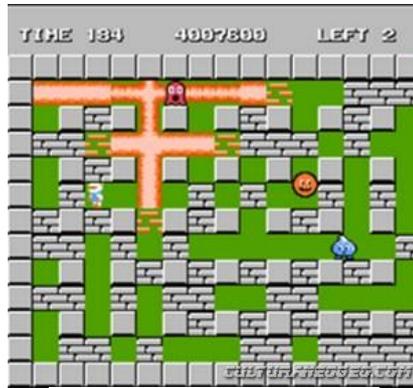


Game Boy

Videojuegos



Donkey kong



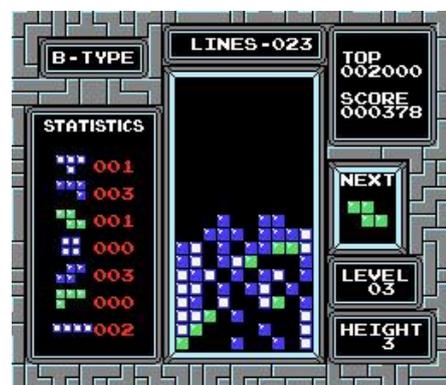
Bomberman



Super Mario Bros.



King's Quest I



Tetris



The Legend of Zelda



Metroid



Castlevania



Metal Gear



Maniac Mansion



MegaMan



La Abadía del Crimen



Final Fantasy

Años 90

Consolas



Super Famicom/ Super NES



Neo Geo



Sega Saturn



PlayStation



Nintendo 64



Dreamcast

Videojuegos



The Secret of Monkey Island



Sonic



The Legend of Zelda: A Link to the Past



Street Fighter II



Mortal Kombat



Alone in the Dark



Doom



FIFA International Soccer (FIFA 94)



Warcraft



Super Mario 64



Resident Evil



Final Fantasy VII



Age of Empires



The Legend of Zelda: Ocarina of Time



Metal Gear Solid



Gran Turismo



Tomb Raider III



Crash Bandicoot

Últimos Años

Consolas



PlayStation 2



Game Boy Advance



Xbox



GameCube



Nintendo DS



PSP (PlayStation Portable)



Xbox 360



PlayStation 3



Wii



Nintendo 3DS



Xbox One



PlayStation 4



Wii U

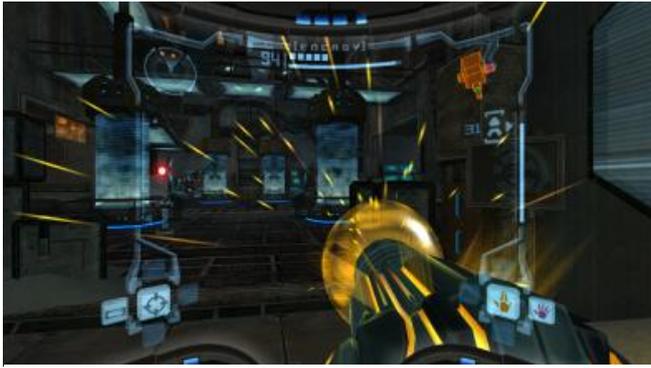
Videjuegos



Super Mario Sunshine



Devil May Cry 3



Metroid Prime



Grand Theft Auto: Vice City



League of Legends



Uncharted 2



The Last of Us



Portal 2

Anexo 2: Documento de Diseño del Juego (GDD)

Concepto

- **Título:** Gaia
- **Género:** Plataformas, puzles.
- **Plataforma:** PC.
- **Versión del GDD:** 3
- **Jugadores:** Un jugador. Aunque no hay una limitación en este aspecto, el público objetivo se establecerá en el rango de jugadores casuales de entre 15 y 30 años.
- **Historia:** Gaia, el espíritu del bosque y de la vida, está presente en todos los animales, plantas y, en general, en todos los seres vivos de la Tierra... salvo en los humanos, pues éstos han rechazado a Gaia, destruyendo la naturaleza, torturando a otros animales y transformando todo a su alrededor para convertirse en sus propios “dioses”.

El juego contará la historia del viaje de Gaia para reencontrarse con el ser humano y restablecer así el vínculo del hombre con la naturaleza.

- **Objetivos:** El objetivo del jugador será avanzar por el escenario para llegar al final de cada nivel, sorteando los diversos obstáculos que se le presenten.

- **Reglas:** El jugador controlará a Gaia, quien sólo se podrá desplazar de izquierda a derecha (no puede saltar). Sin embargo, Gaia posee la habilidad de controlar animales, además de tener las propiedades del fuego (ya que es una llama). El jugador deberá aprovechar estas habilidades para encontrar el camino de salida, quemando objetos, iluminando zonas o controlando animales que se encuentren por la zona.

Además de lo mencionado en el párrafo anterior, en algunos niveles nuestro protagonista deberá desenvolverse en un entorno sin más luz que la de su propia llama, por lo que es importante aprovechar los objetos del entorno para iluminar la zona (como antorchas) y escuchar los sonidos de alrededor, que nos darán pistas de la situación de algunos peligros.

- **Diseño de niveles:** Habrá 4 niveles o “mundos”. Al final de cada uno de ellos habrá un enemigo final o “jefe” al que el jugador deberá enfrentarse. Los 4 mundos diferentes serán:

- o **Bosque:** Es en el que empezamos la partida. En él podremos controlar conejos, lobos y búhos, tendremos que quemar plantas para sortear algunos obstáculos o abrir el

camino, y nos toparemos con leñadores como enemigos. Nos centraremos en este mundo para la realización de la demo del juego.

- **Cueva:** este mundo estará a oscuras, por lo que el jugador tendrá que buscar formas de iluminar la zona y prestar atención a los sonidos de su alrededor para evitar obstáculos.

- **Robot:** el 3º mundo será una ciudad en forma de robot gigante, y representará el núcleo de la civilización humana. El jugador tendrá que evitar obstáculos eléctricos, robots, cámaras de vacío...

- **Vertedero:** Este mundo representa la máxima expresión de la decadencia que provoca el ser humano en la naturaleza y será el último nivel del juego.

Interfaces y estados del juego

Además de la pantalla del juego, el proyecto dispondrá de 6 interfaces diferentes: Menú principal, Opciones, Pausa/opciones (dentro del juego), Game Over, Historia, Controles y Créditos.

- Menú principal: Es la primera interfaz que verá el jugador al ejecutar el juego. Presenta 3 botones:

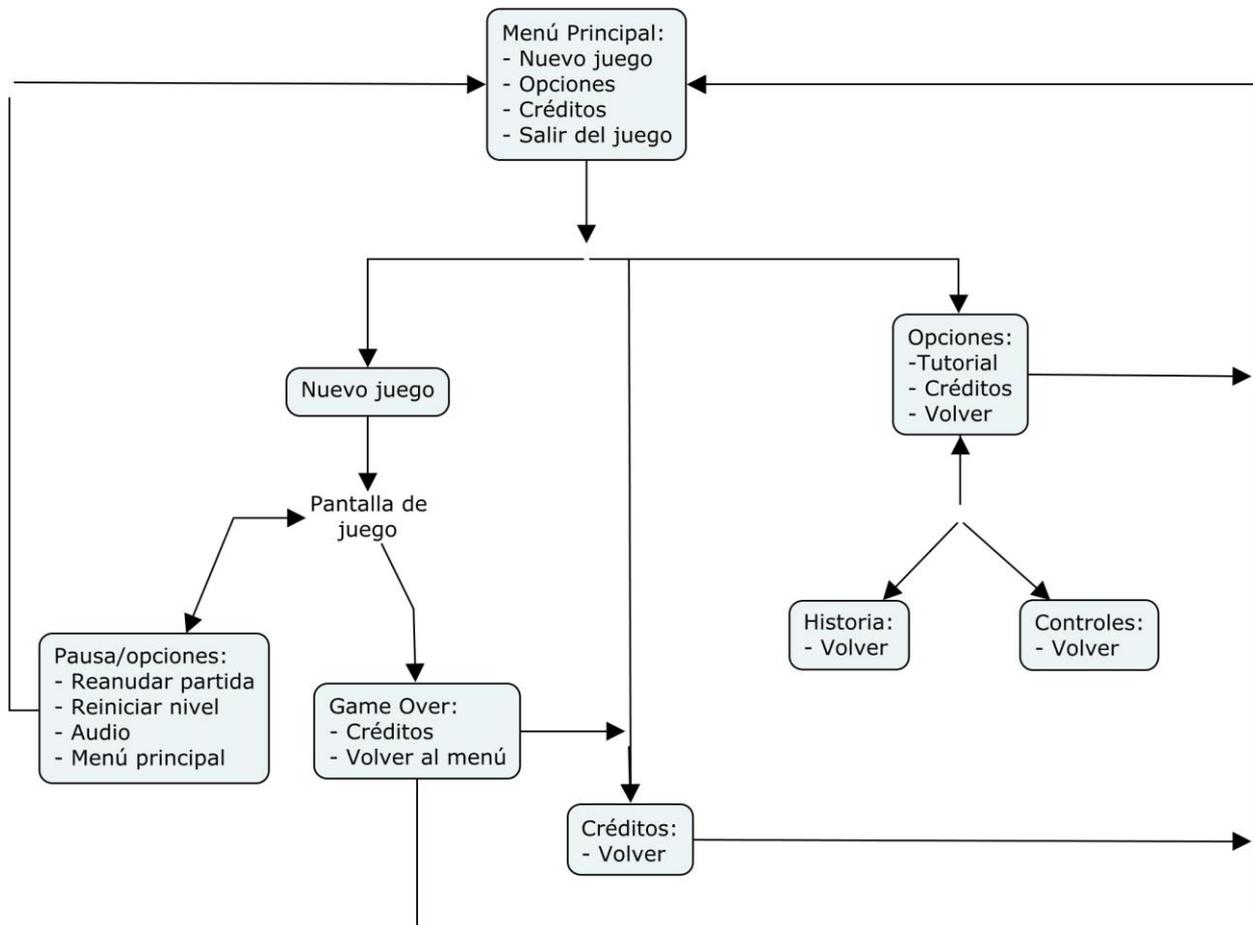
- Nuevo juego: Lleva al comienzo del primer nivel del juego.
- Opciones: Lleva al menú principal de opciones.
- Créditos: Lleva a la pantalla de créditos.

- Opciones: Permite acceder a las pantallas de historia y de controles.
- Pausa/opciones: A este menú se accede desde la pantalla del juego. Permite modificar el volumen del audio y reiniciar el nivel, así como reanudar la partida o salir de la misma.

- Game Over: Esta ventana aparece al completar la demo del juego. Contiene un botón para acceder al menú de créditos y otro para acceder al menú principal.

- Controles: Muestra una breve explicación de los controles del juego.
- Créditos: Relación de todas las personas que han intervenido en la producción del videojuego.

La siguiente imagen muestra un esquema de las interfaces, indicando con flechas a cuáles conduce cada una de ellas.



8 Diagrama de interfaces

Personajes y objetos

Personajes controlables: El protagonista es la llama Gaia, pero como se mencionó antes, podrá controlar algunos animales para poder avanzar en su camino (si el animal que está siendo controlado muere, Gaia también morirá).

- **Gaia:** Es el protagonista de la historia y el personaje al que controlaremos desde el principio.
- **Conejo:** Nos permitirá saltar, correr y pasar por espacios reducidos.
- **Lobo:** Sus principales ventajas son la posibilidad de saltar, cavar y atacar.
- **Pájaro:** Ofrece al jugador la habilidad de volar.

Enemigos: Si cualquiera de los enemigos ve al protagonista, se lanzarán a atacarle, por lo que, aunque algunas veces el jugador podrá defenderse, otras veces deberá evitar que le vean.

- **Leñador:** Su arma es un hacha, así que sus ataques son cuerpo a cuerpo.
- **Cazador:** Con su escopeta puede atacar a distancia (no aparece en la demo).
- **Ente Oscuro:** Es el jefe del bosque. Perseguirá a Gaia sin descanso y es mortal al contacto con personaje controlado. Para vencerle hay que lanzarle una roca y, en cuanto caiga al suelo, quemarle (morirá tras quemarle todas las patas).
 - **Minero:** Es capaz de abrirse camino a través de las paredes de roca con su pico, además de atacar con él (no aparece en la demo).
 - **Dinamitero:** Puede lanzar explosivos o poner trampas (no aparece en la demo).
 - **Humano:** Es el objetivo de Gaia, pues le busca para volver a unirse a él, pero el humano siempre le rehúye. Al final Gaia tendrá que enfrentarse a él (no aparece en la demo).

Anexo 3: Documento Técnico de Diseño (TDD)

Diseño Artístico

Biblia de la Historia:

Gaia es el dios de la naturaleza y está presente en todos los seres vivos de la Tierra. Sin embargo, el ser humano, al ver potencial superioridad, decidió usar la naturaleza a su favor, rompiendo así su vínculo con Gaia y perjudicando gravemente al resto de la naturaleza. Viendo los destrozos causados por el hombre, Gaia decide ir en su busca para restablecer ese vínculo y devolver la armonía a la naturaleza.

El viaje comienza en el bosque, el hogar de Gaia, quien se encuentra un panorama desolador: el bosque está devastado. Al ver esto, Gaia decide ir en busca de la causa del estado del bosque. Tras una larga búsqueda, encuentra a los responsables del desastre: humanos. Un grupo de leñadores están destrozando el bosque, mientras que algunos cazadores se dedican a matar a cualquier animal que encuentran a su paso. Gaia decide entonces restablecer el vínculo con los humanos. Sin embargo, son muy peligrosos, así que decide ir en busca de su nuevo “hábitat”, la ciudad, para poder influirles desde su núcleo. A lo largo de su viaje deberá evitar a los humanos que acechan la zona y, si puede, neutralizarlos para que no causen más daños. El camino le llevará a la entrada de una cueva, la cual está custodiada por un ente oscuro (el jefe de la zona), al que Gaia deberá enfrentarse para poder avanzar.

Diseño de la mecánica del Juego

Interactuación con el juego:

La navegación por las diferentes interfaces y menús del juego será la explicada en el GDD, siguiendo el esquema ya mostrado en la imagen 8, en el anexo 2.

Durante el juego no aparecerá ninguna GUI (Interfaz Gráfica de Usuario), ya que no es necesaria información alguna de salud, nivel, armas, etc.

Los controles del juego se explicarán en el apartado siguiente, junto a toda la información sobre personajes, objetos y enemigos.

Cada uno de los niveles estará compuesto por dos escenas: la escena del nivel y la del jefe de la zona.

Diseño del comportamiento, habilidades y otros elementos importantes de los personajes y del entorno:

Personajes controlables:

Gaia. Es el personaje principal y sólo podrá desplazarse hacia izquierda y derecha. Aparte de esto, también podrá quemar diversos objetos y controlar animales para abrirse paso a través del escenario. Muchas zonas del juego estarán a oscuras, por lo que el jugador deberá aprovechar la luz de Gaia, así como prender antorchas y otros elementos para poder ver el entorno y avanzar con mayor facilidad. El jugador manejará a Gaia con los siguientes controles:

- Flecha de dirección derecha: desplazamiento a la derecha.
- Flecha de dirección izquierda: desplazamiento a la izquierda.
- Tecla “z”: controlar a un animal.
- Tecla “x”: prender objetos.

Conejo. Personaje controlable por Gaia. Es más pequeño que Gaia y cualquiera de los otros animales controlables, lo que nos permitirá pasar por zonas estrechas. Sus controles son:

- Flecha de dirección derecha: desplazamiento a la derecha.
- Flecha de dirección izquierda: desplazamiento a la izquierda.
- Barra espaciadora: saltar.
- Tecla “z”: abandonar cuerpo.

Lobo. Sus controles son:

- Flecha de dirección derecha: desplazamiento a la derecha.
- Flecha de dirección izquierda: desplazamiento a la izquierda.
- Barra espaciadora: saltar.
- Tecla “x”: atacar.
- Tecla “z”: abandonar cuerpo.

Búho. Sus controles son:

- Flecha de dirección derecha: desplazamiento a la derecha.
- Flecha de dirección izquierda: desplazamiento a la izquierda.
- Flecha de dirección superior: saltar/ desplazamiento ascendente (durante el vuelo).
- Flecha de dirección inferior: desplazamiento descendente (durante el vuelo).
- Barra espaciadora: volar/ aterrizar (durante el vuelo).
- Tecla “z”: abandonar cuerpo.

Objetos del entorno:

Arbusto. Este objeto se puede quemar para abrir un camino.

Antorcha. Gaia puede encenderlas para iluminar la zona.

Cuerda. Pueden servir para prender un objeto lejano o para transportar a Gaia a una zona lejana.

Roca. Este objeto servirá a Gaia como plataforma para llegar a zonas inaccesibles, como arma para derribar enemigos o, en general, para superar diversos obstáculos.

Enemigos:

Los enemigos atacarán a Gaia si éste está dentro de su campo de visión, por lo que cuando el protagonista esté indefenso deberá avanzar evitando que le vean. Los enemigos no pueden saltar.

Leñador. Va armado con un hacha, por lo que sus ataques son a corta distancia. Gaia morirá (en cualquiera de sus formas) si es golpeado por su hacha.

Cazador. Va armado con una escopeta, lo que le permite realizar ataques a larga distancia. Gaia morirá (en cualquiera de sus formas) si es alcanzado por uno de sus disparos. No aparece en la demo.

Ente Oscuro: Es el jefe del bosque. Perseguirá a Gaia sin descanso y es mortal al contacto con personaje controlado. Para vencerle hay que lanzarle una roca y, en cuanto caiga al suelo, quemarle (morirá tras quemarle todas las patas).

Trampas:

Agua. Si Gaia toca el agua, morirá, ya que es una llama. El agua puede aparecer de diversas formas: ríos, cascadas, lagos... En los casos en los que el agua fluya, ésta emitirá sonido, indicando así su presencia.

Caídas. Están determinadas por el terreno del escenario. Cuando Gaia caiga por una abertura que le haga salir de plano, morirá.

Elementos a programar:

- Personajes controlables (los controles están especificados en las páginas anteriores).
- Cambio de personaje (y aparición-desaparición de la llama).
- Enemigos (ataque, movimiento, muerte, etc.).
- Objetos del entorno.
- Muerte del jugador.
- Aparición y desaparición de los niveles.
- Audio.
- Menús.
- Opciones de audio.

Anexo 4: Nociones básicas de programación en C#

A continuación se exponen unas nociones básicas para manejar el lenguaje C#, con el que he realizado el videojuego adjunto a este documento. Con el objetivo de diferenciar con mayor facilidad las explicaciones de la correcta sintaxis del lenguaje programático, ésta última la represento con texto de color naranja.

Antes de nada es importante indicar dos herramientas bastante útiles para programar:

`Debug.Log ()` → Esto permite ejecutar en la consola la variable incluida en el paréntesis.

`//` → Esto sirve para realizar comentarios de una línea. El programa no lo reconocerá como código ejecutable y es muy útil para hacer anotaciones explicativas u orientativas acerca del código.

`/*` → Con esto se pueden realizar comentarios multilínea (o comentarios dentro de una línea de código). Se cierra con `*/`.

Tipos de variables

Existen cuatro tipos de datos de variables:

- `int` → Son números enteros. Se representan simplemente con el número del valor correspondiente (1, 5, 17, 450, -43, etc).
- `float` → Indica números reales. Se representan con un punto que separe la parte entera de la parte decimal y con una “f” al final (3.1415f; 5.32f; -54.12f; etc.).
- `string` → Es una cadena de texto. Para representarla, se escribe el texto entre comillas (“¡Hola mundo!”). Si incluimos números dentro de las comillas, el programa lo interpretará como texto, no como valores numéricos. De esta forma, no es lo mismo `2*3+15` que `“2*3+15”`, ya que el primero daría como resultado 21, mientras que el segundo daría `2*3+15` (no se resuelve porque no es una operación matemática, sino una cadena de texto).

`Debug.Log (2*3+15);`

21

`Debug.Log (“2*3+15”);`

`2*3+15`

Para usar comillas dentro de una cadena de texto se escribe `\"`. A su vez, para usar una barra, se escribirán dos barras: `\\`. También podemos hacer un salto de línea usando `\n`.

- **bool** → Indica un valor lógico. A diferencia de los tipos de variables anteriores, que posibilitan una cantidad ilimitada de valores, las variables de tipo bool (o booleanas) sólo presentan dos posibilidades: **true** o **false**.

Una variable se establece de la siguiente forma: “tipo de variable” “nombre de variable” = “valor de la variable”;. El símbolo “;” se pone siempre para cerrar una línea de código (es muy importante, y su ausencia es una causa frecuente de errores de código). Ejemplos:

```
int puntos = 0;
```

```
float precio = 13.5f;
```

```
string nombre = “David”;
```

```
bool personajeVivo = true;
```

Para definir más de una variable del mismo tipo en una sola línea se usa la coma, y para modificar el contenido de una variable tras ser creada, se le vuelve a otorgar un valor sin indicar el tipo de variable (el último valor asignado será el que adquiera, ya que el código se ejecuta en el orden en que se escribe):

```
float precioCamiseta = 15.80f, precioPantalon = 19.95f, precioCalcetines = 6.45f;
```

```
precioPantalon = 17.95f;
```

Arrays

Un array es un conjunto de elementos del mismo tipo. Para crear uno hay que seguir la siguiente estructura: “tipo de variable” [] “nombre del conjunto” = **new** “tipo de variable” [“número de elementos”] {“valor1”, “valor2”, etc};. Pueden no incluirse los valores de las variables, y en ese caso adoptarían el valor predeterminado del tipo de variable (a su vez, si se ponen los valores, no es necesario poner el número de elementos).

```
int [] puntos = new int [10];
```

```
int [] clasificacion = new int [5]{1, 2, 3, 4, 5};
```

Operadores (aritméticos, lógicos y de comparación)

Podemos realizar operaciones matemáticas con los operadores aritméticos: `+`, `-`, `*`, `/` y `%` (con este último obtendremos el resto de una división). Se pueden usar para realizar operaciones entre números, dentro de una variable, pero también se pueden usar para modificar el valor de una variable si se anteponen al signo `=`. Por otro lado, también se puede sumar o restar una unidad poniendo dos veces el operador correspondiente.

```
int total = 21;
```

```
total += 7;
```

```
total --;
```

En este ejemplo, el valor de total acaba siendo 27.

También disponemos de operadores lógicos y operadores de comparación:

&	AND: La acción incluye a los dos elementos unidos.	<pre>true && true = true true && false = false</pre>
	OR: La acción afecta sólo a uno de los elementos.	<pre>true false = true true true = true false false = false</pre>
	NOT: La acción no afecta a ninguno.	<pre>true ! true = false true ! false = false false ! false = true</pre>

Para cualquier tipo de variable	=	Igual que
	!=	Distinto que

Para variables numéricos (int y float)		Mayor que
	=	Mayor o igual que
		Menor que
	=	Menor o igual que

Además de los anteriores, existe un operador más, el operador `?`, que funciona de la siguiente manera:

```
int nivel = true ? 1:0;
```

Si se cumple que la variable es true, tomará el valor 1. En caso contrario, la variable nivel tomará el valor 0.

Condicional if

Al igual que el resto de partículas que veremos a continuación, el condicional `if` tiene un significado similar al que tendría la palabra en inglés, es decir, al usarlo haría el siguiente efecto: si se cumple la condición X, se producirá la acción Y. Además de la condición `if`, es necesario especificar qué pasaría en el resto de casos, lo cual se establece con `else`. Si se quieren especificar más condiciones, es posible hacerlo incluyendo condiciones mediante `else if`, entre las condiciones `if` y `else`.

```
if (usuario = David) Debug.Log ("Hola, David");
```

```
else if (usuario = Antonio) Debug.Log ("Hola Antonio");
```

```
else Debug.Log ("Usuario desconocido");
```

Al ejecutar el código del ejemplo, si se cumple que usuario es David, se mostrará la frase "Hola David", si usuario es Antonio, la frase "Hola Antonio", y si no es ninguno de estos dos, se mostrará la frase "Usuario desconocido".

Switch

Esta partícula es bastante similar al condicional if. Establece distintas acciones según el caso que se cumpla para una variable.

```
Switch (tipoObjetoTocado){  
  
    case 0:  
  
    case 2:  
  
        Debug.Log (“has ganado una vida”);  
  
        vidas ++;  
  
        break;  
  
    case 1:  
  
        Debug.Log (“Has perdido una vida”);  
  
        vidas --;  
  
        break;
```

Si se ponen dos casos seguidos (como el case 0 y el case 2) el mismo código afectará a esos casos.

while/ do-while

El bucle **while** llevará a cabo la acción que se especifique siempre que se cumpla la condición, es decir, si se cumple la condición y tras ejecutarse la acción sigue cumpliéndose, volverá a ejecutarse (haciendo un bucle). Por otro lado, **do-while** funciona igual, con la diferencia de que la acción se ejecuta siempre la primera vez, y luego comprueba si se cumple la condición. En ambos casos es importante tener cuidado y asegurarse que la condición no pueda cumplirse siempre para impedir que la condición entre en un bucle infinito.

```
int contador = 1;  
  
while (contador <= 10){  
  
    Debug.Log(contador);  
  
    contador++;
```

```
int contador = 1;  
  
do{  
  
    Debug.Log(contador);  
  
    contador++;
```

```
    }                               }while (contador <= 10);
```

```
Debug.Log ("Finalizado")
```

for

El bucle for es bastante similar al bucle while, solo que en este caso la variable a la que afecta la condición (el elemento de iniciación) es creada en el bucle. A su vez, la operación que se usa para modificar el valor de la variable se especifica también al comienzo del bucle (en los ejemplos anteriores sumaba 1 unidad al contador para que llegara un momento en que incumpliera la condición). La estructura es la siguiente:

```
for (inicialización; condición; operaciones finales){  
    Instrucciones del bucle;  
}
```

Ejemplo:

```
for (int contador =1; contador<= 10; contador++){  
    Debul.Log (contador);  
}
```

Para incluir más de un elemento de iniciación, condición u operaciones finales se separan con comas. Sólo pueden especificarse variables de inicialización del mismo tipo, pero se pueden usar otras variables creadas con anterioridad.

foreach

Sirve para recorrer un array de forma más sencilla. El siguiente ejemplo servirá para entender mejor su funcionamiento.

```
int [] valores = {1,4,5,7,13,15};  
foreach (int x in valores) {  
    Debug.Log (x);  
}
```

El resultado de ejecutar este código sería que aparecieran uno por uno todos los números de la variable “valores” en orden. Básicamente el código dice que para cada x (se le puede asignar cualquier nombre) dentro de “valores” muestre x en la consola.

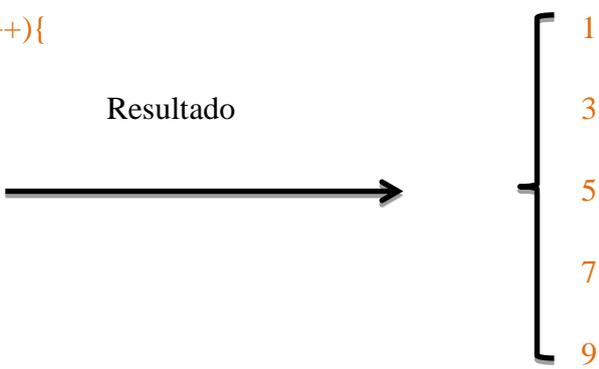
A la hora de usar switch y bucles podemos necesitar las siguientes dos herramientas: **break** y **continue**.

break sirve para cerrar un bucle o un switch (observar el ejemplo del switch).

continue se usa para pasar a la siguiente vuelta del bucle:

```
for (int c=1; c<=10; c++){  
    if (c%2 == 0){  
        continue;  
    }  
    Debug.Log (c);  
}
```

Resultado



1
3
5
7
9

Métodos (o funciones)

Los métodos sirven para encapsular varias líneas de código en una sola instrucción mediante la expresión **void**, lo cual simplifica el trabajo a la hora de usar un mismo código varias veces (sobre todo si es extenso). Pongamos como ejemplo el código anterior.

```
void numerosImpares (x) {  
    for (int c=1; c<=x; c++){  
        if (c%2 == 0){  
            continue;  
        }  
        Debug.Log (c);  
    }  
}
```

}

Así, si queremos que aparezcan los números impares entre 1 y 15 sólo tendremos que escribir:

```
numerosImpares (15);
```

Por último, pero no menos importante, vamos a ver la resolución de errores, concretamente los errores sintácticos, que son aquellos que indica la consola. Cuando haya un error de este tipo en el código, la consola mostrará el nombre del archivo y su localización, la posición del error dentro del código y una descripción del motivo del error de la siguiente forma:

```
Assets/Script/Teoría/T04.cs (10, 23): error CS1002: Expecting “;”
```

Diagrama de anotación del mensaje de error:

- Una línea superior etiquetada como "Fila" apunta a "10" en "(10, 23)".
- Una línea inferior etiquetada como "Columna" apunta a "23" en "(10, 23)".
- Una línea inferior etiquetada como "Causa del error" apunta a "error CS1002: Expecting “;”".
- Una línea inferior etiquetada como "Archivo del error" apunta a "Assets/Script/Teoría/T04.cs".

Éste es uno de los ejemplos más frecuentes de error, ya que el programador se ha olvidado de poner el signo “;” en una línea de código (en este caso, en la línea 10).

A continuación se muestran algunos ejemplos de scripts empleados en el desarrollo del videojuego “Gaia”:

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class Pausa : MonoBehaviour {
5
6     public GameObject CamaraPausa;
7
8     // Use this for initialization
9     void Start () {
10         Time.timeScale = 1.0f;
11     }
12 }
13
14 // Update is called once per frame
15 void Update () {
16     if (Input.GetKeyDown (KeyCode.P)) {
17         CamaraPausa.SetActive (true);
18         Time.timeScale = 0.0f;
19     }
20 }
21 }
22
```

9 Script de pausa

El script de la imagen 9 sirve para activar el menú de pausa, el cual se encuentra en la escena del juego, encuadrada por una cámara (distinta a la cámara principal) que se encuentra desactivada. Según este script, cuando se pulse la tecla “p”, el objeto establecido como “CamaraPausa” (en este caso, la cámara ya mencionada) se activará. Esto hará que, además de la imagen recogida por la cámara principal, se muestre también la del menú de pausa. Por otra parte, se paraliza el juego mediante `Time.timeScale`, que controla la velocidad de la física de la escena, con lo que se “pausa” al darle el valor 0.

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class Salto : MonoBehaviour {
5
6     public float fuerzaSalto = 400f;
7     public bool enSuelo = true;
8     public Transform comprobadorSuelo;
9     float comprobadorRadio = 0.47f;
10    public LayerMask mascaraSuelo;
11    Animator anim;
12
13    void Start () {
14        anim = GetComponent <Animator>();
15    }
16    void FixedUpdate () {
17        enSuelo = Physics2D.OverlapCircle (comprobadorSuelo.position, comprobadorRadio, mascaraSuelo);
18        anim.SetBool ("En Suelo", enSuelo);
19    }
20    void Update () {
21        if(enSuelo && Input.GetButtonDown ("Jump")){
22            rigidbody2D.AddForce(new Vector2 (0, fuerzaSalto));
23        }
24    }
25 }
26
```

10 Script para conservar un objeto

El script de la imagen 10 permite conservar un objeto entre escenas. Al cambiar de una escena a otra, todos los objetos presentes en la primera se borran para crear los objetos de la nueva escena. Mediante la función `DontDestroyOnLoad` se puede conservar el objeto al que se le asigne el script. A su vez el código establece que si el script `Conservar_Objeto` (este mismo script) está presente en otro objeto de la escena, elimina dicho objeto (esto permite evitar que se duplique el objeto a conservar si ya estaba presente en la escena). Este script lo hemos usado para conservar la música del menú principal al cambiar de un menú a otro (mediante otro script se borra si se pulsa el botón de “comenzar partida”, pues la escena del juego tiene su propia música).

```

1 using UnityEngine;
2 using System.Collections;
3
4 public class Conservar_Objeto : MonoBehaviour {
5
6     public static Conservar_Objeto conservarObjeto;
7
8     void Awake() {
9         if (conservarObjeto == null) {
10             conservarObjeto = this;
11             DontDestroyOnLoad (gameObject);
12         } else if (conservarObjeto != this) {
13             Destroy (gameObject);
14         }
15     }
16
17     // Use this for initialization
18     void Start () {
19
20     }
21
22     // Update is called once per frame
23     void Update () {
24
25     }
26 }
27

```

11 Script de salto

Este último ejemplo es el más complejo de estos 3 y sirve para hacer que el personaje controlado salte (este script no se ha usado en el juego, pero su contenido fue incluido en otros scripts que sirven para establecer los controles de los personajes). En este script, el valor de la variable booleana **enSuelo** se establece como **true** si el collider del objeto comprobadorSuelo (que se establecerá en la base del personaje) y el collider de un objeto que tenga asignada la máscara de capa **mascaraSuelo** (está asignada a aquellos objetos que actúen como suelo) están a una distancia igual o menor a la establecida por **comprobadorRadio**. Si **enSuelo** es **true** y se pulsa el botón llamado “**Jump**” (en este caso, la barra espaciadora), se añade una fuerza vertical (en el eje y) al rigidbody del personaje equivalente al valor de **fuerzaSalto**.