

Título del Trabajo

Fin de Grado



Grado en Ingeniería Multimedia

Trabajo Fin de Grado

Autor:

Álex Verdú Miralles

Tutor/es:

Pedro Pernías Peco

Junio 2017



Universitat d'Alacant
Universidad de Alicante

Título del Trabajo de Fin de Grado

Subtítulo del proyecto

Autor

Álex Verdú Miralles

Directores

Pedro Pernías Peco

Lenguajes y Sistemas Informáticos

Nombre Apellido1 Apellido2)

Departamento del cotutor



GRADO EN INGENIERÍA MULTIMEDIA



Escuela
Politécnica
Superior



Universitat d'Alacant
Universidad de Alicante

ALICANTE, 25 de mayo de 2017

Preámbulo

Poner aquí un texto breve que debe incluir entre otras:

“las razones que han llevado a la realización del estudio, el tema, la finalidad y el alcance y también los agradecimientos por las ayudas, por ejemplo apoyo económico (becas y subvenciones) y las consultas y discusiones con los tutores y colegas de trabajo. [?]”

*A mi esposa Marganit, y a mis hijos Ella Rose y Daniel Adams,
sin los cuales habría podido acabar este libro dos años antes*¹

¹Dedicatoria de Joseph J. Roman en “An Introduction to Algebraic Topology”

*Si consigo ver más lejos
es porque he conseguido auparme
a hombros de gigantes*

Isaac Newton.

Índice general

1. Introducción	1
1.1. Emprendimiento y el fenómeno startup	2
1.2. Estado actual del emprendimiento en España	2
1.3. Lean startup	3
1.4. Educación sobre emprendimiento	4
2. Marco Teórico	7
2.1. Actividades en grupo	7
2.2. Juegos de mesa	7
2.3. Videojuegos	9
2.4. Motores de videojuegos	11
3. Objetivos	15
3.1. Objetivo principal	15
3.2. Objetivos específicos	15
3.3. Objetivos secundarios	15
4. Metodología	17
5. Desarrollo	21
5.1. Descripción general	21
5.2. Perspectiva del producto	21
5.3. Funcionalidad del producto	21
5.4. Características de los usuarios	21
5.5. Restricciones	22
5.6. Requisitos específicos	22
5.6.1. Interfaces de usuario	22
5.6.2. Requisitos funcionales	23
5.6.3. Requisitos de usuario	30
5.6.4. Requisitos de interfaz	30
5.6.5. Requisitos de sistema	30
5.7. Requisitos no funcionales	30
5.8. Arquitectura del sistema	30
5.8.1. Arquitectura de Unity3D	31
5.8.2. Arquitectura del juego	37
6. Resultados	45
6.1. Section1	45

6.2. Section2	45
7. Conclusiones	47
7.1. Mejoras y ampliacionesl	47
7.2. Modelo de negocio	47
Bibliografía	49
A. Anexo I	51

Índice de figuras

1.1. Lienzo propuesto por Ash Maurya para el proceso LEAN STARTUP. Fuente: Blog de Javier Megias. http://javiermegias.com/blog/2012/10/lean-canvas-lienzo-de-modelos-de-negocio-para-startups-emprendedores/	3
2.1. Tablero del juego Colonos de Catán. Fuente: Blog del club ERTAI. https://clubertai.wordpress.com/2013/03/10/ludoteca-del-club-los-colonos-de-catan/	8
2.2. Tablero del juego Pandemia. Fuente: Blog Estantería de juegos. http://estanteriadeguegos.blogspot.com.es/2013/12/resena-pandemia.html	9
2.3. Tablero del juego Flea market. Fuente: Tienda online Shopyourway. http://www.shopyourway.com/articles/355984	10
2.4. Captura de pantalla de iPhone con el juego Hipster CEO	12
2.5. Vista del lean canvas en el juego U-startup	12
4.1. Diagrama de Gantt con la distribución temporal y las dependencias de las tareas	19
5.1. Vista del editor de Unity3D donde se pueden ver los componentes de un Gameobject	32
5.2. Diagrama de clases de Unity3D	33
5.3. Bucle de juego del motor Unity3D	36
5.4. Representación del formato utilizado para los ficheros de diálogo	38
5.5. Herramienta inklewriter	39
5.6. Diagrama de flujo nueva conversación	40
5.7. Diagrama de flujo de selección de respuesta	41
5.8. Vista de los escenarios en el inspector de Unity3D	42
5.9. Diagrama de clases del sistema de escenarios	43

Índice de tablas

5.1. My caption	23
5.2. My caption	23
5.3. My caption	23
5.4. My caption	24
5.5. My caption	24
5.6. My caption	24
5.7. My caption	24
5.8. My caption	24
5.9. My caption	25
5.10. My caption	25
5.11. My caption	25
5.12. My caption	25
5.13. My caption	25
5.14. My caption	26
5.15. My caption	26
5.16. My caption	26
5.17. My caption	26
5.18. My caption	26
5.19. My caption	27
5.20. My caption	27
5.21. My caption	27
5.22. My caption	27
5.23. My caption	28
5.24. My caption	28
5.25. My caption	28
5.26. My caption	28
5.27. My caption	29
5.28. My caption	29
5.29. My caption	29
5.30. My caption	29
5.31. My caption	30
5.32. My caption	30

Índice de Listados

5.1. Código de bucle de juego	34
---	----

1. Introducción

El 12 de octubre de 1492 un temerario explorador, Cristobal Colón, y su tripulación pisan la arena de una isla muy al oeste de Europa conocida como Guanahani. Este hecho marca un hito en la historia de la humanidad pues los cambios culturales, económicos, políticos y militares que produce dan lugar a la llamada Edad Moderna.

Colón vio una oportunidad de negocio en el control de las rutas comerciales que unían Europa con Asia pues eran recorridas por miles de comerciantes que traían especias y productos de lujo desde las tierras de Extremo Oriente. El comercio además se realizaba por tierra, lo que lo convertía en un proceso lento, inseguro e ineficiente, además de enriquecedor para los árabes que controlaban las rutas comerciales.

El proyecto tenía un gran interés económico pues como se ha dicho anteriormente, el control de una ruta comercial con Asia era muy lucrativo, pero a su vez tenía un gran riesgo ya que el futuro de la expedición era tremendamente incierto y había pocas posibilidades de encomendarse al vasto océano y volver para contarlo. Debido a esta incertidumbre sobre el retorno de la inversión a Colón le fue complicado encontrar financiación para su proyecto, hasta que finalmente, tras recurrir a varios monarcas y mecenas, los Reyes Católicos le proveyeron de los recursos necesarios para iniciar su aventura.

Se podría considerar a Cristobal Colón como un emprendedor, a pesar de que el término fue usado por primera vez doscientos años después por el economista Richard Cantillon que define al emprendedor como

La persona que paga un cierto precio para revender un producto a un precio incierto, por ende tomando decisiones acerca de la obtención y el uso de recursos, y admitiendo consecuentemente el riesgo en el emprendimiento.

[Navale, 2013, pág 21].

De esta definición se puede apreciar que un emprendedor inicia proyectos y acepta la incertidumbre y el riesgo que ello conlleva, puesto que en caso de desastre es él quien asume la responsabilidad.

La actitud emprendedora ha sido una constante a lo largo de la historia de la humanidad: desde Cristobal Colón hasta Bill Gates, pasando por Leonardo Da Vinci, Henry Ford o Nikola Tesla; hombres y mujeres con coraje han empezado proyectos bajo una idea prometedora y asumiendo grandes riesgos, motivados por la pasión y las perspectivas de éxito.

El emprendimiento es una actividad especialmente necesaria para el progreso de una sociedad pues es un proceso que crea riqueza, innovación y empleo. Los emprendedores

crean productos y servicios revolucionarios que hacen la vida de las personas más fácil, mejorando por ello su calidad de vida. Además suele ser una opción frecuente en épocas de crisis económicas debido a la escasez de empleo.

1.1. Emprendimiento y el fenómeno startup

Cada vez es más frecuente escuchar el término «startup», pequeñas empresas dedicadas al ámbito tecnológico que alcanzan en pocos años grandes cuotas de mercado y se venden por cantidades astronómicas a empresas más grandes.

El fenómeno goza de tanta popularidad que ha inspirado incluso a series televisivas como «Silicon Valley»¹, que narra las aventuras de un grupo de jóvenes ingenieros que crean una «startup» tecnológica y se enfrentan al reto de sobrevivir en un ecosistema hostil como es el mercado; la película «Piratas de Silicon Valley», que narra la historia de enfrentamiento entre Microsoft y Apple; la película «La red social» que cuenta la historia de Mark Zuckerberg y como crea la red social «Facebook».

Llegado a este punto cabe preguntarse: ¿Qué es exactamente una «startup»? Es un error común pensar que las «startup» son simplemente versiones más pequeñas de empresas grandes. En palabras de los gurús del emprendimiento Steve Blank y Bob Dorf, «Una “startup” es una organización temporal en busca de un modelo de negocio rentable, que pueda repetirse y que es escalable» [Blank and Dorf, 2013].

De la anterior definición se puede extraer que una «startup»:

- Es una organización temporal, es decir, el objetivo no es ser siempre una «startup». El objetivo es convertirse en una empresa consolidada.
- No conoce con seguridad cual va a ser su actividad. En su lugar parten de un modelo de negocio temporal que va evolucionando a medida que interactúa con el mercado.
- Busca un modelo de negocio repetible y escalable, que le permita ejecutar dicho modelo de negocio durante un tiempo indefinido y además expandirse.

El emprendimiento es inherente al fenómeno «startup» pues la incertidumbre es un pilar fundamental al crear una de estas empresas, que ni siquiera tienen un modelo de negocio que se pueda asegurar que va a funcionar.

1.2. Estado actual del emprendimiento en España

Si bien el fenómeno «startup» nació en EEUU y es allí donde está más consolidado, en España es una tendencia igualmente extendida. Atendiendo a cifras de financiación «en 2015, las startups españolas lograron financiación por valor de 500 millones de euros, un 87 % más que en 2014, cuando apenas se invirtieron 286 millones de euros» [Fraga, 2016].

¹http://www.imdb.com/title/tt2575988/?ref_=nv_sr_1

Actualmente en nuestro país hay 1783 empresas emergentes distribuidas principalmente en Madrid, Cataluña y la Comunidad valenciana. Dichas empresas se dedican principalmente al ecommerce(22 %), social media(13 %) y las empresas(12 %). En cuanto a la financiación, 172 inversores operan en el ámbito «startup» a lo largo de la península [Startupxplore, 2017] y los fondos que han aportado crecen año a año:

En 2013, tres «startup» lograron rondas de financiación que superaran los 10 millones de euros [...] en 2014, esta cifra aumentó a cuatro [...] el pasado año la explosión no tuvo parangón, ya que hasta 13 startups lograron capitalizar más de 10 millones de euros para fomentar su desarrollo.

[Fraga, 2016].

1.3. Lean startup

«Lean STARTUP» es un modelo de gestión empresarial dinámico ampliamente utilizado en la creación de empresas emergentes. En contraposición a las metodologías tradicionales, «Lean STARTUP» se basa en ciclos de desarrollo cortos que permiten sacar el producto al mercado de forma temprana. De este modo se puede obtener retroalimentación de los clientes en las etapas iniciales de la empresa, lo que da lugar a que el producto cambia y se adapta a las necesidades de los clientes.

El primer paso para crear una «startup» según esta metodología es plasmar las hipótesis sobre el modelo de negocio en el Lean Canvas (ver fig. 1.1)



Lean Canvas is adapted from The Business Model Canvas (<http://www.businessmodelgeneration.com>) and is licensed under the Creative Commons Attribution-Share Alike 3.0 Un-ported License.

Figura 1.1.: Lienzo propuesto por Ash Maurya para el proceso LEAN STARTUP. Fuente: Blog de Javier Megias. <http://javiermegias.com/blog/2012/10/lean-canvas-lienzo-de-modelos-de-negocio-para-startups-emprendedores/>

Estas hipótesis no conforman el modelo de negocio definitivo, si no que irán evolucionando a lo largo de la vida de la empresa de acuerdo al feedback de los clientes. Esta evolución del producto en relación a los deseos del clientes se denomina **customer development** y es uno de los conceptos claves en Lean startup.

El ciclo de vida de una «startup» se basa en tres pasos fundamentales que se repiten cíclicamente:

- **Construir:** se diseña el producto en función de las hipótesis que se establecen en el **lean canvas**. En la primera iteración se crea una versión del producto que tenga las mínimas funcionalidades necesarias para aportar valor a los potenciales clientes. Esta versión del producto se denomina **producto mínimo viable**. El objetivo de esta etapa es "comenzar a recopilar datos y medir resultados. Este modelo de producto no busca ser el resultado final sino un producto suficiente para testar la reacción del potencial cliente"[antevenio, 2016].
- **Medir:** tras contrastar nuestras hipótesis de negocio con los clientes a través del **producto mínimo viable** obtenemos información sobre nuestro producto y sobre la propia empresa mediante **métricas clave**. Dichas métricas (tales como ¿Cuánto cuesta captar un cliente? o ¿Cuánto dinero gastamos mensualmente?) son valoraciones objetivas sobre el rendimiento del producto y de la empresa, y calcularlas de forma periódica es importante ya que permite trazar una evolución y detectar errores y mejoras en la estrategia empresarial.
- **Aprender:** es una etapa clave ya que si el conocimiento obtenido se aplica, se estará más cerca de crear un producto que los clientes quieran comprar. "este conocimiento adquirido se debe aplicar a un nuevo proceso que comienza de nuevo. Se vuelve a crear un producto, que será una mejora del mismo lo que hace arrancar de nuevo el círculo de crear, medir y aprender"[antevenio, 2016]. Al llegar a este punto las startups se deben plantear si realizar un pequeño ajuste al producto y volver a **iterar** o si bien, en caso de que los resultados del producto hayan sido un desastre, hacer cambios de base al modelo de negocio. Estos cambios que afectan a una o más hipótesis del **lean canvas** se denominan **pivotar** y consisten en "cambiar una hipótesis fundamental sobre el producto, la estrategia, y el motor de crecimiento"[e mooc,].

El proceso se puede realizar cuantas veces sea necesario hasta conseguir el producto que se considere más acorde al cliente. La metodología Lean «startup» no trata de evitar que fallemos en el primer intento de lanzar al mercado nuestro servicio, sino que trata de que ese fallo nos salga más 'barato' al haber empleado una cantidad considerablemente menor de tiempo y de recursos materiales y económicos. [Peláez, 2015] .

1.4. Educación sobre emprendimiento

Dado que la creación de iniciativas empresariales es un fenómeno cada vez más extendido, ha surgido la necesidad de formar a profesionales que sean capaces de poner en

marcha tales proyectos y dirigirlos de forma exitosa. Actualmente existe un gran abanico de opciones formativas, entre las cuales destacaremos los videojuegos. El videojuego, visto como elemento narrativo, supone una poderosa herramienta de comunicación y de experimentación de vivencias.

El tema alrededor del cual gira este Trabajo fin de grado es el videojuego como elemento de aprendizaje sobre emprendimiento. La creación de un videojuego formativo provee a los futuros alumnos no solo de una formación teórica, si no que permite vivir una experiencia inmersiva en una historia que transmita conocimientos importantes a lo largo de la misma. Además es notablemente más divertido aprender jugando a un videojuego que atendiendo a interminables sesiones teóricas. Incluso se puede considerar como formación práctica el jugar ya que el jugador deberá de poner en acción los conocimientos teóricos previamente aprendidos.

2. Marco Teórico

El tema del emprendimiento no solo ha proporcionado ideas para la creación de películas cinematográficas o series de televisión. Como apoyo a los cursos formativos sobre emprendimiento han surgido multitud de juegos para enseñar y aplicar de forma práctica conceptos sobre la creación de iniciativas empresariales.

Estos juegos son una parte importante de la formación ya que permiten aprender de forma más amena. Además el utilizar de forma práctica los conocimientos adquiridos ayuda a que se entiendan mejor y se recuerden durante más tiempo.

Para los juegos mencionados anteriormente existen varios formatos que serán explicados en detalle a continuación.

2.1. Actividades en grupo

Es el tipo más sencillo y tradicional. Solo necesita de los participantes y una actividad previamente elegida. Una de las ventajas que tiene este tipo de actividades es que ponen a las personas en contacto directo, de modo que tienen que dejar a un lado la vergüenza e interactuar como lo harían ante clientes, inversores o trabajadores. Además estos encuentros pueden servir para hacer contactos útiles en un futuro.

Este tipo de actividades potencian las habilidades sociales y la creatividad de los participantes ya que los únicos elementos del juego son las personas, y las mecánicas del juego son sus discursos, explicaciones, gestos, actuaciones, etc.

2.2. Juegos de mesa

Los juegos de mesa mantienen muchos de los aspectos positivos de las actividades en grupo ya que también son presenciales, con las ventajas y desventajas que ello conlleva.

Además estos juegos pueden ser más divertidos debido a que introducen elementos como tableros, cartas, textos o ilustraciones entre otros elementos. Son especialmente interesantes para personas que no se sientan cómodas con las actividades en grupo debido al alto grado de interacción que demandan.

También es más fácil el uso de mecánicas complejas ya que hay instrumentos para contabilizar y describir el estado del juego: dados para contar el número de vidas, poder mágico o turnos; tableros con diferentes casillas, territorios o zonas; fichas de jugador con parámetros, habilidades, y características.

Actualmente hay numerosos juegos de mesa entre los que destacaremos:

Colonos de Catán: ¹ (figura 2.1) Apto para 6 jugadores (a través de una expansión),

¹<http://devir.es/producto/catan/>



Figura 2.1.: Tablero del juego Colonos de Catán. Fuente: Blog del club ERTAI. <https://clubertai.wordpress.com/2013/03/10/ludoteca-del-club-los-colonos-de-catan/>

este juego de gestión de recursos y comercio nos pone en la piel de un colono que debe ir construyendo sus aldeas y caminos. En Colonos de Catán prima tu habilidad para negociar por el contrario y tu capacidad de estrategia a medio y largo plazo. [Entrepreneurship, 2016]

El Catán [...] evita el enfrentamiento tan directo y obliga a negociar para ganar.

Ahí reside una de las claves, en el hecho que de entrada nadie posea recursos suficientes de todos los tipos para progresar. En cada turno se comercia con las materias primas, un trueque básico que permite saber cómo funciona un mercado libre en el que cada uno tiene sus propios intereses. [Albertini, 2015]

Lleva unas 18 millones de copias vendidas [...] Ha aparecido en «The Big Bang Theory» [...] Mark Zuckerberg se ha declarado adicto, «es uno de esos juegos a los que debes jugar si no quieres ser el “margi” de Silicon Valley» [Albertini, 2015]

Pandemia: ² (figura 2.2) En Pandemia somos un grupo de hasta 4 científicos que tienen

²<http://zacatrus.es/pandemia.html>



Figura 2.2.: Tablero del juego Pandemia. Fuente: Blog Estantería de juegos. <http://estanteriadegames.blogspot.com.es/2013/12/resena-pandemia.html>

que mantener a raya una serie de virus, o de lo contrario la Humanidad tendrá un grave problema. [...] debes aprender a formar equipo y hacerlo funcionar si quieres tener éxito en tu «startup». En este sentido, Pandemia puede ser la terapia perfecta para ti y tu equipo ya que o colaboráis y funcionáis perfectamente engrasados o correréis el riesgo de fracasar. Y también en la vida real. [Entrepreneurship, 2016]

Flea market: ³ (figura 2.3) Juego de dados en el que tendrás que descubrir los tesoros escondidos en un mercado de segunda mano. Tú serás el cliente que compra, y tu objetivo es adquirir bienes lo más barato posible para venderlos por más dinero una vez el dado de la demanda dice que ya son populares de nuevo. [González, 2015]

2.3. Videojuegos

Con el creciente éxito del mercado de los videojuegos y el entretenimiento digital, los videojuegos sobre el emprendimiento no se han hecho esperar. Los hay de diferentes tipos y temáticas aunque todos ellos comparten la esencia de dirigir un negocio.

³<https://boardgamegeek.com/boardgame/172410/flea-market>



Figura 2.3.: Tablero del juego Flea market. Fuente: Tienda online Shopyourway. <http://www.shopyourway.com/articles/355984>

Gestionar un negocio: en este tipo de videojuegos el jugador es el gerente de un negocio como por ejemplo una cafetería o una peluquería. El objetivo es gestionar la actividad diaria del establecimiento y ampliar el mismo utilizando los ingresos obtenidos. Ejemplos de este tipo de videojuegos son «Diner dash⁴» o «My Cafe: Recipes & Stories⁵». Este tipo de juegos tienen un carácter más infantil y recreativo y distan de ser un juego basado realmente en el emprendimiento y el mundo «startup». El motivo es que el peso del juego reside principalmente en la gestión cotidiana del negocio como servir pedidos, cobrar a los clientes o comprar materiales; valores muy importantes para el emprendedor como la creatividad, la estrategia, la pasión o la innovación no suelen tener cabida en estos juegos.

Hipster CEO: ⁶ (figura 2.4) este juego se puede considerar la contraposición al juego mencionado en el punto anterior ya que elimina las tareas de interacción directa con el cliente y en su lugar el jugador se centra en la gestión desde un punto de vista más técnico. El objetivo de este juego es lanzar y dirigir una «startup» con todo lo que ello conlleva.

Entre las tareas del jugador se pueden destacar: contratar personal, dedicar recursos a los diferentes departamentos (ventas, producción o marketing)

Para ello se dispone de un dashboard con varias **métricas clave** sobre la «startup» que ayudan al jugador a tomar decisiones acerca del rumbo de la misma. El juego no dispone de ningún tipo de historia o personajes que controlar, si no que la interacción con el mismo es a través de informes, dashboards, emails, etc.

U-startup: ⁷ (figura 2.5) es un juego desarrollado por la Universidad de Cádiz en España, en colaboración con la Cátedra de Emprendedores y OmniumLab. Lo novedoso de esto es que es el primer videojuego sobre emprendimiento con el que aprenderás a construir tu negocio con la metodología de CANVAS. [del Bosque, 2016]

Este juego se puede considerar la contraposición a los juegos mencionados en el punto anterior ya que explícitamente se muestran elementos típicos de la metodología **lean startup** tales como el **lean canvas**. El juego se basa en completar este canvas con las hipótesis de negocio del jugador a la vez que se completa una aventura en la que un fantasma es el mentor del jugador en el mundo empresarial, la pitonisa ayuda a encontrar los clientes objetivo y otros personajes y escenarios intervienen.

2.4. Motores de videojuegos

La creación de un videojuego sin la utilización de herramientas que agilicen el proceso puede ser una tarea más que ardua. Son muchos los motores de videojuegos que hacen

⁴https://es.wikipedia.org/wiki/Diner_Dash

⁵<https://play.google.com/store/apps/details?id=com.melesta.coffeeshop&hl=es>

⁶<http://www.hipsterceo.com/>

⁷<http://ustartup.es/>

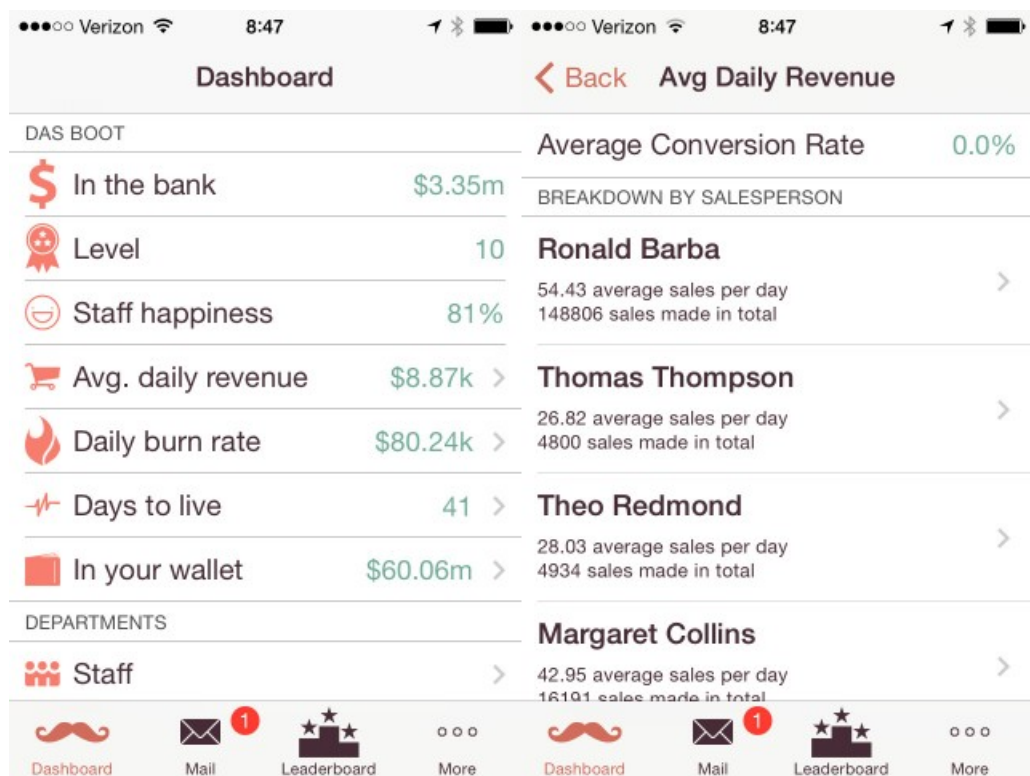


Figura 2.4.: Captura de pantalla de iPhone con el juego Hipster CEO



Figura 2.5.: Vista del lean canvas en el juego U-startup

este proceso más sencillo proporcionando utilidades como sistema de renderizado, motor físico o gestor de sonido. Además estos motores proporcionan código encargado de la gestión de la escena, de la memoria, de las entidades, etc.

3. Objetivos

3.1. Objetivo principal

El objetivo principal de este proyecto la creación un videojuego mediante el cual se puedan aprender conceptos claves acerca del emprendimiento y en concreto sobre la metodología «Lean startup».

3.2. Objetivos específicos

- Emplear el concepto storytelling para diseñar y escribir una historia que se desarrolle a lo largo de conversaciones entre el jugador y los demás personajes.
- Implementar un sistema conversacional en el que el jugador pueda seleccionar las respuestas que desea dar y en función de ello cambie la historia.
- Crear una interfaz de usuario que cumpla con unos criterios de calidad y que permita al jugador interactuar con la aplicación.
- Crear modelos 3D de personajes y otros elementos tales como edificios. Texturizar dichos modelos.

3.3. Objetivos secundarios

- Crear un videojuego usando el motor de videojuegos «Unity3D».
- Planificar el desarrollo de un proyecto de desarrollo de software y llevarlo a cabo.
- Aprender nuevas habilidades sobre el motor de videojuegos «Unity3D» y el lenguaje de programación C#.
- Crear un GDD (documento de diseño de videojuego) donde se documente con precisión como será el juego.
- Aprender nuevas habilidades sobre el software de modelado 3D «Blender».

4. Metodología

Para la realización del proyecto, la creación del videojuego, se dispondrán de 6 meses que se dividirán de la siguiente forma: inicialmente un mes y medio para la investigación, especificación del producto y planificación; cuatro meses desarrollando el producto; finalmente medio mes para dar los últimos retoques al juego y lanzar una versión beta.

Para el desarrollo del videojuego se utilizará una metodología ágil, ya que al disponer de poco tiempo para el desarrollo se necesitará tener un producto cuanto antes para poder enfrentarlo al público y obtener el feedback de este. Utilizando una metodología ágil se focaliza el esfuerzo en el desarrollo en lugar de en una excesiva planificación. De esta forma

se trabaja realizando entregas parciales pero funcionales del producto. De ese modo, es posible entregar en el menor intervalo de tiempo posible una versión funcional del producto.

[Martínez, 2014].

Además la utilización de estas metodologías ayuda enormemente a reducir el riesgo: al crear un videojuego, en este caso educativo, a pesar de las mejores intenciones de los desarrolladores no se puede saber con certeza si será del agrado de los jugadores.

Es por ello que la mejor estrategia posible es crear un producto mínimo viable (MVP) y que posteriormente se desarrolle y corrija según los deseos de aquellos que lo jugarán.

Existen actualmente una gran cantidad de metodologías ágiles. Entre las más populares se pueden destacar:

- Scrum: es una metodología orientada a equipos. Proporciona herramientas para el seguimiento diario del proyecto, la planificación de trabajo de forma iterativa y la comunicación y cooperación de los integrantes del grupo.
- Extreme Programming: orientada a equipos con pocos programadores.

se aplica en equipos con muy pocos programadores quienes llevan muy pocos procesos en paralelo. Consiste entonces en diseñar, implementar y programar lo más rápido posible, hasta en casos se recomienda saltar la documentación y los procedimientos tradicionales.

[Pastrana, 2015].

- Kanban: esta metodología propone dividir el trabajo en diferentes etapas bien diferenciadas. El objetivo es evitar los cuellos de botella limitando el trabajo en curso. Para ello, se establece un límite de trabajo en curso, lo que obliga a que cuando una tarea se empieza se debe terminar antes de iniciar una nueva.

La metodología ágil a utilizar será Kanban ya que es tremendamente sencilla de implementar: con unas simples tarjetas se pueden especificar las tareas a realizar, y con un tablero se pueden crear columnas que representan los estados de las diferentes tareas.

Dada la facilidad con la que se puede implementar Kanban, y que no es un sistema directamente orientado a equipos como SCRUM, será muy adecuado para el proyecto.

En cuanto al producto a desarrollar, los cuatro meses de desarrollo se dividirán en iteraciones de duración variable. Al finalizar la segunda iteración se espera tener un mínimo producto viable (MVP) y en las dos siguientes se perfeccionará dicho producto.

En cuanto al tiempo de desarrollo se dividirá en cuatro iteraciones:

- Primera iteración: desarrollo de la parte software del MVP usando placeholders en lugar de los modelos 3D y los demás elementos gráficos
- Segunda iteración: inclusión de los modelos 3D e imágenes definitivas
- Tercera iteración: recolección de feedback y corrección de errores. Optimización del juego
- Cuarta iteración: recolección de feedback y corrección de errores. Detalles finales del juego

Para ilustrar esta distribución de tareas se ha creado un diagrama de Gantt simplificado como se puede apreciar en la siguiente imagen 4.1.

Se utilizará la herramienta online TargetProcess ¹ para disponer de un tablero virtual «Kanban» en el cual poner las tarjetas y separarlas por procesos. En este aspecto goza de más popularidad la herramienta Trello ², aunque TargetProcess es más completa y ofrece muchas opciones como filtros, métricas y otras utilidades para monitorizar el trabajo.

¹www.targetprocess.com

²<https://trello.com/>

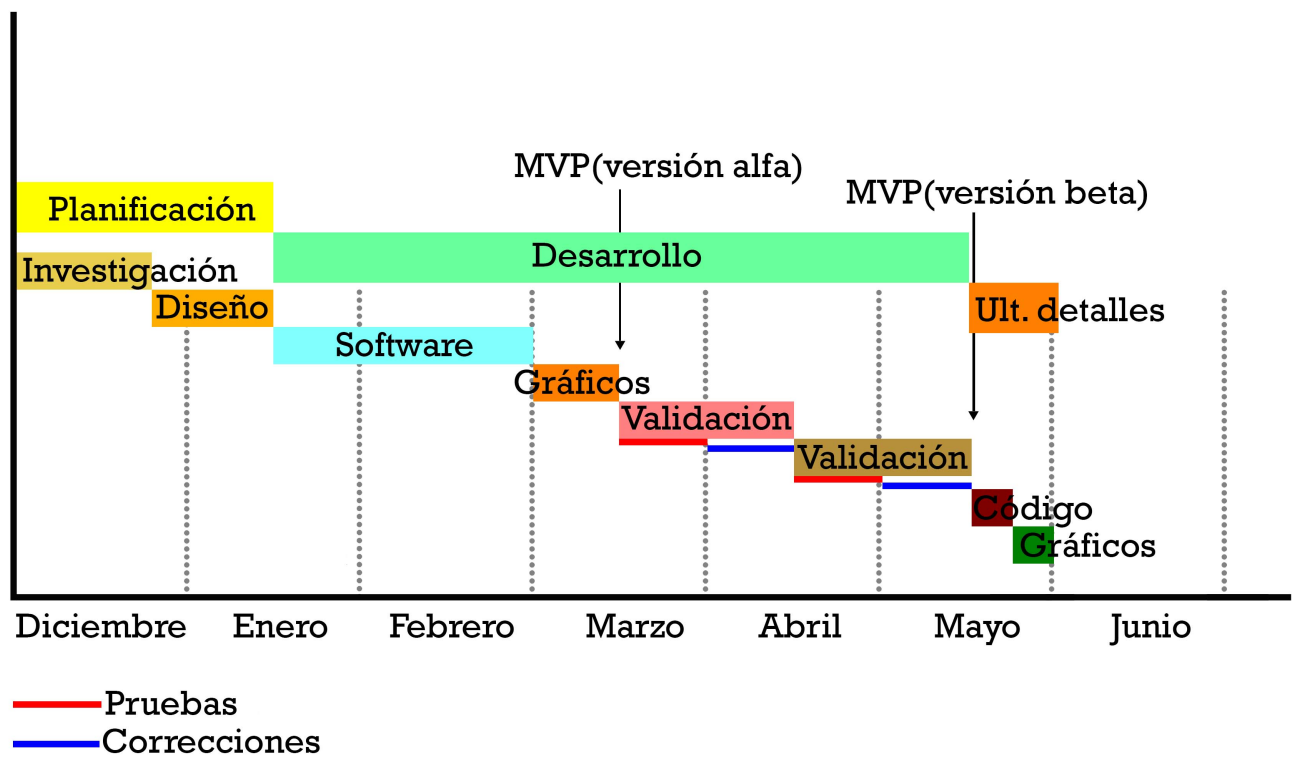


Figura 4.1.: Diagrama de Gantt con la distribución temporal y las dependencias de las tareas

5. Desarrollo

5.1. Descripción general

Para describir el desarrollo se ha decidido seguir el estándar IEEE 830 para la especificación de requisitos ¹.

5.2. Perspectiva del producto

El sistema a construir consistirá en una aplicación móvil para dispositivos Android ² desarrollada con el motor de videojuegos Unity3D ³.

La aplicación no formará parte de un sistema mayor, será un videojuego totalmente independiente. Aun así se hará uso de servicios de terceros tales como librerías de software, frameworks y APIs entre otros.

5.3. Funcionalidad del producto

Resumen de las funcionalidades principales que el producto debe realizar, sin entrar en información de detalle. El videojuego deberá permitir el movimiento del jugador en el plano 2D, además podrá interactuar con los NPCs y dialogar con ellos. Se podrán completar objetivos y logros para de este modo avanzar en la historia.

El juego además enseñará conceptos clave sobre «Lean Startup» y sobre el emprendimiento en general.

5.4. Características de los usuarios

El perfil de un consumidor de formación sobre emprendimiento es muy amplio: desde jóvenes recién graduados llenos de optimismo hasta personas de mediana edad que desean reinventarse y dejar de ser asalariados.

Es por ello que es difícil concretar un perfil ya que son diferentes personas de diferentes edades y perfiles socioculturales las que desean aventurarse en el emprendimiento.

En cualquier caso sí se pueden encontrar aspectos comunes en esta gran variedad de usuarios:

- conocimiento tecnológico y como desenvolverse con aplicaciones móviles

¹www.fdi.ucm.es/profesor/Gmendez/docs/is0809/ieee830.pdf

²www.android.com

³<https://unity3d.com/es/>

- interés por el mundo del emprendimiento y la empresa
- interés por los juegos educativos

5.5. Restricciones

Existen varias limitaciones a tener en cuenta a la hora de diseñar y desarrollar el sistema:

- para la realización del proyecto se dispondrá de un presupuesto nulo. Es por ello que las herramientas, frameworks y demás productos que se utilicen deberán ser gratuitas.
- el motor de videojuegos a utilizar deberá ser Unity 3D ya que se tiene conocimiento del mismo y no se dispone de tiempo para aprender a utilizar otro motor de videojuegos.
- el lenguaje de programación a utilizar será C# ya que de entre los disponibles para Unity 3D es el más adecuado por potencia, documentación y dominio por parte del desarrollador.
- el sistema operativo objetivo será Android ya que junto con los sistemas operativos para ordenador no requiere ninguna licencia para publicar aplicaciones. La plataforma serán los dispositivos móviles ya que es muy sencillo llegar al público de esta plataforma, aunque no tanto hacerse hueco entre dicha audiencia.

5.6. Requisitos específicos

5.6.1. Interfaces de usuario

Respecto a las interfaces de usuario se pueden observar dos estilos claramente diferenciados: las interfaces del menú principal y las de la pantalla de juego. En cuanto a las primeras deberán seguir un estilo minimalista y utilizar controles simples; respecto de las segundas, la simplicidad es obligatoria.

Es un requisito imprescindible que la interfaz mostrada durante el juego no sea intrusiva y entorpezca la experiencia de usuario. Esto se conseguirá disponiendo pequeños botones en la pantalla situados de forma estratégica para que la visión del jugador se centre principalmente en el mundo del juego y los personajes.

Un requisito común de las interfaces de usuario es que debido a que serán mostradas en un dispositivo móvil tendrán que adecuarse a una pantalla pequeña y recibir la interacción del usuario mediante toques en la pantalla del dispositivo.

Tabla 5.1.: My caption

Identificador	RF-USR-01
Nombre	Mover personaje
Requerimiento	El usuario podrá elegir donde mover al personaje controlado
Descripción	Al tocar con el dedo en cualquier punto de la pantalla el personaje controlado se moverá a esa posición (solo en el eje x)
Prioridad	Imprescindible

Tabla 5.2.: My caption

Identificador	RF-USR-02
Nombre	Interactuar con NPCs
Requerimiento	El usuario podrá seleccionar un NPC con el que interactuar
Descripción	Al tocar con el dedo sobre un NPC, si se está lo suficientemente cerca se abrirá el menú conversacional
Prioridad	Imprescindible

5.6.2. Requisitos funcionales

Para describir las funcionalidades del sistema se utilizará una aproximación basada en mecánicas. Cada posible acción del usuario sobre el sistema se considerará una mecánica y se definirá como un requisito.

Dichas mecánicas son activadas al detectarse cierto estímulo. Por ejemplo al producir el estímulo de tocar en algún lugar del mundo, se desencadena la mecánica de movimiento.

Al identificador de cada funcionalidad le acompañarán unas siglas (USR, UI, SYS) dependiendo de si dicha funcionalidad se refiere a mecánicas del usuario, la interfaz de usuario o al sistema.

Tabla 5.3.: My caption

Identificador	RF-USR-03
Nombre	Cambiar de escenario
Requerimiento	Se podrá navegar entre escenarios
Descripción	Al clicar en una de las puertas de cada escenario se avanzará al escenario asociado a dicha puerta
Prioridad	Imprescindible

Tabla 5.4.: My caption

Identificador	RF-UI-01
Nombre	Navegar menú principal
Requerimiento	Se podrá cambiar entre las diferentes pantallas del menú principal
Descripción	Arrastrando con el dedo en la pantalla hacia la derecha/izquierda se cambiará a la correspondiente pantalla
Prioridad	Imprescindible

Tabla 5.5.: My caption

Identificador	RF-UI-02
Nombre	Comenzar juego
Requerimiento	Se empezará el juego al seleccionar el botón correspondiente en el menú principal
Descripción	En el el menú principal en la vista inicial se podrá comenzar el juego al presionar el botón "play"
Prioridad	Imprescindible

Tabla 5.6.: My caption

Identificador	RF-UI-03
Nombre	Desactivar/Activar música
Requerimiento	Se podrá desactivar la música ambiente del juego
Descripción	Clicando en el icono de Música en el menú de opciones se activará/desactivará la música
Prioridad	Baja

Tabla 5.7.: My caption

Identificador	RF-UI-04
Nombre	Desactivar/Activar sonidos
Requerimiento	Se podrán desactivar los efectos de sonido del juego
Descripción	Clicando en el icono de Sonidos en el menú de opciones se activarán/desactivarán los efectos de sonido
Prioridad	Baja

Tabla 5.8.: My caption

Identificador	RF-UI-05
Nombre	Eliminar logros
Requerimiento	Se podrán eliminar los logros conseguidos
Descripción	Clicando en el icono de Eliminar logros del menú de opciones se eliminarán todos los logros obtenidos
Prioridad	Baja

Tabla 5.9.: My caption

Identificador	RF-UI-06
Nombre	Ver otros juegos
Requerimiento	Se podrá acceder a la descarga de otros juegos del autor
Descripción	Clicando en los iconos de juegos del apartado "Mas juegos" se accederá a Google Play donde se podrá descargar dicho juego
Prioridad	Baja

Tabla 5.10.: My caption

Identificador	RF-UI-07
Nombre	Navegar entre logros
Requerimiento	Se podrán seleccionar logros y ver su descripción
Descripción	En el menú de logros se podrá navegar entre los logros usando las flechas de navegación y ver la descripción y la imagen de los mismos
Prioridad	Media

Tabla 5.11.: My caption

Identificador	RF-UI-08
Nombre	Ver logros desbloqueados
Requerimiento	Se podrá ver la cantidad de logros desbloqueados hasta el momento
Descripción	En el menú de logros aparecerá un texto representando el número de logros desbloqueados hasta el momento y el total a conseguir
Prioridad	Baja

Tabla 5.12.: My caption

Identificador	RF-UI-09
Nombre	Ver descripción de logro
Requerimiento	Se podrá ver una descripción de los logros
Descripción	En el menú de logros se podrá ver un texto descriptivo del logro seleccionado
Prioridad	Media

Tabla 5.13.: My caption

Identificador	RF-UI-10
Nombre	Navegar entre logros
Requerimiento	Se podrá seleccionar el logro del que se desea ver información
Descripción	En el menú de logros se podrá navegar entre logros tocando y arrastrando en la lista de logros. El logro que se sitúe en el centro de la pantalla será el logro activo
Prioridad	Media

Tabla 5.14.: My caption

Identificador	RF-UI-11
Nombre	Abrir selector de personaje
Requerimiento	Se podrá abrir un menú donde seleccionar el personaje a controlar
Descripción	Clicando en el icono del selector de personaje se abrirá una ventana con los iconos de cada personaje. El icono del personaje actualmente controlado se mostrará en gris y no será seleccionable
Prioridad	Imprescindible

Tabla 5.15.: My caption

Identificador	RF-UI-12
Nombre	Seleccionar personaje
Requerimiento	Al clicar en un icono de personaje se cambiará el personaje seleccionado
Descripción	Al clicar en uno de los iconos de personaje se cambia el personaje controlado. En el icono del selector de personaje aparecerá el icono del nuevo personaje seleccionado
Prioridad	Imprescindible

Tabla 5.16.: My caption

Identificador	RF-UI-13
Nombre	Cerrar selector de personaje
Requerimiento	Se podrá cerrar el menú donde seleccionar el personaje a controlar
Descripción	Clicando en el selector de personaje, si este está abierto se cerrará
Prioridad	Imprescindible

Tabla 5.17.: My caption

Identificador	RF-UI-14
Nombre	Notificar objetivo completado
Requerimiento	Cuando se cumpla el objetivo actual se mostrará una notificación
Descripción	Al completar un objetivo el icono del indicador de objetivos girará. El texto del nuevo objetivo se añadirá a la lista de objetivos. El objetivo completado se mostrará tachado
Prioridad	Media

Tabla 5.18.: My caption

Identificador	RF-UI-15
Nombre	Abrir indicador de objetivos
Requerimiento	Se podrá abrir la lista de objetivos
Descripción	Al clicar en el icono del indicador de objetivos se desplegará una lista con los objetivos completados tachados y el objetivo actual sin tachar
Prioridad	Media

Tabla 5.19.: My caption

Identificador	RF-UI-16
Nombre	Cerrar indicador de objetivos
Requerimiento	Se podrá cerrar la lista de objetivos
Descripción	Al clicar en el icono del indicador de objetivos se cerrará la lista si esta estaba abierta
Prioridad	Media

Tabla 5.20.: My caption

Identificador	RF-UI-17
Nombre	Abrir menú conversacional
Requerimiento	Se abrirá el menú conversacional al interactuar con un personaje
Descripción	Al abrir el menú conversacional se hará zoom sobre los personajes involucrados. Se mostrará un texto en la pantalla correspondiente a lo que dice el NPC y botones con las posibles respuestas
Prioridad	Imprescindible

Tabla 5.21.: My caption

Identificador	RF-UI-18
Nombre	Contestar NPC
Requerimiento	Se podrá contestar a los NPC usando el menú conversacional
Descripción	Clicando en uno de los botones se contestará al NPC y se actualizará la conversación
Prioridad	Imprescindible

Tabla 5.22.: My caption

Identificador	RF-UI-19
Nombre	Terminar conversación
Requerimiento	Al alcanzar cierto punto de la conversación se cerrará el menu conversacional
Descripción	Cuando la última decisión tomada tiene asociada tiene la marca correspondiente de fin de conversación el menú conversacional se cerrará. El punto en el que se encuentra la conversación se guarda para retomarla desde ese punto posteriormente
Prioridad	Imprescindible

Tabla 5.23.: My caption

Identificador	RF-SYS-01
Nombre	Iniciar menú principal
Requerimiento	Al iniciar el juego el menú principal comienza en la vista correspondiente
Descripción	Al iniciar el juego el menú principal mostrará la vista con el título principal y el botón para iniciar el juego
Prioridad	Imprescindible

Tabla 5.24.: My caption

Identificador	RF-SYS-02
Nombre	Seleccionar personaje
Requerimiento	Al clicar en un icono del selector de personaje se cambiará el personaje controlado
Descripción	Cuando se clique en un icono del selector de personaje se moverá la cámara hasta enfocar al nuevo personaje seleccionado. Cuando se clique en la pantalla se moverá el nuevo personaje controlado
Prioridad	Imprescindible

Tabla 5.25.: My caption

Identificador	RF-SYS-03
Nombre	Cargar conversación
Requerimiento	Se cargarán las conversaciones desde archivos de texto
Descripción	Al abrir el menú conversacional se cargará el fichero de texto correspondiente a la conversación del personaje. Si ya se ha conversado con ese personaje se cargará la conversación desde el punto guardado
Prioridad	Imprescindible

Tabla 5.26.: My caption

Identificador	RF-SYS-04
Nombre	Contestar NPC
Requerimiento	Al contestar a un NPC se actualizarán las respuestas y el texto del NPC
Descripción	Al contestar a un NPC se leerá del JSON el nuevo texto y las nuevas respuestas y se actualizará la interfaz para mostrar estos datos
Prioridad	Imprescindible

Tabla 5.27.: My caption

Identificador	RF-SYS-05
Nombre	Codificar conversaciones
Requerimiento	Las conversaciones estarán codificadas en formato JSON y guardadas en ficheros de texto
Descripción	Cada NPC tendrá un fichero de texto asociado con la conversación que ofrece en formato JSON. Los elementos del JSON serán nodos con texto y nodos hijos con las respuestas asociadas a ese texto. Las respuestas podrán tener "markups" asociadas que indican eventos
Prioridad	Imprescindible

Tabla 5.28.: My caption

Identificador	RF-SYS-06
Nombre	Completar objetivo
Requerimiento	Se completarán objetivos al tomar decisiones con la markup correspondiente
Descripción	Cuando se elige una opción de conversación con la "markup" de objetivo asociada, se completará dicho objetivo
Prioridad	Imprescindible

Tabla 5.29.: My caption

Identificador	RF-SYS-07
Nombre	Cambiar de escenario
Requerimiento	Cuando el jugador cambia de escenario se modifica la escena
Descripción	Cuando se produzca un cambio de escenario se desactivará de la escena el escenario abandonado y el jugador aparecerá en el punto de "spawn" asociado al nuevo escenario
Prioridad	Imprescindible

Tabla 5.30.: My caption

Identificador	RF-SYS-08
Nombre	Guardar juego
Requerimiento	El estado del juego se guardará al salir
Descripción	Cuando el usuario salga de la aplicación o al menú principal, el estado del juego se guardará en la memoria del dispositivo
Prioridad	Baja

Tabla 5.31.: My caption

Identificador	RF-SYS-09
Nombre	Cargar juego
Requerimiento	Al iniciar el juego se cargarán las partidas guardadas
Descripción	Cuando se inicie el juego se cargarán las partidas guardadas en caso de existan. De ser así, al iniciar la partida el mundo se encontrará en el estado de la partida guardada
Prioridad	Baja

Tabla 5.32.: My caption

Identificador	RF-SYS-10
Nombre	Actualizar conversaciones
Requerimiento	Al completar un objetivo se actualizan las conversaciones
Descripción	Al completar un objetivo en una conversación, todas las demás conversaciones se actualizarán a un punto correspondiente de la misma para reflejar el objetivo completado. Las nuevas conversaciones empezarán desde dicho punto.
Prioridad	Imprescindible

5.6.3. Requisitos de usuario

5.6.4. Requisitos de interfaz

5.6.5. Requisitos de sistema

5.7. Requisitos no funcionales

- Rendimiento: el sistema debe de proporcionar una experiencia de juego óptima. Para ello el flujo de juego debe de ser fluido, sin parones o interrupciones.
- Usabilidad: el sistema debe de ser sencillo de utilizar. De modo que personas sin un contacto o entrenamiento previo sepan desenvolverse por el sistema
- Costo: el coste del sistema deberá ser cero

5.8. Arquitectura del sistema

Para poder explicar la arquitectura del sistema construido es necesario en primer lugar exponer la arquitectura del motor de videojuegos utilizado. El motivo es que el sistema construido tendrá que cumplir con las normas y reglas del motor, ya que se usarán las herramientas que este provee. Es por ello que se puede afirmar que la arquitectura del videojuego está condicionada y subordinada a la del motor utilizado.

Dado que para la realización de este proyecto se ha utilizado el motor de videojuegos Unity3D ⁴, se explicará a continuación el diseño de dicho motor.

5.8.1. Arquitectura de Unity3D

Unity3D basa su diseño en el modelo entidad-componente ⁵. En este diseño, las entidades del mundo del juego obtienen su funcionalidad mediante la agregación de diferentes componentes.

En el paradigma de la programación orientada a objetos ⁶ el comportamiento de una entidad se define en la clase que la representa. Sin embargo, para reducir la duplicación de código se emplea el mecanismo de la herencia, por el cual una clase puede heredar de otra y de esta forma obtener su comportamiento y ampliarlo. El problema que reside en basar el diseño en jerarquías de herencia es que a medida que crece la complejidad del sistema crece el "árbol" que forman las clases. Llegados a este punto es probable que se de el caso en el que se precisa cambiar la funcionalidad de una de las clases, y que al hacerlo se modifique de forma involuntaria el comportamiento de otras clases que heredan de la modificada. Llevando el caso a un punto más extremo, se podría dar la situación en la que el comportamiento de una clase sea incompatible con el comportamiento heredado de una clase padre.

Para solucionar este problema el diseño entidad-componente propone crear entidades que actúen como meros contenedores. Dichos contenedores agregarán componentes que les darán la funcionalidad que los definirá. Por ejemplo tanto un jugador como los objetos del escenario tendrían el componente Mesh (malla gráfica) pero solo el jugador tendría el componente Movement (movimiento). Como se puede observar este diseño es altamente flexible, ya que si se toma la precaución de hacer los componentes lo suficientemente genéricos, se pueden aplicar a varias entidades, de forma que hay una gran reutilización de código.

Otra ventaja es que se produce un muy bajo acoplamiento ya que las funcionalidades están encapsuladas en los componentes y estos son independientes entre sí.

La forma en la que Unity3D aplica este diseño entidad-componente es mediante el uso de «Gameobjects» (entidades) y components. Los «Gameobjects» son la entidad fundamental en Unity3D: todos los elementos que existan en la escena son un «Gameobject». Además todas las entidades en Unity3D (en adelante Gameobjects) incorporan como mínimo un componente. Dicho componente se llama «Transform» y se encarga de la gestión de la posición, la rotación y la escala del «Gameobject» que lo posee.

En la siguiente imagen (ver fig. 5.1) se puede observar los componentes que posee el «Gameobject» Leonardo.

⁴<https://unity3d.com/es/>

⁵<https://www.genbetadev.com/programacion-de-videojuegos/disenio-de-videojuegos-orientado-a-entidades-y-componente>

⁶<https://desarrolloweb.com/articulos/499.php>

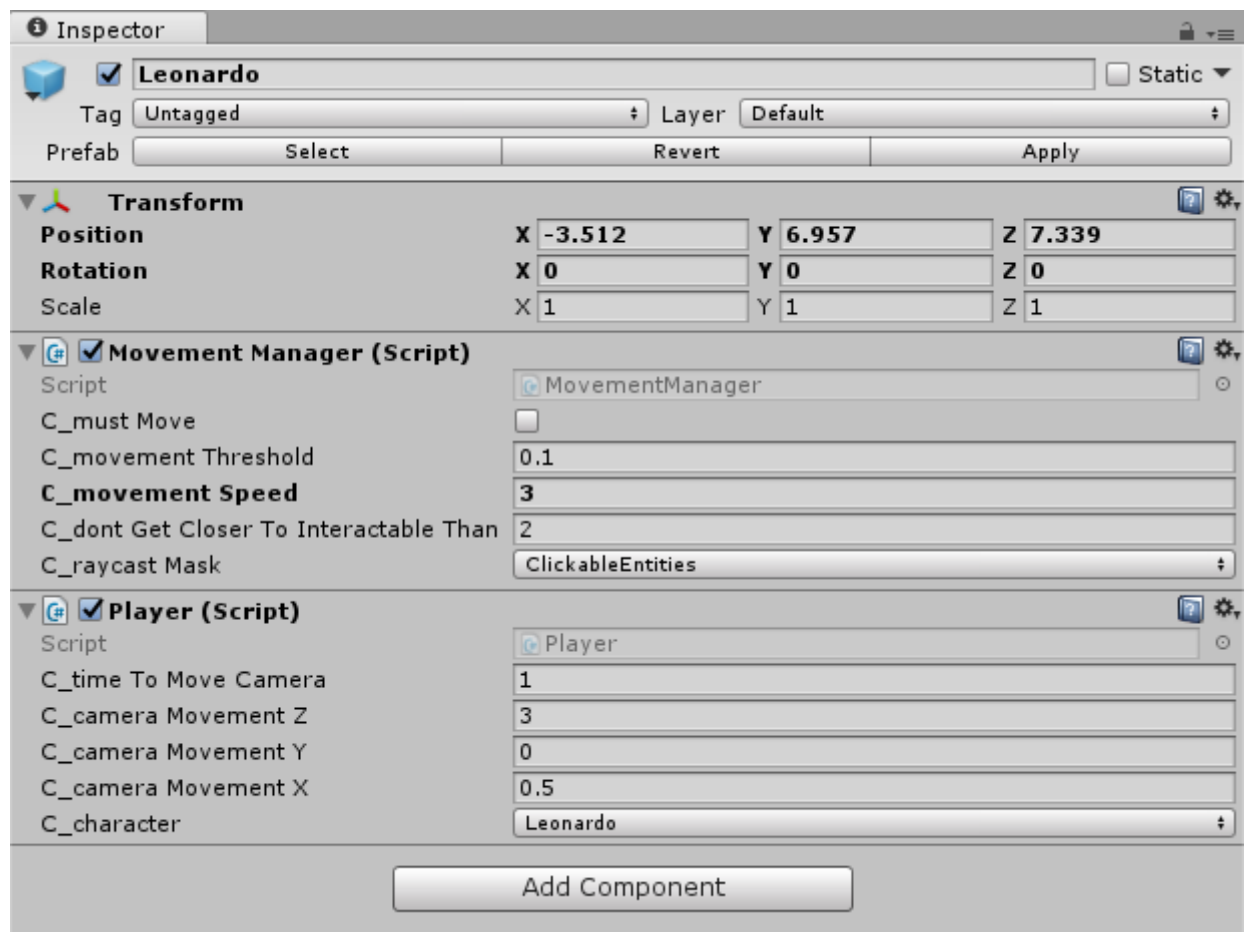


Figura 5.1.: Vista del editor de Unity3D donde se pueden ver los componentes de un GameObject

Como se ha mencionado anteriormente el componente «Transform» es obligatorio en todos «Gameobjects». También se pueden ver otros dos componentes, «Movement Manager» y «Player», que han sido creados por el desarrollador. Dichos componentes dotan a la entidad, que en este caso es el personaje controlable por el jugador, de las características que lo hacen único. Entre otras cosas le dan la habilidad de reaccionar ante los inputs del usuario y moverse por el escenario. También se pueden observar una serie de variables con parámetros. Dado que cada instancia de los componentes es única para la entidad que lo agrega, se pueden personalizar para que incluso entre entidades con los mismos componentes se comporten de manera diferente.

En la siguiente imagen (ver fig. 5.2) se puede observar un diagrama de clases que muestra la estructura que compone el sistema de componentes y los GameObjects.

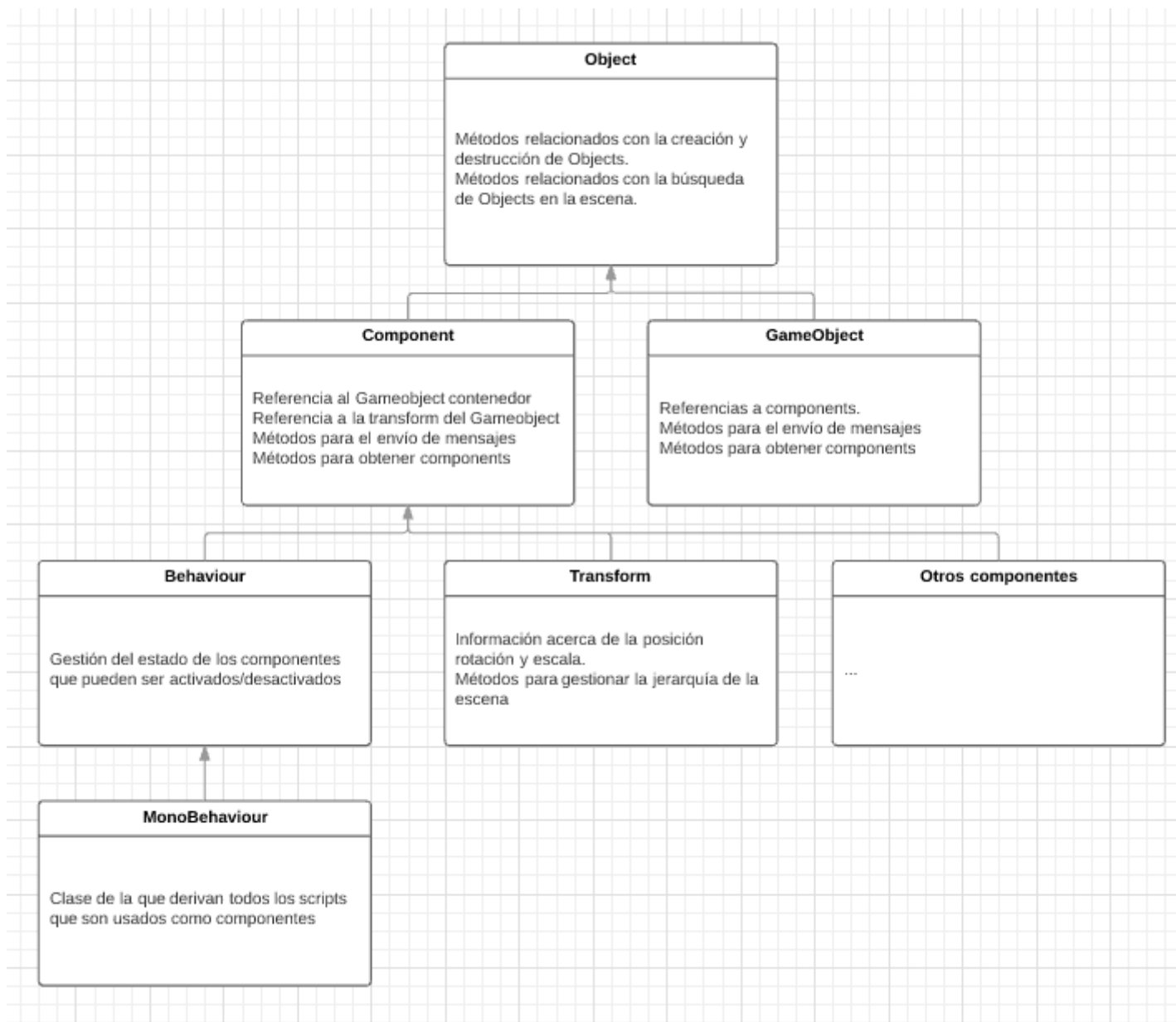


Figura 5.2.: Diagrama de clases de Unity3D

Los componentes que provee Unity3D derivan de la clase **Component**. En el caso de la imagen solo aparece el componente «Transform», pero hay muchos más como «Renderer», «Rigidbody», «Collider», etc. En la clase especializada cada componente contendrá la lógica necesaria para desempeñar su función, y en la clase **Component** se referenciará

al «GameObject» que lo contiene. Cabe notar que un componente no puede existir de forma independiente. Debe estar siempre asociado a una entidad.

La clase MonoBehaviour es también muy importante ya que es la clase base para los scripts creados por los desarrollares. Cualquier fragmento de código que se desee añadir como componente a una entidad deberá heredar de dicha clase.

Por supuesto se pueden escribir scripts que no hereden de la clase MonoBehaviour, pero en ese caso no podrán ser asignados como componentes a una entidad. Sin embargo no dejan de ser útiles ya que pueden ser usados como POCO ⁷, definir interfaces, contener lógica o cualquier otra cosa que el desarrollador desee.

En última instancia los scripts que no heredan de MonoBehaviour deben de ser utilizados desde un script que sí lo haga, ya que este es el único punto de entrada que tienen hacia el sistema de Unity3D.

El último aspecto de la arquitectura que se comentará es el bucle de juego: una parte fundamental de cualquier videojuego. El bucle de juego es la parte software más importante en cualquier videojuego. En él se ejecutan las tareas que hacen que el juego 'esté vivo'. En el siguiente fragmento de código (ver fig. 5.1) se presenta un ejemplo de bucle de juego básico:

Listado 5.1: Código de bucle de juego

```
while(true)
{
    ProcesarInput();
    ActualizarJuego();
    Renderizar();
}
```

En este fragmento de código se pueden observar las tres tareas fundamentales de las que se compone cualquier videojuego:

- Procesar input: consiste en capturar la interacción del usuario con el sistema mediante el hardware. En esta etapa se puede aplicar algún tipo de filtrado sobre dicho input.
- Actualizar juego: se actualiza el estado del juego. Esto consiste en actualizar IA, realizar las acciones del personaje, actualizar el mundo, efectos de objetos, etc.
- Renderizar: los elementos del juego se procesan para ser dibujados en la pantalla.

El ejemplo de bucle mostrado anteriormente es tremendamente básico pero sirve para explicar las bases de un bucle de juego. Probablemente ningún videojuego moderno lo utilice ya que tiene múltiples inconvenientes como que la frecuencia de actualización del juego dependerá de la carga de trabajo y del rendimiento del sistema que la procese. Actualmente existen variaciones del bucle que solucionan este problema, pero son temas que quedan fuera del alcance de este proyecto.

⁷https://es.wikipedia.org/wiki/Plain_Old_CLR_Object

En la siguiente imagen (ver fig. 5.3) se puede observar el bucle de juego utilizado por el motor Unity3D, y evidentemente por el juego creado para este proyecto:

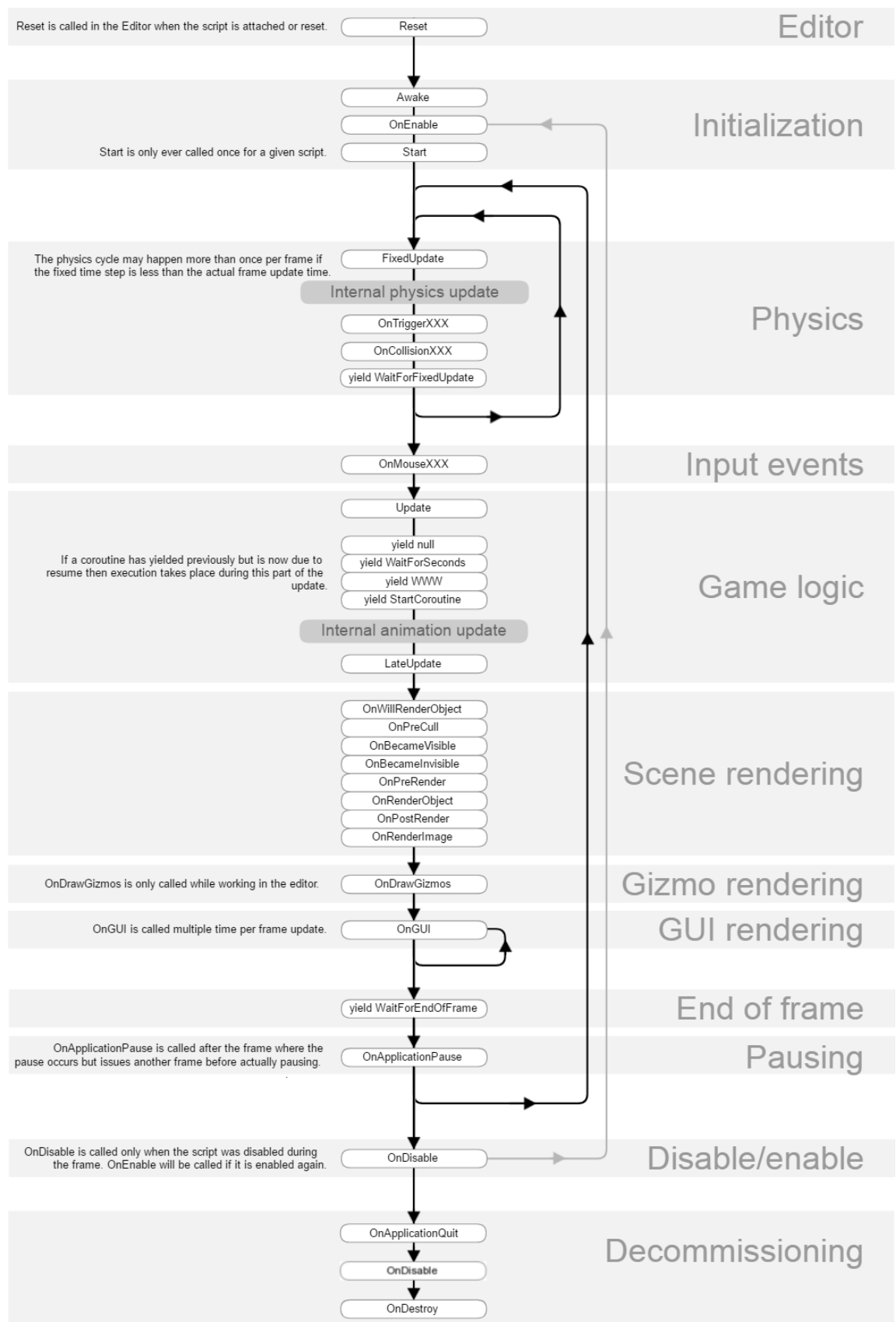


Figura 5.3.: Bucle de juego del motor Unity3D

En la imagen se pueden observar las 3 fases clave del bucle de juego: la captura de acciones del jugador, que se produce en la parte denominada como «Input Events»; la actualización del mundo de juego que se lleva a cabo en «Physics» y «Game logic»; finalmente el dibujado del mundo, que se produce en «Scene rendering», «Gizmo rendering» y «GUI rendering».

5.8.2. Arquitectura del juego

A continuación se describirá la arquitectura utilizada para desarrollar los componentes más importantes del juego. Estos son: el sistema de conversaciones, el sistema de movimiento, el sistema de escenarios y el sistema de cambio de personajes

Sistema de conversaciones

El sistema de conversaciones es el más importante del juego ya que los diálogos con los personajes del juego son el hilo conductor de este.

Los diálogos en lugar de estar insertados directamente en el código se almacenan en ficheros de texto plano independientes. Esto permite que se puedan modificar los diálogos sin tener que recompilar el código del juego. Además permite que cualquier persona sin conocimientos de programación escriba diálogos.

Cada diálogo es un archivo que sigue el formato JSON ⁸. Las conversaciones tienen una estructura arbórea donde los nodos representan la parte del diálogo correspondiente al NPC y las ramas salientes de dicho nodo representan las posibles contestaciones. La representación de una conversación en el formato JSON sigue la estructura que se puede ver en la siguiente imagen (ver fig. 5.4).

⁸<https://geekytheory.com/json-i-que-es-y-para-que-sirve-json>

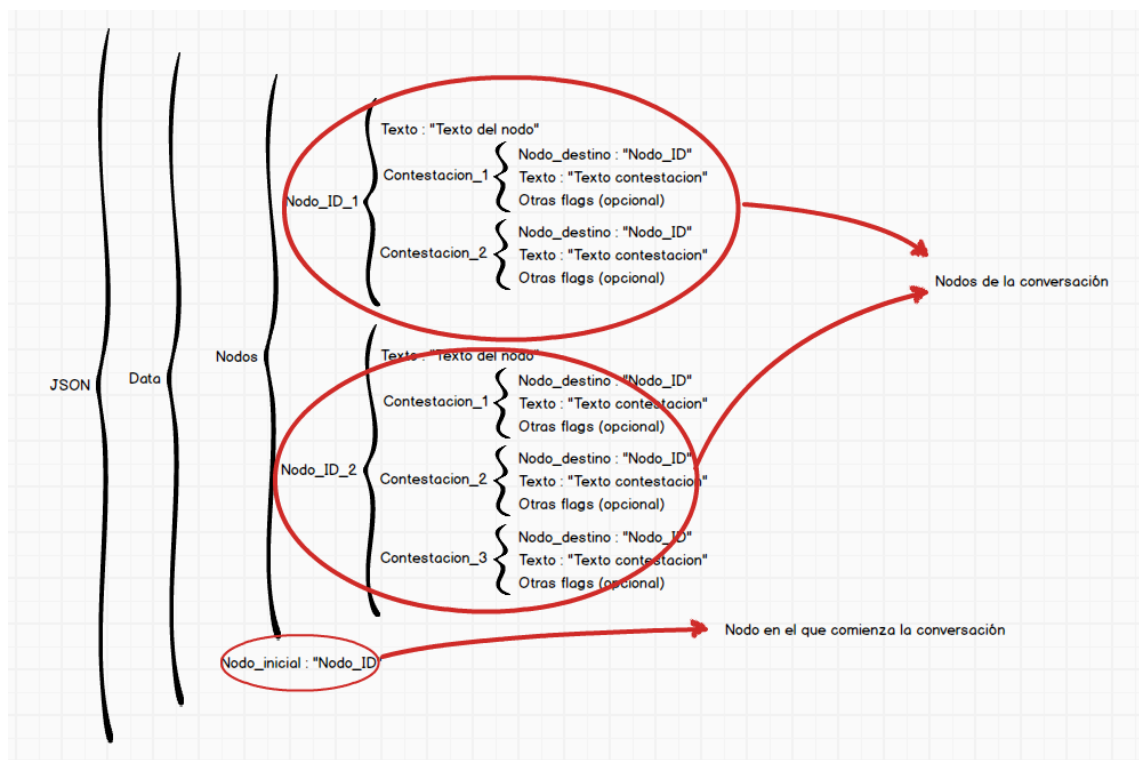


Figura 5.4.: Representación del formato utilizado para los ficheros de diálogo

Como se puede observar, los datos contenidos en el JSON son un array de objetos **Nodo** y un string que indica el nodo inicial de la conversación.

Los objetos **Nodos** están indentificados por una cadena de texto que se compone de las tres primeras palabras del diálogo. Cada **Nodo** contiene el texto del NPC y las posibles contestaciones que el jugador le puede dar.

Las contestaciones se componen de el identificador del nodo al que conduce dicha contestación y el texto de la contestación en sí. Además puede contener ,o no, diferentes «flags» que le dan a la elección una funcionalidad extra. Entre las «flags» disponibles se encuentran la de acabar la conversación, desbloquear un objetivo o desbloquear un logro.

Para la creación de los ficheros de diálogos se puede escribir el JSON manualmente rellenando con los datos deseados o se puede utilizar la herramienta *inklewriter*⁹. Dicha herramienta presenta una forma más sencilla de escribir conversaciones de múltiple respuesta (ver fig. 5.5). Además cuenta con la posibilidad de exportar el trabajo final a un fichero de texto en formato JSON. El parser JSON del videojuego está construido específicamente para poder procesar los diálogos contruidos con *inklewriter*.

⁹<http://www.inklestudios.com/inklewriter/>

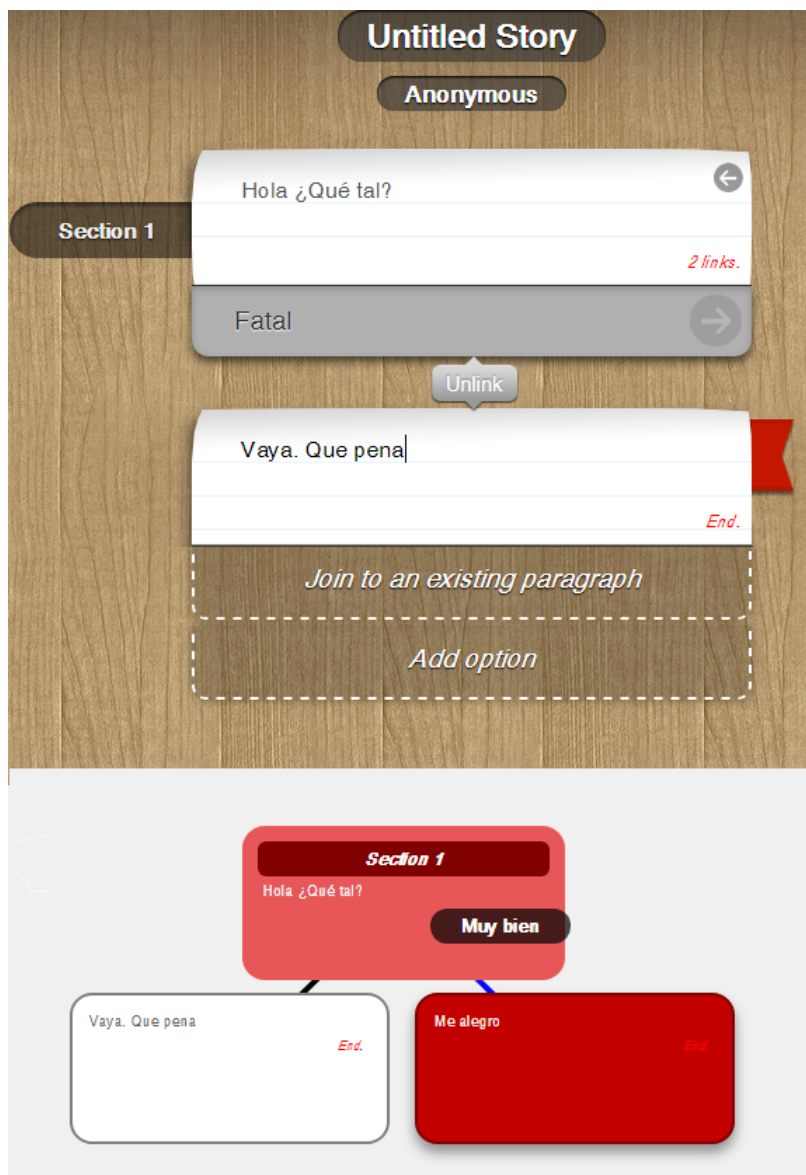


Figura 5.5.: Herramienta inklewriter

Para explicar la arquitectura del sistema de conversaciones se emplearán sendos diagramas de secuencia: uno para explicar el proceso de iniciar una conversación y otro para explicar el proceso de selección de contestaciones.

En el siguiente diagrama (ver fig. 5.6) se ve el flujo de código necesario para iniciar una conversación.

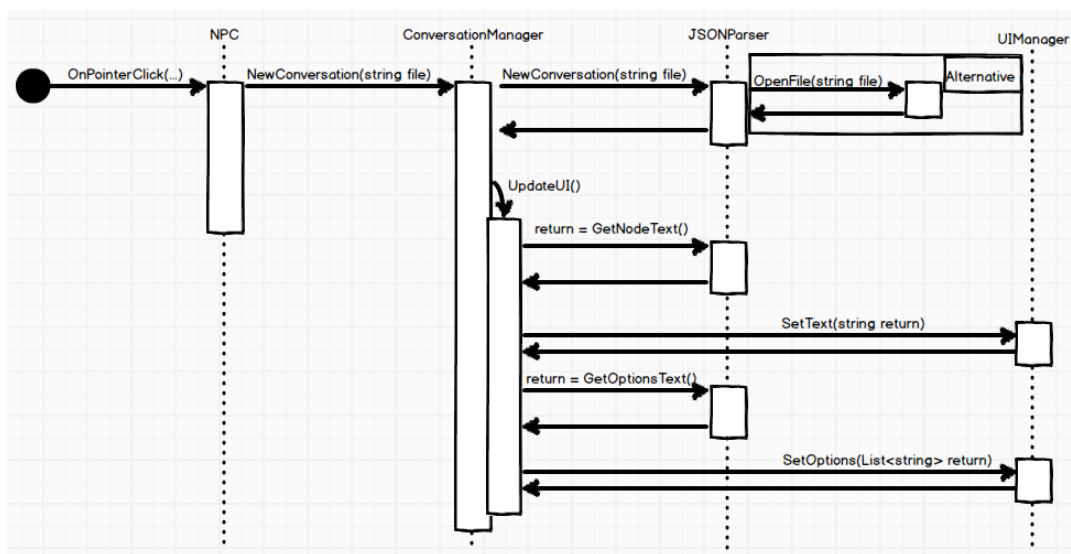


Figura 5.6.: Diagrama de flujo nueva conversación

No se ha indicado en el diagrama pero para el tratamiento JSON del fichero y para todas las operaciones JSON se utiliza la librería SimpleJSON ¹⁰.

El punto de entrada es el click del jugador en un NPC. Previamente el NPC se habrá registrado en el ConversationManager, de forma que el NPC guarda en un delegado el método NewConversation(string) del ConversationManager.

Cuando se le pide al ConversationManager que inicie una nueva conversación, el NPC le pasa por parámetro la ruta del fichero en la que está almacenado su diálogo con el personaje que está controlando el jugador en ese momento. El ConversationManager informará al JSONParser de que se debe iniciar una nueva conversación, y este será el encargado de leer y parsear el fichero o de abrir una de las conversaciones que tiene almacenadas.

Una vez informado el JSONParser, el ConversationManager le pedirá el texto del nodo actual de la conversación y el texto de las posibles respuestas. Una vez tenga esa información se la pasará al UIManager que se encarga de la interacción con los elementos gráficos de la interfaz.

En este arquitectura se ha seguido un diseño inspirado en el patrón Model-view-Controller ¹¹. Para ello todas las referencias a los elementos de la interfaz y los métodos de interacción con los mismos se han encapsulado en la clase llama UIManager. El acceso a los datos de las conversaciones, la gestión del estado de las mismas y la lectura de los ficheros de diálogos se guardan en la clase JSONParser, que además utiliza la librería SimpleJSON. En cuanto al punto de encuentro entre los datos y la interfaz, se emplea

¹⁰<http://wiki.unity3d.com/index.php/SimpleJSON>

¹¹<https://es.wikipedia.org/wiki/Modelo-vista-controlador>

la clase `ConversationManager` para sincronizar las operaciones entre las otras dos clases y para proveer un punto de entrada al sistema de conversaciones.

Una vez abierta la conversación es necesario actualizar el juego cada vez que el jugador selecciona una respuesta. El funcionamiento de esta mecánica es muy similar y utiliza los mismos componentes que el sistema anterior. En el siguiente diagrama de flujo (ver fig. 5.7) se explica el proceso.

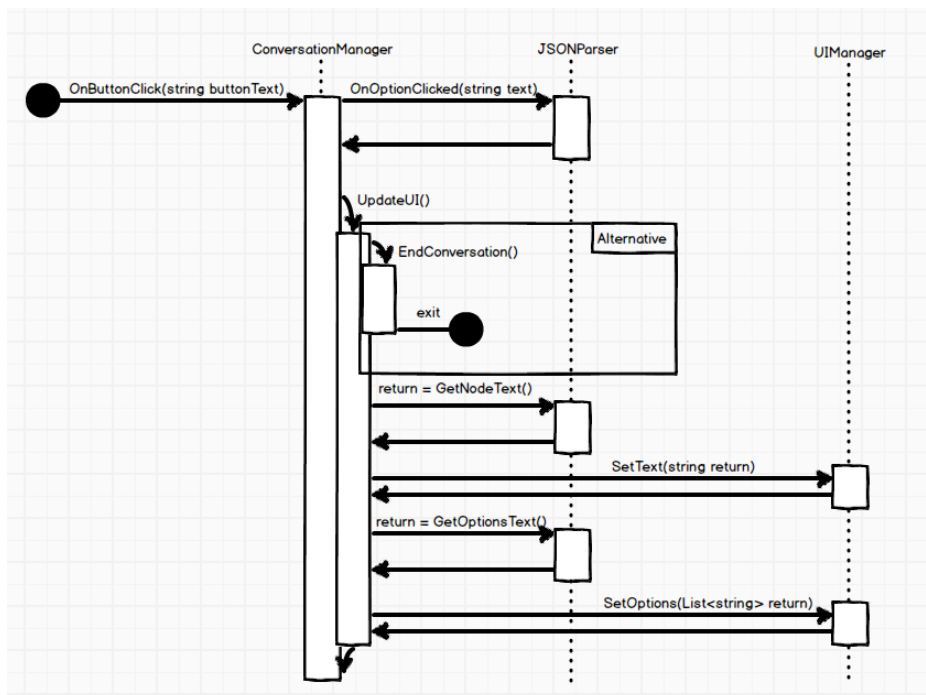


Figura 5.7.: Diagrama de flujo de selección de respuesta

Sistema de escenarios

A lo largo del juego se puede navegar por diferentes escenarios. Por temas de eficiencia y debido a que cada escenario tiene características únicas se han agrupado según la temática. En lugar de tener un gran escenario con montones de mallas correspondientes a los edificios y los personajes, solo están activos los `Gameobjects` correspondientes al escenario actual, los demás están desactivados (que no eliminados). En los escenarios hay puertas que al ser clicadas transportan al personaje activo hacia el escenario al que conduce la puerta.

En el inspector de escena de Unity3D dichos escenarios lucen como se en la siguiente imagen (ver fig. 5.8).

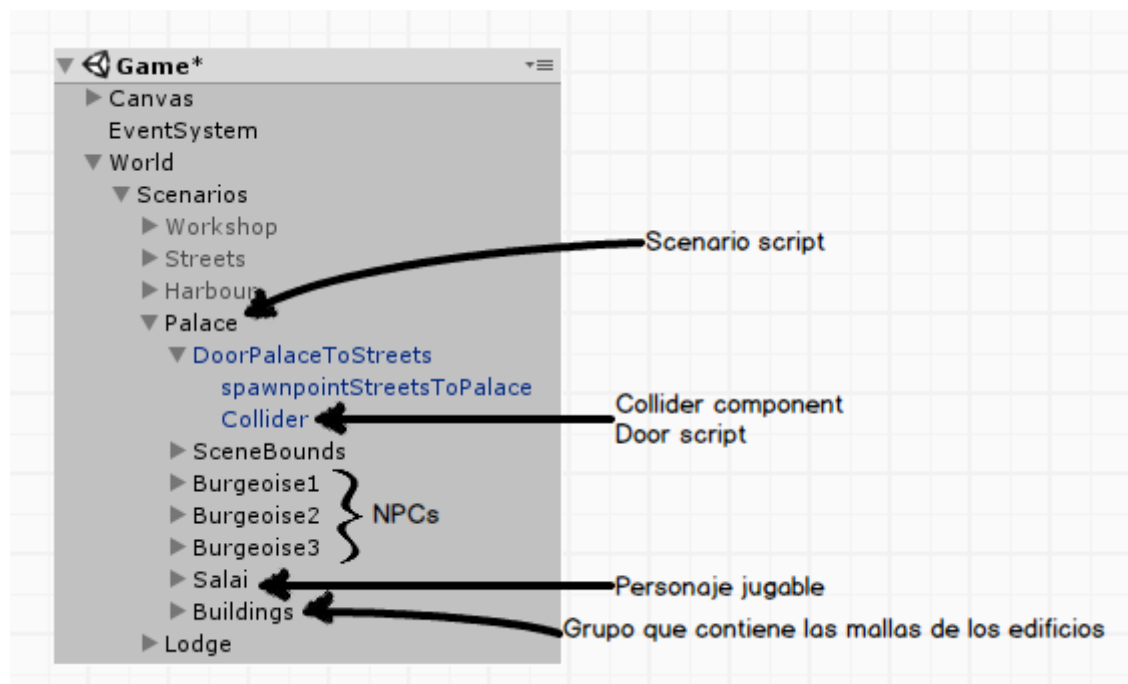


Figura 5.8.: Vista de los escenarios en el inspector de Unity3D

Como se puede observar algunos Gameobjects de la escena contienen scripts y componentes (otros Gameobjects tambien pero se han omitido para mayor simplicidad). En el siguiente diagrama de clases (ver fig. 5.9) se muestra la estructura de estos scripts y posteriormente se explicará su funcionamiento y utilidad.

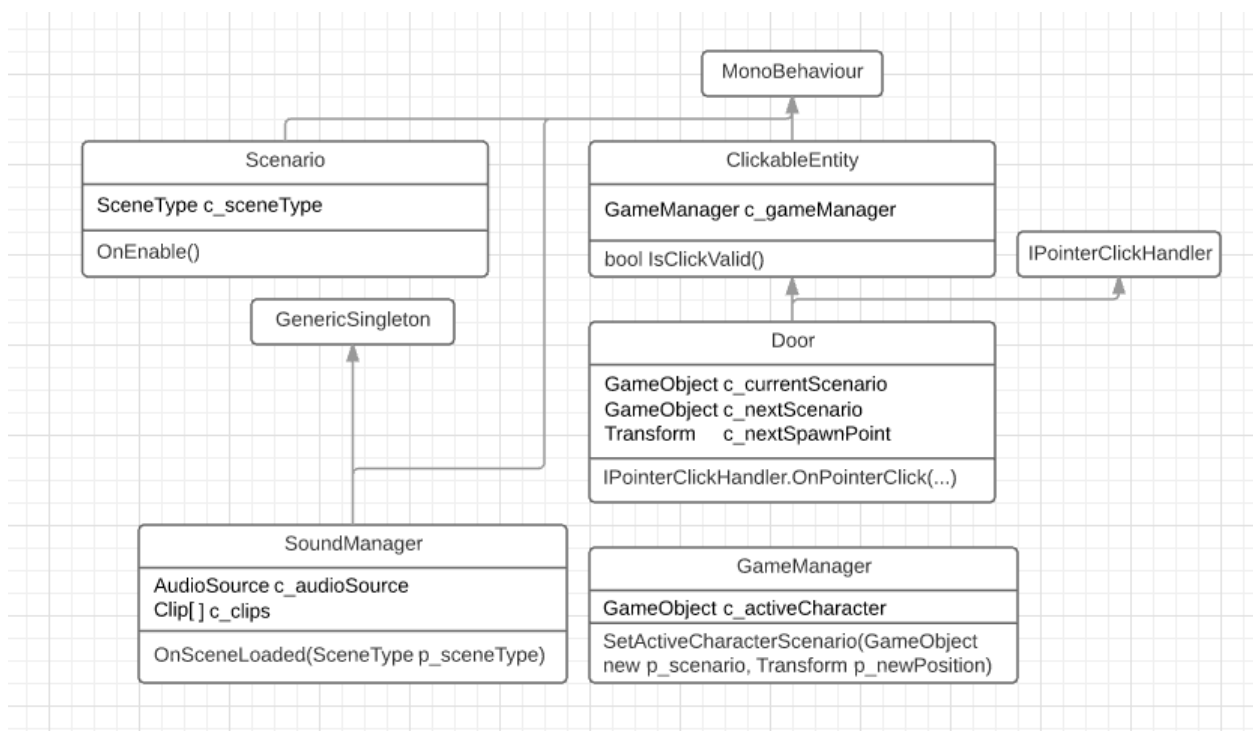


Figura 5.9.: Diagrama de clases del sistema de escenarios

El elemento central de todo el sistema de escenarios es la puerta. El nombre del Gameobject puerta sigue una notación que indica el escenario destino y origen. En el caso de la imagen anterior (ver fig. 5.8) se puede observar que la puerta lleva del Palacio (el escenario activo) a Las calles (un escenario desactivado).

El método **IPointerClickHandler.OnPointerClick(...)**, heredado de la interfaz **IPointerClickHandler** que provee Unity3D, es llamado cuando se hace click en el componente Collider asociado a la puerta. Posteriormente se comprobará que el click es válido (el jugador no está muy lejos por ejemplo) con el método heredado **IsClickValid()**. Finalmente se notificará al **GameManager** de que se quiere hacer un cambio de escena y se le suministrará el Gameobject que contiene la nueva escena y la posición de inicio para el jugador.

El sistema de escenarios se encarga también de gestionar la pista de audio que se reproduce a modo de música ambiental. En el objeto raíz de cada escenario se encuentra un script que hace uso del método provisto por Unity3D, **OnEnable()**. Dicho método es llamado cada vez que se activa un escenario. Desde el método **OnEnable()** se notifica al **SoundManager**, que es accesible de forma global ya que implementa el patrón Singleton, el nuevo escenario activado. El **SoundManager** dispone de un clip de sonido por cada escenario y cuando recibe el tipo del nuevo escenario cambia el clip de sonido reproducido.

por el componente AudioSource.

Sistema de cambio de personajes

Durante el juego se puede cambiar

6. Resultados

6.1. Section1

blabla

6.2. Section2

bla bla

7. Conclusiones

7.1. Mejoras y ampliaciones

Para describir el desarrollo se ha decidido seguir el estándar IEEE 830 para la especificación de requisitos ¹.

7.2. Modelo de negocio

El sistema a construir consistirá en una aplicación móvil para dispositivos Android ² desarrollada con el motor de videojuegos Unity3D ³.

¹www.fdi.ucm.es/profesor/Gmendez/docs/is0809/ieee830.pdf

²www.android.com

³<https://unity3d.com/es/>

Bibliografía

- [Albertini, 2015] Albertini (2015). Bienvenidos a catán, la isla que jubiló a un dentista.
- [antevenio, 2016] antevenio (2016). ¿en qué consiste la metodología lean startup?
- [Blank and Dorf, 2013] Blank, S. and Dorf, B. (2013). *El manual del emprendedor*. Grupo Planeta Spain.
- [del Bosque, 2016] del Bosque, D. (2016). 7 juegos virtuales que te enseñarán sobre negocios.
- [e mooc,] e mooc. Iterar o pivotar.
- [Entrepreneurship, 2016] Entrepreneurship, F. (2016). 7 juegos de mesa para emprendedores.
- [Fraga, 2016] Fraga, A. I. (2016). Así latén las startups en españa.
- [González, 2015] González, M. (2015). Si te gustó 'los colonos de catán', atento a estos dos nuevos juegos de mesa.
- [Martínez, 2014] Martínez, E. (2014). Las 8 grandes ventajas de las metodologías ágiles.
- [Navale, 2013] Navale, A. B. (2013). Developing entrepreneur skills for corporate work. *Research Directions*, 1:1–3.
- [Pastrana, 2015] Pastrana, O. (2015). 5 beneficios de aplicar metodologías ágiles en el desarrollo de software.
- [Peláez, 2015] Peláez, A. (2015). ¿en qué consiste la metodología lean startup?
- [Startupxplore, 2017] Startupxplore (2017). A view of the spanish startup community.

A. Anexo I

Aquí vendría en anexo I