

# El documento de diseño

Por Gorka Suárez García  
e-mail: [gorkasg@yahoo.es](mailto:gorkasg@yahoo.es)

Fecha: 30-10-2008  
Versión 1.2.2

Esta obra está bajo una licencia Creative Commons:  
<http://creativecommons.org/licenses/by-nc-sa/2.5/es/>

## **¿Ingeniería del software aplicada a los videojuegos?**

Desarrollar software de cierta calidad es tremendamente complicado y si hablamos de realizar un videojuego, podemos asegurar que se trata de algo terrible y dolorosamente complicado. Pero no nos desanimemos, tomemos simplemente ciertas cosas en consideración. Una de ellas es el aplicar la ingeniería del software, rama de la informática que muchos tratan lamentablemente de evitar a cualquier precio. Y digo lamentablemente, ya que algunos no alcanzan a sospechar que el precio a pagar puede ser considerablemente caro, como acabar cancelando el proyecto en el que se está metido, postergarlo infinitamente o acabar finalizando un churro de programa que solo dios sabe porque llega a "funcionar".

Sin embargo no todo lo que se encuentra en la I.S. es interesante o útil, porque tampoco es necesario caer en el autoengaño y creer en las cuentas milagrosas de las que algunos libros hablan, como por ejemplo el infame libro Pressman, pero sí que hay algunas cosas que bien pueden ser útiles para llevar a buen puerto un proyecto. Para este caso en concreto, el de hacer un videojuego, existe una bonita herramienta llamada "documento de diseño". Y a lo largo de este ensayo espero poder sentar unas bases mínimas para de su comprensión y de su desarrollo, dejando claro que esto no es una ciencia exacta y en algunos puntos es un tema muy subjetivo. También existen otras herramientas dentro de la I.S. que serán mencionadas brevemente a lo largo del ensayo por su gran utilidad a la hora de diseñar.

## **¿Qué es un documento de diseño y para qué sirve?**

Yo defino el documento de diseño como un muro de contención para el desarrollador. Una definición que puede sonar a chiste y que posiblemente a algunos les resulte una idea malévolamente perversa, ya que les podría parecer una forma de cortar las alas a su instinto creativo. Pero la realidad es que a la hora de hacer un videojuego, el perfeccionismo o la constante actualización de nuevas características al proyecto, solo logran hacer que este se retrase e incluso pudiera llegar a cancelarse. Como en todo, obviamente hay que tener suficiente mano izquierda para saber cuando hay que añadir algo que realmente es necesario y cuando otras cosas simplemente hay que dejarlas para la segunda parte del juego y se acabó.

Dicha mi definición sobre la función del documento de diseño, pasaré a definirlo de una forma un poco más seria. Para hacer un buen juego, que no acabe pareciendo que está a medio hacer, como diseñadores hemos de tener muy claro lo que queremos hacer. ¿Pero cómo evitar olvidar detalles en el transcurso de los meses? Porque no todo el mundo tiene una memoria de elefante a prueba de balas. ¿O cómo hacer que la visión del grupo de desarrollo con el que trabajas sea homogénea? Porque si nos olvidamos de pequeños detalles y nos los vamos reinventando sobre la marcha cada vez que alguien nos pregunta por ellos, el juego resultante puede ser una conjunción de disparates desproporcionados. Por eso, para evitar que las cosas se desmadren por culpa del olvido y para hacer llegar a cualquier ser humano la visión del diseñador, es necesario tener una guía para este viaje que es el desarrollar un videojuego.

Y ahí está, el documento de diseño viene a ser un manual de instrucciones sobre el funcionamiento de nuestro juego escrito en palabras y no en código, para ayudarnos en la tarea de hacerlo realidad. Por ello debemos dedicarle el tiempo que sea necesario hasta estar bien seguros de lo que deseamos hacer, para lograr avanzar lo máximo posible en la dirección correcta. Y aun así nos encontraremos con que a medida que avanzamos en el desarrollo tendremos que actualizarlo y cambiar aspectos importantes de los que no nos dimos cuenta para darle una buena calidad al juego, procurando no caer en el bucle infinito del perfeccionismo obsesivo, ni en el miedo a tener que rehacer trabajo hecho si tenemos una buena justificación para ello. Es por cosas como esta que la mejor forma de aprender a hacer documentos de diseño, está más basado en la práctica y la experiencia personal, que en fórmulas matemáticas.

### **Una posible estructura a seguir para un documento de diseño**

Después de afirmar que la forma de hacer este documento es más un arte que una ciencia, no quita que en algunos detalles determinados sobre este tema, sí existan algunas recomendaciones y algunas plantillas a seguir, sobre todo si es la primera vez que te pones a hacer un documento de diseño. Una cosa importante a tener en cuenta, es que dependiendo del tipo de género al que pertenece el juego que se quiere desarrollar, el documento contendrá una serie de apartados que en otro género lo mismo no tendrá o será algo muy breve.

Existen por internet diversos ejemplos de documentos de diseño creados para algunos juegos famosos ([DooM](#), [Leisure Suit Larry](#)) o de diseñadores famosos ([Chris Taylor](#)), que son fácilmente localizables gracias a nuestro gran amigo google y de ellos podremos sacar bastantes ideas interesantes para hacer nuestro documento de diseño. Pero ya que estamos expondré de forma esquemática la composición de un documento de diseño, para luego explicar punto por punto su contenido:

Parte I: Introducción.

1. Historial de cambios en el documento.
2. Resumen ideológico del juego.
3. Lista de características.

Parte II: El universo del juego.

4. La historia.
5. Los personajes.
6. El jugador.
7. El mundo.
8. Los enemigos.
9. Las entidades.
10. Los objetivos o misiones.
11. Los diálogos.
12. La interfaz de usuario.
13. La mecánica del juego.
14. La mecánica del multijugador.

Parte III: Los detalles técnicos.

15. El entorno de programación.
16. La arquitectura del juego.
17. El renderizado.
18. La cámara.
19. La lógica.
20. El sonido.
21. La red.
22. Las herramientas.

Parte IV: El apartado artístico.

23. El estilo visual general del juego.

24. El estilo visual de los personajes.
25. El estilo visual de las entidades.
26. El estilo visual de los enemigos.
27. El estilo visual del mundo.
28. Listado de texturas, imágenes, tiles y/o sprites.
29. Listado de modelos 3D estáticos.
30. Listado de modelos 3D animados.
31. Listado de escenarios 3D.

Parte V: El apartado sonoro.

32. El estilo sonoro general del juego.
33. El estilo de la banda sonora.
34. Listado de sonidos ambientales.
35. Listado de diálogos hablados.
36. Listado de pistas de música.

Apéndices.

37. Créditos.
38. Nomenclatura del proyecto.
39. Apéndice para extender más algún apartado anterior en concreto.
40. Otros temas no cubiertos por los apartados anteriores.

Recordad que esto es un esquema y que dependiendo del tipo de juego, algunos puntos no tendría sentido ponerlos, ya que en un Tetris por ejemplo, no tiene sentido hablar sobre historia, personajes o enemigos, a no ser que pretendáis darle un enfoque distinto claro.

### **Punto 1: La introducción al documento de diseño**

La introducción en el documento de diseño sirve principalmente para hacer una lista de las directrices principales propias, con las que pretendemos definir de forma concisa el proyecto. Lo primero es tener un **historial de cambios**, para ir registrando los grandes cambios que se han ido realizado al documento. De esa forma siempre podremos ver su avance en cuanto a al diseño.

Lo segundo es el **resumen ideológico**, que viene compuesto por:

- + Filosofía del proyecto
- + ¿Cuál es la idea del juego?
- + ¿Dónde transcurre la acción?
- + ¿Qué controla el usuario?
- + ¿Cuántos personajes se controla?
- + ¿Cuál es la finalidad?

Siendo un resumen, no es necesario ser muy exactos en lo que hace el juego, ya que lo que se pretende con esta introducción es definir a grandes rasgos de qué va este y cuál es la motivación detrás de él. Lo tercero y último es la **lista de características** soportadas, es decir qué cosas se pueden hacer y/o qué se va a permitir hacer al jugador (en inglés a esto se le llama "features"). También se puede poner una lista de las mecánicas que se pretenden soportar. Dentro de este apartado tenemos:

- + Características principales
- + Características secundarias
- + Para futuras versiones

Las diferenciación entre características principales y secundarias viene a ser lo que tu consideras que es más importante y que define el alma de tu juego (características principales), y qué consideras que sería interesante tener, pero que no cambiaría demasiado la esencia de este si no las incluyeras (características secundarias). Además tenemos el tercer sub-apartado de "para futuras versiones", donde podemos colocar todas aquellas características que estarían bien tener, pero que tenemos que desechar para no caer en una

espiral infinita de desarrollo, también se le suele llamar "Para futuras secuelas" el que lo prefiera. Un ejemplo mejor de estas cosas es el siguiente:

- + Vamos a hacer un juego de naves como el Ikaruga o el 1942 y como característica principal tenemos "El jugador manejará una nave espacial", como se puede ver sin esto no tendría ningún sentido el juego.
- + Luego tenemos como característica secundaria "Los modelos 3D soportarán normal-mapping", que es una característica chula pero que en caso de no tener tiempo para desarrollar el juego completamente, si la fuéramos a descartar de este no perdería su esencia (aunque en este caso en particular perdiéramos calidad gráfica).
- + Y como característica para ejemplificar el tercer sub- apartado, imaginemos que hemos tras una reunión de brain-storming llenando hojas de ideas interesantes, pero todas no las podemos meter si queremos terminar el juego en un plazo razonable de tiempo, por lo que imaginemos que no nos da tiempo para meter algo como "El jugador podrá comprar piezas para mejorar la nave en las tiendas", porque tendríamos que emplear mucho tiempo y preferimos hacer un proyecto simple primero, pues es la clase de cosas que se pondrían en "Para futuras versiones". Ya que si hacemos una secuela, siempre podemos recolectar todas aquellas ideas que se nos quedaron en el tintero para afrontar el desarrollo de algo nuevo y novedoso para nosotros como desarrolladores.

## **Punto 2: El universo del juego**

En este segundo punto va todo lo referente con el funcionamiento de la mecánica del juego, qué lo compone y como se comporta este. Lo primero a tener en cuenta, y creo que normalmente es algo bueno a tener en consideración, es **la historia** que vamos a contar. Como dije antes, algunos tipos de juegos no requieren historia, como por ejemplo podría ser el caso de uno de fútbol, pero ya que estamos imaginemos que tenemos uno que sí tiene historia. Con estas cosas uno puede ser todo lo detallado que quiera, así que cuanto más se ponga mejor nos vendrá, sin tener miedo de tener que cambiarlo si fuera necesario. Dependiendo del tipo de juego, podemos tener ramificaciones en la historia que dependan de las acciones del jugador y la cosa se puede complicar mucho, por lo que siempre podemos poner como historia los hechos más genéricos que tienen en común todas las ramificaciones, y luego en un apéndice especificar todas las ramificaciones y las implicaciones de estas. Claro que esto ya son cuestiones de gustos. De hecho incluso podemos poner el tema de las ramificaciones, si está muy unido a ello, en el apartado de los objetivos que tiene que conseguir el jugador o las misiones que tiene que hacer.

El segundo apartado son **los personajes** del juego, que aparecen en el argumento descrito en el anterior apartado. Dentro de este apartado podemos hacer las distinciones que nos parezcan oportunas, como diferenciar entre aliados y enemigos, entre principales y secundarios, o no hacer diferenciación alguna. La importancia de este apartado se debe a que es una buena forma de tener reunida la información en forma de ficha sobre los personajes, con cosas como su aspecto, datos personales como la fecha y lugar de nacimiento, su carácter y/o psicología, la historia "completa" de su vida, la repercusión que tiene, lo que pretendemos representar con su figura, etcétera. Cualquier cosa que podamos aportar siempre será útil para la hora de tener que definir como se va a comportar en el juego, la clase de cosas que va a decir y sobre todo será útil para que los grafistas tengan una referencia de lo que tienen que crear. Porque no es lo mismo diseñar un personaje alegre y bondadoso, que uno oscuro y despiadado.

En el siguiente apartado se trata el tema de todo lo relacionado con **el jugador**, que viene a ser cual es la forma con la que este interactúa con el mundo o qué es lo que como jugadores vamos a controlar en nuestro juego. Tenemos que especificar si se trata de un personaje o varios, y qué es lo que este o estos le permiten al usuario hacer (por ejemplo si pueden montarse en vehículos, o si pueden usar objetos del usuario para diversos propósitos). Por ejemplo si estamos en un juego de acción, normalmente el jugador lo que manejará es a un personaje que tendrá una serie de acciones que puede realizar en un mundo, pero si es un RTS el jugador lo que maneja es un ejército, que no es un solo monigote sino muchos. Del mismo modo en un juego de carreras lo que manejamos es un coche y en un shoot'em up

como el 1942 lo que manejamos es un avión. Siendo un poco más ordenados podríamos decir que la lista de cosas que tenemos que indicar serían:

- + ¿Tiene el jugador un avatar que lo represente en el mundo? Y si lo tiene, ¿cómo es este avatar? ¿Existen diferentes tipos de avatar? (Si el avatar del jugador es un coche, lo normal es que existan diferentes tipos de coche que se pueden controlar en el juego.) ¿Puede el jugador configurar el aspecto y creación del avatar (sexo, altura, color del pelo, de los ojos, etcétera)?
- + ¿Qué es lo que el jugador controla en el juego? ¿Su avatar, si lo tiene, o un grupo de entidades? ¿Qué propiedades, características o datos componen aquello que controla el jugador y al jugador en sí? ¿Qué aspecto tiene? ¿Y cuáles son los mecanismos que se dispone para interactuar con el entorno o las acciones con las que el jugador puede interactuar con aquello que controla? (¿Puede moverlo? ¿Puede mandarle disparar? ¿Puede hacer que se enamore de otra entidad? Etcétera...)
- + ¿Cuál es la lista de todos los posibles objetos o entidades con las que el jugador puede interactuar? ¿Cuáles son las acciones posibles que puede realizar con cada una de dichas entidades u objetos? ¿De entre esos objetos tiene la entidad o entidades que controla el jugador accesorios que pueden complementarla ya sea de forma meramente estética o funcional? (Armas, escudos, motores, recambios, vestidos, etcétera...) ¿Cuál es la lista de tipos de accesorios que tiene? ¿Y dentro de cada tipo cuál es la lista de características de cada modelo de accesorio, que lo diferencian de otro modelo?

Ahora voy a poner una serie de ejemplos aproximativos de varios casos, donde ver un poco por encima lo que estoy tratando de explicar:

- + En un juego de acción FPS:
  - El jugador tendrá un avatar que consistirá en un personaje del mundo que será un valiente soldado ruso que ama a la madre Rusia.
  - El jugador podrá elegir entre diferentes tipos de soldados a la hora de jugar. Está la infantería, los francotiradores, los granaderos, el espía, el doctor.
  - El jugador no podrá configurar el aspecto del avatar, serán modelos estándar iguales para todo el mundo.
  - El jugador controlará y asumirá directamente el papel del avatar que ha escogido.
  - El jugador tendrá las siguientes propiedades: nivel de vida, nivel de armadura, posición en el mundo, lista de armas, lista de municiones para las armas y la velocidad actual de su movimiento.
  - El aspecto de los diferentes tipos de avatares del jugador son: el soldado de infantería será..., el soldado francotirador será..., etcétera...
  - Las acciones que el jugador podrá realizar con los diferentes tipos de avatares se dividen en comunes y específicas del tipo. Las comunes son: moverse, disparo primario, disparo secundario, cambiar de arma, coger items por el escenario, darles uso, saltar. Y las específicas son: para el francotirador hacer zoom con su rifle de francotirador, para el granadero pegar patadas con las que derribar las puertas, para el doctor curar a la gente, y para el espía disfrazarse de soldado enemigo.
  - El jugador podrá interactuar con los siguientes objetos y/o entidades de forma común a todos los tipos de avatares: colisionar con el mundo y los objetos, coger armas del suelo y equipárselas, coger munición y dispararla, coger objetos clave, activar eventos secuenciales (triggers), etcétera... Y de forma específica el rifle de francotirador solo lo podrá llevar el francotirador, el lanza-granadas solo lo podrá llevar el granadero, el espía no podrá llevar armas más grandes que una pistola o un cuchillo y además podrá llevar inyecciones con veneno, etcétera...
- + En un juego de estrategia RTS:
  - El jugador tendrá un avatar que consistirá en un comandante un ejército galáctico.
  - Solo habrá un tipo o clase de avatar, el de comandante, pero se podrá elegir el bando en el que luchar: el imperio galáctico o la república espacial.
  - El jugador podrá elegir el aspecto de entre varios modelos pre-renderizados.
  - El jugador controlará al ejército del bando que ha escogido.
  - El jugador tendrá las siguientes propiedades: lista de edificaciones construidas, lista de unidades fabricadas, lista de edificios en construcción y su estado, lista de unidades en fabricación y su estado, la posición y vida de edificios y unidades, etcétera...

- El aspecto de los diferentes modelos a elegir serán el de tipo duro, que será una persona...; el de tipo sabio, que será una persona...; el de maquinador, que será una persona...; etcétera...
  - Las acciones que el jugador podrá realizar se dividen en comunes y específicas del tipo de entidad seleccionada. Las comunes son: seleccionar entidades. Las específicas son para las unidades: mover, atacar, atrincherar, construir (solo para los obreros), sabotear (solo para los espías), etcétera...; para los edificios: fabricar unidades.
  - El jugador podrá interactuar con los siguientes objetos y/o entidades de forma: colisionar con el mundo y los objetos, activar eventos secuenciales (triggers), etcétera... Y de forma específica las unidades podrán coger objetos clave, etcétera...
- + En un juego de coches:
- El jugador tendrá un avatar que consistirá en un corredor de carreras.
  - Solo hay un tipo de avatar, el de corredor de carreras, pero habrá distintos tipos y modelos de coches.
  - El jugador podrá configurar el sexo del avatar.
  - El jugador controlará el coche que conduce el avatar.
  - El jugador tendrá las siguientes propiedades: puntuación, nivel de vida del coche, posición en el mundo, puntos de control activados y la velocidad actual de su movimiento.
  - El aspecto de los diferentes avatares del jugador son: el corredor será..., la corredora será... Los tipos de coches serán: el modelo XYZ será...; etcétera...
  - Las acciones que el jugador podrá realizar son: acelerar, girar a la derecha, girar a la izquierda, frenar, frenar con el freno de mano.
  - El jugador podrá interactuar con los siguientes objetos y/o entidades: colisionar con el mundo y los objetos, coger power-ups, destruir el mobiliario urbano, atropellar al público despistado, etcétera...

(Nota: Allá donde pone etcétera en los ejemplos significa que habría que seguir poniendo características hasta acabar poniéndolas todas, pero siendo esto un ejemplo tampoco es plan de extendernos demasiado, con entender la idea es más que suficiente.)

Después de definir al jugador, le toca el turno **al mundo** donde transcurre nuestro juego, aunque a veces el mundo es algo que se define antes que el jugador, depende mucho de como veamos que nos resulta más útil ordenar la información. Tenemos que responder donde nos encontramos y como es este mundo con el que vamos a interactuar. Cosas como cuáles son las localizaciones clave si las hay, de qué forma viajamos por el mundo, cuál es la escala de este, cuáles son los objetos o entidades que lo componen, cuál es el modelo del clima que vamos a usar, cuál es el modelo de tiempo (si alternamos el día con la noche, o si es de noche, o si es de día solamente, etcétera) y cuál es su avance en la línea temporal, etcétera... son aspectos que podemos encontrar interesantes para definir y dar riqueza y profundidad a nuestro juego. Cualquier dato que se os pueda ocurrir sobre el mundo, siempre es buena idea dejarlo apuntado, ya sea algo funcional o decorativo.

De todo lo expuesto sobre **el mundo**, lo más importante es definir todas **las localizaciones** que nos vamos a encontrar y hago un hincapié en esto porque normalmente lo clásico en muchos juegos es encontrarnos una serie de misiones encadenadas una detrás de otra, que representan el mundo por el que nos movemos. Pero también nos podemos encontrar con localizaciones que no son misiones, como por ejemplo el caso de la tienda de recambios para nuestro coche, en un juego de carreras. Sin embargo en algunos géneros específicos donde la acción transcurre en una ciudad, planeta o universo por el que nos podemos mover a nuestro libre albedrío, podemos solo indicar las localizaciones clave, aunque cuanto más indiquemos mejor. No obstante es raro y poco aconsejable que se llegue a un detallismo que roce lo enfermizo y lo aconsejable es centrarse en lo más importante, de ese modo si tenemos una ciudad como mundo, detallaremos donde transcurren las misiones y donde se encuentran cosas como las tiendas en las que el jugador puede interactuar.

Lo siguiente que convendría definir a continuación es **los enemigos** que nos podemos encontrar en el mundo. Realmente se puede definir a un enemigo como una entidad, por lo que podríamos haberlo puesto en el siguiente apartado, el de las entidades. Sin embargo, en

determinados juegos "simples" el enemigo es fácilmente reconocible, ya que es aquel al que podemos destruir y que busca destruirnos. En otros más complejos resulta más difusa la línea entre enemigo y aliado, porque depende de nuestras acciones que los NPCs (personajes o entidades no controladas por el jugador) se comporten con nosotros de una forma u otra. Así que a la hora de definir los enemigos, se busca el definirlos de forma muy similar a como definimos al jugador: ¿Cuál es su aspecto, sus propiedades y características? ¿Cuáles son las acciones que puede realizar? ¿Cómo puede interactuar con el mundo que le rodea, con las demás entidades y con el jugador? Y como principal diferencia con respecto al jugador, tenemos que el enemigo tiene inteligencia artificial propia y tenemos que detallar cuál es el comportamiento del enemigo, si es en solitario o en grupo, si es muy listo o más tonto que un zombie.

Después de definir a los enemigos, tenemos **las entidades**, aunque como he dicho antes, sería perfectamente admisible definir a los enemigos dentro de estas. La definición de entidad en un juego, viene a ser algo así como cualquier cosa que no pertenece a la geometría estática del "mundo". Esto implica que se podría desplazar por cualquier lugar del mundo, aunque fuera una miserable caja de municiones, ya que por ejemplo con la "revolución" de la física ahora es fácil ver que cualquier cosa de ese tipo que antes estaban bien sujetas al suelo, ahora se pueden mover por donde el jugador le plazca. El propio jugador viene a ser una entidad en cierta forma o al menos el avatar o las entidades que maneja. Y dentro de estas tenemos las que son visibles y las que no. Entre las visibles las que están animadas y las que no, las que tienen comportamiento y las que no. Por lo que igual que se hace con los enemigos, con las entidades también tenemos que responder a: ¿Cuál es su aspecto (si lo tiene), sus propiedades y características? ¿Tiene comportamiento? ¿Si lo tiene, cuál es dicho comportamiento o inteligencia artificial? ¿Cuáles son las acciones que puede realizar? ¿Cómo puede interactuar con el mundo que le rodea, con las demás entidades y con el jugador?

Teniendo ya definidos todos los elementos que forman el mundo y existen en este, le toca el turno a **los objetivos o misiones** del juego. Todos ellos tienen un objetivo que cumplir para que este finalice satisfactoriamente, es lo que se llama la condición de victoria y es importante definirla, porque no siempre es algo trivial y es mejor dejarlo claro. Si por ejemplo se tienen misiones que cumplir a lo largo de este, también tendremos que definir en qué consisten dichas misiones, cuáles son sus propiedades o características, cuáles son sus mecánicas de funcionamiento y cuál es la condición de victoria para completar la misión, teniendo en cuenta que podemos ofrecer una o varias. De esta forma, cuando un juego está dividido en misiones, normalmente la condición de victoria final es pasarse todas las misiones principales, ya que podemos tener misiones secundarias que no sean necesarias para finalizarlo con éxito. Si no está dividido en misiones, niveles o fases, tendremos que indicar cuáles son los objetivos que el jugador tiene que cumplir para salir victorioso. Como en los anteriores apartados, todo detalle que complementa la información del diseño del proyecto, vendrá siempre bien para tener todos los cabos bien atados.

Siempre que tengamos un juego con una historia, será normal que existan **diálogos** entre los personajes, para avanzar la historia y la relación que existe entre los personajes. Normalmente es algo que sería lógico meter en la historia, pero cada vez es más normal tener más libertad de acción y poder hablar con montones de NPCs que dan detalles que enriquecen y dan profundidad al universo del proyecto, pero que no son diálogos principales por decirlo de un modo. Distinguir entre un diálogo principal y uno secundario u opcional, es sencillo, normalmente uno secundario solo será mostrado al jugador si este se dedica a ir hablando con el NPC en cuestión, pero uno principal normalmente suele estar en escenas cinemáticas de esas que interrumpen la acción del jugador para contarle la historia principal.

Otro punto más sobre el funcionamiento del juego y muy importante es **la interfaz de usuario**, pues de ella depende la totalidad de la forma en la que el jugador interactúa con este. Normalmente hay que especificar los controles de los que dispone el jugador (teclado, ratón y/o gamepad), y si tiene la opción de poder configurarlos a su gusto y en qué grado. Luego hay que definir como va a ser el HUD que tendrá en pantalla el jugador mientras juegue, qué elementos lo componen, la información que representan y la apariencia de dichos elementos. Por último y no menos importante, también hay que definir como va a ser

el menú del juego, igual que con el HUD, solo que el menú normalmente dispone de muchas pantallas por las que navegar, aunque no suele ser un elemento muy difícil de diseñar. Normalmente es más fácil el diseño del HUD que el del menú, pero en algunos géneros como el de los RTS suele ser al revés, el HUD es bastante más difícil de diseñar que el menú. Hay que tener en cuenta que el menú y el HUD son bastante importantes, para dar una buena apariencia al proyecto y que no parezca muy cutre, además el HUD es algo con lo que el jugador va a convivir todo el tiempo en su experiencia de juego, por lo que si tenemos un HUD horrible, tendremos una mala experiencia. Así que son detallitos, pero que hay que cuidarlos para dejar un buen sabor de boca.

Ya después de definir todas las piezas del puzzle tenemos **la mecánica del juego**, que viene a ser como se interconecta todas las piezas. Aunque dependiendo de como lo deseemos muchos aspectos los habremos expuesto en los apartados anteriores o los podemos exponer aquí. Indicar cuál es el funcionamiento, como si de un juego de mesa se tratara, de temas como las misiones, el combate, la interacción con diferentes elementos del mismo, etcétera, es el cometido de este apartado de una forma completamente detallada y sin lagunas. También es un buen punto en donde indicar cuál es la duración estimada que esperamos que tenga en horas. Igualmente es un buen lugar para indicar cuáles son las condiciones de terminación del juego, ya sea porque el jugador ha sido derrotado o porque este ha salido victorioso y lo ha finalizado.

Por último queda el apartado de **la mecánica del multijugador**, que no todos los juegos tienen multijugador, por lo que muchas veces es un apartado que no tendremos en nuestros documentos de diseño. Pero en caso de tener que realizar un apartado multijugador para nuestro proyecto hay diversas cuestiones que deberíamos responder como cuál es el máximo número de jugadores que pretendemos soportar, cuál es el funcionamiento de los servidores y del cliente, cuáles son las opciones de configuración disponibles para el jugador o si se va a contemplar la idea de tener persistencia de datos o no (como ocurre en los MMORPG).

### **Punto 3: Los detalles técnicos**

Explicado ya con palabras como funciona nuestro juego, este punto servirá para entrar un poco más en detalles técnicos sobre la implementación del mismo sin tener que llegar necesariamente a poner código. Lo primero que se tiene que decidir es **el entorno de programación** y la plataforma para la que va a ser desarrollado, porque dependiendo de lo que se elija tendremos unas limitaciones u otras. También podemos indicar las API, frameworks o motores que vayamos a querer usar.

Decidido el entorno, pasamos a **la arquitectura del juego**, para lo que podríamos usar herramientas como UML para representarla de una forma relativamente sencilla. La idea es indicar de forma global como está compuesto este, las piezas que lo forman y como encajan. No es necesario profundizar mucho para este punto, ya que eso lo podemos hacer en los siguientes apartados.

Con el apartado del **renderizado** se pretende mostrar cuál es el funcionamiento de esta parte del juego, indicando los algoritmos que se vayan a usar, si va a ser un renderizado 2D o 3D, definiendo los detalles sobre el modelo de iluminación y sobre el renderizado del mundo o de las entidades. Para ello a parte de las palabras también puede resultar útil tener diagramas de clase, de estado y de secuencia hechos con UML.

En el apartado de **la cámara** lo que se busca es definir qué tipos de cámaras se van a tener, como funcionan y como se van a comportar con respecto al mundo en el que van a estar. Cosas como el tratamiento de las colisiones de la cámara con el escenario son cosas a tratar aquí.

Tras todo lo anterior, en el apartado de **la lógica** nos encontraremos algo parecido a lo que hicimos con el renderizado, pero aplicándolo a la lógica interna del juego. Principalmente consiste en indicar como funciona el algoritmo mediante los diagramas UML que

necesitemos para ello. También se puede meter aquí qué es lo que se ha de hacer cuando se salva o se carga una partida, el flujo de datos que ello conlleva. Y desde luego es el lugar idóneo para explicar el funcionamiento de las colisiones.

Con el apartado del **sonido** trataremos de explicar el funcionamiento de este en nuestro juego, si usaremos sonidos en 3D o no, como utilizaremos los canales de audio para tener música, sonido ambiente y voces al mismo tiempo.

El apartado de **red** que bien podría llamarse el multijugador, explicaremos qué algoritmos usaremos para la gestión del servidor y del cliente, si usaremos tecnologías de comunidades virtuales como el Steam o el Xbox Live, si en el caso de tener persistencia en nuestro multijugador, como va a ser dicha persistencia y como funcionará, cómo vamos a mantener la seguridad y evitar que los jugadores hagan trampas.

Y después de todo esto tenemos **las herramientas**, que es un tema bien importante porque cada vez es más habitual tener que hacer herramientas para que los diseñadores de niveles hagan los niveles o fases del juego, o que los grafistas necesitan ver como quedan sus modelos 3D con la tecnología del mismo, y también exportadores de un formato a otro. Todo aquello que queremos facilitar a los diseñadores y grafistas, tenemos que indicarlo aquí y hacer un análisis del software para ello, utilizando diagramas UML para representar el funcionamiento de las herramientas y haciendo especificaciones de la organización de los ficheros de datos que usamos para almacenar el contenido del proyecto.

Por último, para aclarar qué es esto del [UML](#) por si no le suena a alguno, decir que es una herramienta del análisis del software y sirve para representar el análisis y diseño de una aplicación a distintos niveles, ya sea en su comportamiento o en su estructura. Existen aplicaciones para realizar diagramas en UML que son gratuitas como [ArgoUML](#). Y también hay tutoriales que se pueden encontrar desde google para aprender este "lenguaje". Para el que sepa inglés y un poco de qué va el tema aquí tiene un enlace a una página con una [referencia completa](#) a la simbología del UML.

#### **Punto 4: El apartado artístico**

Este tercer punto como programadores, que suelen ser los que se dedican a hacer estas cosas tan majas de los documentos de diseño y el análisis del software, puede que sea un poco menospreciado igual que con el sonoro. Pero no hay que caer en ese error, ya que este apartado puede resultar muy importante también, ya que en él se va a recoger cuáles son las directrices principales en el apartado artístico. La importancia de esto radica en que para evitar tener que repetirse trabajo, hay que encontrar una forma de hacer llegar hasta los artistas la visión que se tiene sobre el estilo visual del juego. Por eso todo los datos y referencias posibles que puedan facilitar a los grafistas el hacerse una idea de qué le estamos pidiendo, será tiempo y disgustos que ahorremos tanto nosotros como ellos. También es una buena idea incorporar bocetos (si uno tiene talento para ello), fotografías o imágenes que encontremos por ahí que sean representativas de lo que queremos conseguir.

Además también es el lugar oportuno para hacer un listado de todo el material gráfico que se necesita y todas las posibles características que ha de cumplir, tales como: el nombre del fichero, descripción del contenido, el formato del fichero; y si se trata de una imagen: la profundidad de color y la resolución. Si podemos poner todas esas, es trabajo que nos vamos a ahorrar de tener que ir con el photoshop a redimensionar las imágenes y reconvertirlas a otro formato, porque como no se lo habíamos indicado al grafista lo que queríamos exactamente, este nos lo ha pasado tal como lo tenía. También viene bien saber todo lo que vamos a necesitar, por si somos un desarrollador de medios limitados y tenemos que pedir un presupuesto a un grafista freelance, ya que tendremos que saber qué le vamos a pedir, a fin de no gastar de más por no haber sabido planificar lo que necesitábamos realmente.

#### **Punto 5: El apartado sonoro**

Exactamente igual que con el tema gráfico, tenemos el tema sonoro, que hay que planificar bien para evitarnos disgustos de más. Tenemos que saber qué es lo que queremos y qué pretendemos transmitir con la música en nuestro juego, y más importante, tenemos que saber hacérselo llegar al músico. Por ello aquí también resulta muy importante recopilar todo tipo de datos y referencias a fin de facilitar a los músicos su trabajo. Para indicar mejor el estilo de música que queremos para nuestro proyecto, podría ser una buena idea buscar canciones de grupos de música, música clásica, bandas sonoras, etcétera... que se acerquen bastante a lo que deseamos conseguir para el juego, no olvidando explicar para qué vamos a usar ese sonido o música, y qué es lo que queremos transmitir en ese momento al usarla.

Tampoco hay que olvidar los efectos especiales o sonidos de ambiente, hay que tener todo tipo de referencias dedicada a estos, describiendo cual es el uso que se le pretende dar y lo que se intenta transmitir con ellos. No obstante en géneros como el survival horror, los sonidos ambientales son una pieza clave para generar el ambiente del mundo. Además de la música y los sonidos ambientales, tendremos también que planificar y organizar el doblaje si vamos a disponer de este, recopilando todos los diálogos que se quieren doblar para pasárselos a los actores y gracias a haber hecho los perfiles de los personajes, podremos ayudar a los actores a que descubran cual es el trasfondo emocional de aquello que interpretan, para poder darle una mayor calidad.

Por razones similares a lo que ocurría con el material gráfico, tendremos que tener un listado del material sonoro que vamos a necesitar, con todas las posibles características que ha de cumplir, siendo estas: el nombre del fichero, descripción del contenido, el formato del fichero, si se trata de un sonido en mono o en estereo y la frecuencia del mismo. De este modo será más fácil poder encargar a otros la realización del apartado sonoro. Algo que posiblemente simplificaría el listado es poner a modo de cabecera, aquellas propiedades que cumplan un conjunto de ficheros, que normalmente suele ocurrir que el formato, la frecuencia o si es mono o estereo, son datos que se repiten de forma idéntica a lo largo del listado. Por ello encontrar una forma de sintetizar nos ahorrará cierto trabajo y también se puede aplicar si es posible al listado de material gráfico.

## **Punto 6: Los apéndices**

Y después de todo los puntos anteriores queda simplemente los apéndices, donde podemos poner cualquier cosa que consideremos interesante de saber, pero que no forma parte realmente de los otros puntos, como es los créditos o la nomenclatura a seguir en el proyecto a la hora de programar. También podemos vernos en la necesidad de añadir algún apartado a los apéndices para extender algún apartado de un punto anterior, a fin de dejar más claro el documento de diseño si lo consideramos oportuno, como podría ser el caso del listado del material gráfico o sonoro, o los listados de entidades. Sin embargo eso ya son cuestiones de gustos y manías de cada uno. Para bien o para mal, no existe una forma única y verdadera de hacer esto del documento de diseño, sino que existen múltiples.

## **Conclusión**

Finalmente quiero aclarar que este es el esquema que a mi me parece más lógico normalmente, aunque el orden de los elementos puede variar dependiendo de las necesidades, e insisto en que esto no es una ciencia exacta, sino un aprendizaje basado en la práctica y la necesidad. Así que no temáis saliros del esquema si eso os facilita hacer vuestro trabajo, pero sabed que en la medida de lo posible no evitéis poner información, porque cuanto más puntos estén bien explicados, menos opciones de salirse por la tangente hay.

Espero de todo corazón que este documento os pueda servir de ayuda a todos aquellos que queréis desarrollar videojuegos y os agradezco que hayáis aguantado todo este rollo hasta el final. Así que animaros a hacer juegos y suerte con vuestros futuros proyectos.

Un saludo,  
Gorka Suárez.

## **Agradecimientos**

Quiero agradecer a RobiHm y Synchrnzr por su colaboración con este documento para mejorar su contenido, hacerlo más legible y ayudarme a corregir errores.