



UNIVERSIDAD NACIONAL DEL LITORAL
FACULTAD DE INGENIERÍA Y CIENCIAS HÍDRICAS



Tecnicatura en diseño
y programación de videojuegos

UNL VIRTUAL



Diseño de videojuegos I

Unidad 2

Level 5, Parte II: Gameplay orientado a una audiencia

Docente
Francisco Cascallares

CONTENIDOS

1. Afinar y afilar el gameplay: Prototipo inicial	2
2. Tres reglas prácticas para hacer prototipos.....	3
3. Primero una torre, luego el mundo	4
4. Audiencia núcleo	6
5. Tipos de audiencias núcleo.....	6
6. Definiendo una audiencia	8

1. Afinar y afilar el gameplay: Prototipo inicial

Ahora, ¿para qué sirve *concretamente* todo esto que hemos visto?

Para entender qué tipo de juego hacemos. Es decir, *para quién* lo hacemos.

No siempre nos hacemos espontáneamente esta pregunta, o sabemos responderla bien, y sin embargo es una de las cuestiones fundamentales para empezar a entender el tipo de *gameplay* inicial que propone (o *debería* proponer) nuestro juego. Y, al entenderlo, podemos empezar a afinar ese *gameplay* como se afina la cuerda de un piano o de una guitarra. Podemos empezar a afilarlo como se afila la hoja de un hacha... un hacha doble de adamantio +4.

Y, así, podemos orientarlo con precisión y astucia hacia la experiencia específica que va a movilizar y a sorprender a la audiencia para la cual está imaginado nuestro juego.

De esta manera, damos en cualquier proyecto nuestro primer paso concreto (y compartible con otros) hacia la producción real de nuestro juego: el *early prototyping* (traducido literalmente como “prototipado temprano”, aunque la verdad es que puede hacerse a cualquier hora del día o la noche; una traducción más correcta sería “prototipado inicial”).

Este es un tema que podría discutirse eternamente, pero digamos que básicamente hay dos modos de empezar a hacer un juego: hay quienes prototipan y quienes no. Quienes no prototipan, prefieren depender de lo que *imaginan que será* y de *cómo imaginan que será* la jugabilidad que plantean; la explican de entrada y completa en un documento y ajustan y modifican ese documento (y la jugabilidad misma) durante la producción. Esta es una práctica tal vez frecuente pero muy arriesgada. Hoy, en una semana o en un día (dependiendo de la complejidad de nuestro desarrollo) podemos saber si algo funciona o no *probándolo en pantalla* en vez de *imaginando cómo sería*. ¿Cómo? Creando prototipos rápidos, desprolijos y descartables, hechos en lenguajes ágiles y simples, con cuadraditos y círculos de colores en vez de arte final, iterando sobre ellos una y otra vez hasta que el núcleo del juego -aquella mecánica inicial- está afinado, probado, y resulta entretenido. Esta es la otra manera de empezar a hacer un juego, basada en pruebas empíricas y ciclos de mejoras incrementales.

El argumento clásico a favor del prototipado es más o menos el siguiente: si es cierto (y hasta ahora la práctica ha comprobado que lo es) que un juego *nunca* termina siendo tal como se lo imagina al principio, prototipar inicialmente nos ahorra más adelante semanas o meses de dolores de cabeza y pérdidas económicas, porque tarde o temprano deberemos cambiar, modificar, alterar, y hasta descartar por completo pedazos completos de nuestro juego... ya escrito en un código elegante y sólido (que tomó tiempo pulir) y con un arte impecablemente desarrollado (que tomó tiempo desarrollar). Prototipando desde el inicio, como primer paso concreto de un proyecto, tenemos pruebas de entrada y para siempre de que el juego posee una jugabilidad apropiada, porque es lo primero que se afina y se afila, y no se construye más allá de este aspecto hasta que este funcione satisfactoriamente.

El argumento alternativo que les propongo a favor del prototipado es el siguiente. Un diseñador no crea juegos: diseña experiencias. Estas son experiencias de juego. Por ende, un diseñador diseña *gameplay* -la experiencia que le ocurre a un jugador al interactuar con un juego momento a momento gracias a la suma de todas sus cualidades. Todos los aspectos de un juego poseen un componente de *gameplay*, que el diseñador inventa y afila y afina a lo largo del proceso de desarrollar un juego. Necesitamos, entonces, empezar a afilar y afinar cuanto antes. No nos sirve hablar en abstracto sobre el juego. Necesitamos convertirnos en jugadores de nuestro propio juego lo más pronto posible, interactuar con este sistema y empezar a ajustarlo obsesivamente... un proceso constante en el que estaremos inmersos durante el resto del proyecto y que verdaderamente constituye la tarea -y la habilidad- más importante, en mi punto de vista, de un diseñador de juegos.

Con el prototipado, estamos entrando en un tema avanzado y demasiado extenso para el espacio y el tiempo con los que contamos. Aquí, nos interesa señalar puntualmente una relación que tiene el prototipado con esta etapa inicial de afinar y afilar el *gameplay* mediante un proceso empírico, ágil e inmediato. La cuestión de *cómo* afinar

y afilar el *gameplay* -es decir, la cuestión de qué estrategias de diseño aplicaremos- las iremos desglosando de los distintos temas que iremos viendo en esta materia.

2. Tres reglas prácticas para hacer prototipos

Veamos ahora algunas cuestiones básicas sobre el prototipado inicial, suficientes como para saber que esto existe y de qué se trata en líneas generales. Existen tres reglas prácticas que es importante tener en cuenta a la hora de encarar prototipos.

El prototipado rápido responde a una sola pregunta

Estas preguntas son hipótesis que tenemos, en general acerca de mecánicas –cosas que suponemos, imaginamos, acerca de cómo funciona un aspecto puntual y aislado de nuestro juego, pero que por supuesto no hemos llegado a comprobar todavía. Por ejemplo, ¿es divertido dibujar líneas moviendo las flechas del cursor mientras presiono la barra espaciadora? Podemos imaginar que puede serlo, y quedarnos con esa respuesta, y escribir todo el documento de juego basados en que la hipótesis es verdadera. O podemos crear un prototipo rápidamente y comprobar por nosotros mismos –y buscando la opinión de los otros mediante *playtesting*– si es o no así. Es posible que descubramos que, después de todo, no es tan divertido hacer esto con las flechas. Ante este “no”, iteramos: creamos una nueva pregunta en base a modificaciones sobre la respuesta anterior, tal como: Entonces, ¿es divertido dibujar líneas arrastrando el mouse? Seguramente sea mucho más divertido –pero cabe la posibilidad de que tampoco sea interesante, o intuitivo, o ágil: habrá que comprobarlo con un segundo prototipo. Si la respuesta es ahora afirmativa, entonces hemos hallado los controles ideales para nuestro juego. Y significa que esta mecánica tiene futuro y puede convertirse en un juego interesante. Y, si no, habrá que seguir intentando otras opciones o bien tomar la decisión de descartar esta idea y pasar a otra.

Eso significa: desprolijo, descartable, sin arte (los prototipos son idealmente casi o totalmente abstractos: están compuestos de *placeholders*¹ que representan a los objetos que más adelante aparecerán en el juego final). Se emplean lenguajes veloces, que permiten maquetar rápidamente un prototipo funcional, como Actionscript 3, e incluso engines de creación de juegos, librerías y frameworks como Game Maker, Unity3D, Panda 3D, Alice, Flixel, Stencil... Todo vale, y nada de esto queda en el juego final.

El prototipado rápido es rápido

El prototipo a veces ni siquiera se parece a nuestro juego (porque muchas veces eso lleva tiempo). Esto también significa un día de programación y no un mes, o dos días en vez de dos semanas, o tres horas en vez de tres días. El menor tiempo que podamos lograr para generar un prototipo jugable es el mejor tiempo, porque nos deja espacio para iterar más sobre lo que ya tenemos y probar otras hipótesis, o ir sumando algunas hipótesis a las que vamos acumulando.

¹ *Placeholders*: literalmente, “reservan” (*hold*) un “lugar” (*place*). Al prototipar, hacemos dibujos de figuras con palitos, bajamos imágenes de internet, usamos puntos y cuadraditos en vez de dragones y guerreros, –hacemos lo que sea. El placeholder es descartable y siempre debe parecer un placeholder: algo que reserva el lugar de otro objeto. Jamás se debería confundir con una imagen que sí será parte del juego final. Cuanto más se parece un prototipo al juego que tenemos en mente (es decir, cuanto más arte final o casi final usamos en el prototipo) menos rápido va a ser su desarrollo, y más problemas vamos a tener con la siguiente cuestión a tocar sobre prototipos.

El prototipado rápido genera un prototipo

Al prototipar, no estamos intentando un juego completo. Estamos desarrollando un prototipo que responde en tiempo récord a *una sola pregunta* que nos hacemos a la vez. Una sola pregunta es igual a una mecánica por vez, una funcionalidad o *feature* por vez. Partir nuestra mecánica base en una serie de preguntas puntuales es, a mi modo de ver, un arte, pero un tipo de arte que alguien con familiaridad en programación domina con facilidad: cada pregunta es igualable con una función o acción que hacemos en el juego.

Es verdad que en un prototipo conviene acumular dos o tres o cuatro hipótesis acerca de la mecánica base, en especial cuando se trata de mecánicas que hemos dividido en acciones muy aisladas. Un ejemplo: A la respuesta afirmativa de si es divertido dibujar líneas arrastrando el mouse, podríamos agregarle en un siguiente prototipo si es divertido hacer desaparecer mediante esa mecánica objetos que aparecen en la pantalla. Incluso (suponiendo que encontremos una respuesta satisfactoria) podríamos querer sumarle si resultaría interesante que algunos de estos objetos puedan combinarse únicamente con otros específicos en una misma línea (por ejemplo, sólo los del mismo color o los de la misma forma). Pero en cuanto cruzamos cierta línea, como por ejemplo empezar a “tunear” magos que hacen aparecer más objetos o menos dependiendo de nuestras acciones en el escenario, ya hemos empezado a hacer el juego y el prototipo inicial pierde rápidamente su sentido: deja de ser rápido y deja de ser un prototipo. Puede haber proyectos en los que la complejidad de un desarrollo puntual haga deseable prototipar algo más allá de una mecánica base, pero el prototipo tiene su máxima utilidad a la hora de afinar o afilar una mecánica base y ponernos en un punto de partida seguro e interesante. Ese es el punto en donde ya no continuamos con el prototipado inicial.

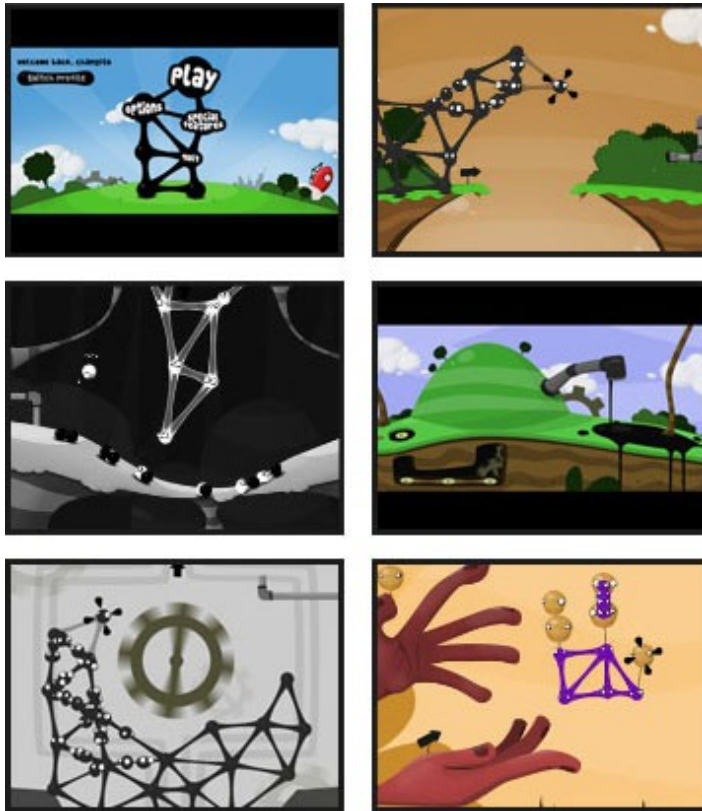
3. Primero una torre, luego el mundo

Gameplay es uno de los temas más vastos y complejos del diseño de juegos. Es posible que, como tema, atraviere *todos los otros temas* de esta disciplina, y tal vez sea la *razón de ser* de muchas ellas –por ejemplo, del prototipado. Continuaremos agregando estrategias de diseño de *gameplay* en los levels restantes, a medida que introduzcamos nuevos temas. Aquí, sólo hemos raspado la superficie para abrir el tema hacia cómo pensar en quienes jugarán nuestros juegos.

Como no hay nada mejor que la experiencia de ver algo con nuestros propios ojos, veamos cómo un prototipo armado en 4 días y conocido como *Tower of Goo*...



...terminó convirtiéndose en el multipremiado, hiperfamoso juego independiente *World of Goo*.



Los siguientes enlaces son las distintas revisiones del prototipo que podemos encontrar... más de 800 en total. Probemos algunas versiones. Podemos aprender mucho si prestamos atención a qué aspectos del juego y del *gameplay* van siendo afinados, descartados, mejorados, etcétera, a medida que se pasa de prototipo a juego y de juego al juego final.

1. <http://experimentalgameplay.com/egp1games/towerofgoo.exe>
(primer prototipo, hecho en 4 días)
2. <http://www.fun-motion.com/files/TowerOfGooUnlimited.exe>
(primera iteración para mejorar gameplay)
3. <http://2dboy.com/lifeofgoo/wog20060820-rev76.zip>
4. <http://2dboy.com/lifeofgoo/wog20060901-rev141.zip>
5. <http://2dboy.com/lifeofgoo/wog20060913-rev200.zip>
6. <http://2dboy.com/lifeofgoo/wog20061011-rev281.zip>
7. <http://2dboy.com/lifeofgoo/wog20061128-rev400.zip>
8. <http://worldofgoo.com/lifeofgoo/wog20070209-rev601.zip>
9. <http://2dboy.com/lifeofgoo/wog20070402-rev821.zip>
10. Y el trailer del juego final, para PC, Mac y Wii (video):
http://www.youtube.com/watch?v=A_JfkzPwww

Observación

Para los enlaces 7, 8 y 9 borrar antes `\HKEY_LOCALMACHINE\SOFTWARE\2dboy` del registro.

Casi todo este material está tomado de un fantástico artículo escrito (en inglés) en 7 partes por Ron Carmel y Kyle Gabler, los creadores del juego, que trata sobre cómo fue la evolución de *Tower of Goo* a *World of Goo*: “El mundo Goo no fue construido en un solo día”. Aquellos que estén interesados, pueden leerlo aquí:

Parte 1:

<http://2dboy.com/2009/03/06/the-world-of-goo-wasnt-built-in-a-day-part-1-of-7/>

Parte 2:

<http://2dboy.com/2009/03/09/the-world-of-goo-wasnt-built-in-a-day-part-2-of-7/>

Parte 3:

<http://2dboy.com/2009/03/11/the-world-of-goo-wasnt-built-in-a-day-part-3-of-7/>

Parte 4:

<http://2dboy.com/2009/03/17/the-world-of-goo-wasnt-built-in-a-day-part-4-of-7/>

Parte 5:

<http://2dboy.com/2009/04/26/the-world-of-goo-wasnt-built-in-a-day-part-5-of-7/>

Parte 6:

<http://2dboy.com/2009/05/12/part6/>

Parte 7:

<http://2dboy.com/2009/05/25/the-world-of-goo-wasnt-built-in-a-day-part-7-of-7/>

4. Audiencia núcleo

Vimos hasta ahora que la temática define en gran medida los juegos que diseñamos. Y que lo hace porque conecta al juego directamente con las *aspiraciones* de los jugadores (aquello que desean *ser* y *hacer*) a través del *gameplay* que ofrece. Es decir, al elegir una temática (y alguna clase de *gameplay*) muchas veces ya estamos definiendo qué tipo de audiencia (*target audience*) es la idónea para nuestro juego.

Esto no significa que sólo un tipo de gente juega a un tipo de juego, o que a alguien sólo le interesa una sola clase de experiencias. Pensemos que existe una audiencia principal, audiencias secundarias (que se “desbordan” de la principal, como veremos), y hasta una audiencia emergente o hasta inesperada (que nunca habríamos imaginado que se interesarían por nuestro juego).

La audiencia principal es la que más nos importa. Es nuestra *audiencia base* o *núcleo* (en el mismo sentido de una mecánica base o núcleo), de donde otras audiencias secundarias podrían desprenderse. Esta audiencia núcleo es la que define nuestro juego y la que va a vivir más intensamente, más personalmente, la experiencia que ofrece nuestro juego. Es para quienes nuestro juego resultará más significativo. Y hacia ella orientamos todos los aspectos del *gameplay*. En otras palabras, nuestro juego está afinado para hablarle al oído a esa audiencia núcleo que hemos logrado definir.

5. Tipos de audiencias núcleo

¿Cuál es, exactamente, la audiencia núcleo? ¿Para quién hacemos un juego puntual? ¿Hacia quién afinamos y afilamos el *gameplay*?

Podemos pensar que hay 4 tipos de jugadores posibles que conforman una audiencia núcleo:

- Yo.
- Nosotros.
- Ustedes.
- Ellos.

La audiencia núcleo de un juego *sólo incluye a uno de estos cuatro tipos de jugadores*. Obviamente, cuál es el que incluye cada vez varía con cada juego. Dependiendo de la naturaleza del proyecto, a veces lo define un cliente para quien trabajamos y otras veces debemos definirlo nosotros mismos. Es fundamental saber si estamos haciendo un juego para la audiencia apropiada o para otra.

- *Yo*: el juego está orientado, afinado, afilado, hacia nuestros propios gustos personales, hacia nuestros intereses, hacia un problema que estamos explorando.
- *Nosotros*: el juego está orientado hacia los gustos y preferencias de otros que comparten con nosotros una cantidad de preferencias y hábitos similares a los nuestros, por ejemplo nuestros amigos, u otros miembros de una comunidad englobada alrededor de un interés concreto que compartimos. Es muy fácil de olvidar que “yo” y “nosotros” no son exactamente lo mismo, que existen diferencias y es clave tenerlas en cuenta. Aquí otros pueden disentir con uno sobre aspectos particulares, aunque en lo general se suele estar de acuerdo.
- *Ustedes*: esta es una audiencia que el diseñador conoce pero en la cual no se incluye a sí mismo, y con la cual no comparte preferencias. Un juego orientado a un grupo de otra edad, o del sexo opuesto, por ejemplo.
- *Ellos*: es una audiencia tan alejada de nuestra experiencia y hábitos que no tenemos ningún conocimiento directo acerca de ellos. Son tan distintos que no sabemos cómo son ni qué prefieren, salvo tal vez de oídas y a través de lugares comunes que tienen poca utilidad real. Su mundo nos resulta extraño, incomprensible, porque no tiene relación con el nuestro.

¿Recuerdan cuando, en el level 4, hablamos sobre actitudes de un diseñador? La actitud de saber ponerse en el lugar del otro, de enriquecerse con otros puntos de vista, de abandonar momentáneamente nuestros preconceptos y preferencias para escuchar los de personas distintas, tiene una gran utilidad aquí. Es muy frecuente que, en mayor o menor medida, los juegos que se nos ocurren tengan alguna orientación directa o indirecta hacia nosotros –más allá de los verdaderos intereses de la audiencia real de nuestro juego. Incluso cuando la audiencia apropiada se trate de gente muy distinta a nosotros, es fácil caer en el error de agregar elementos de *gameplay* que no tienen relación con los intereses o habilidades de nuestra audiencia.

Es natural que, desde que empezamos a desear ser desarrolladores de juegos, hayamos inventado juegos orientados a los tipos de jugador 1 y 2: juegos a los que nos gustaría mucho jugar, y que gente con nuestros mismos gustos también disfrutaría, como nuestros amigos. Es perfectamente válido, y algunos de los juegos más inventivos o conocidos apuntan a estos dos tipos de jugador: “lo que a mí me gusta, y a otros como yo les gusta”. Si pensamos, por ejemplo, en los juegos independientes, una gran parte de ellos no llega a sorprender a jugadores que prefieren otros juegos, como los AAA. Es decir, están diseñados por desarrolladores *indie* para que el resto de la comunidad *indie* los disfrute. (Lo mismo podría decirse de juegos AAA, por supuesto).

Los juegos experimentales son un caso aún más claro de esto: quienes los disfrutan sinceramente, quienes los prefieren por sobre otros tipos de juegos, en general forman parte de la comunidad de juegos experimentales o están interesados en cuestiones afines a las que interesan a los desarrolladores experimentales (vanguardias artísticas, análisis conceptual, investigación, etcétera). Las expectativas en cuanto a estética, rareza del *gameplay* y grado de intuitividad de los controles y de los objetivos de juego, por nombrar algunos aspectos, son radicalmente distintas a las expectativas que tiene una audiencia más bien familiarizada con juegos AAA.

Probemos estas ideas con nuestra experiencia, jugando a estos dos juegos experimentales del brillante creador independiente Cactus (www.cactusquid.com). ¿Qué sucede con nuestras expectativas?



Norrland & Gamma 4, por Cactus. Video y descargas en:

<http://cactusquid.blogspot.com/2010/08/norrland-gamma4-separate-downloads.html>

Crear juegos para las audiencias 1 y 2 (Yo, Nosotros) puede resultar muy gratificante, porque comparte nuestros gustos y desean vivir experiencias similares a las nuestras. Es fácil olvidarse de que hay grados de diferencia entre las expectativas de otros y las propias, pero también es cierto que algunas veces estas diferencias tienden a borrarse por la proximidad. Esto cambia mucho con los tipos de jugadores 3 y 4 (Ustedes, Ellos). Los jugadores que no son como nosotros juegan a juegos que no son como los nuestros.

Eso no significa que esta situación no vaya a darse. No es inusual que un diseñador que prefiere los juegos FPS realistas se encuentre en algún momento con la obligación o la necesidad de diseñar un juego sobre el cuidado de tiernas mascotas, o al revés.

Es aún más extremo diseñar juegos para personas que ni siquiera llegamos a imaginar, por ejemplo porque carecemos de experiencia directa con esa clase de gente. Tal vez el caso más común de esto sea el de un diseñador aún joven, sin hijos ni contacto directo con niños, que deba diseñar un juego infantil. ¿Sabe realmente qué habilidades, gustos, preferencias, posee un niño de 7 años? ¿O tiende a hacerse una idea vaga, tal vez errada y repleta de suposiciones y lugares comunes que no son reales?

6. Definiendo una audiencia

A esta altura, es necesario que encontremos algunas estrategias básicas para pensar en cómo definir una audiencia. Por más que querramos que todo el mundo juegue a nuestro juego, *todo el mundo* no es una audiencia que nos sirva para pensar ni para diseñar una experiencia específica. Y esta especificidad es lo que lleva a ciertas audiencias a elegir y disfrutar un juego. Para que haya una audiencia, para que podamos definirla, debemos dejar algo afuera.

Una audiencia se define por sus diferencias con otras audiencias

Es en base a esas diferencias que vamos a diseñar experiencias acordes a cada *target*. Entonces, vayamos más allá con nuestras preguntas: ¿En qué consisten estas diferencias entre audiencias?

Podríamos describir a cualquier audiencia de jugadores bastante bien a través de la siguiente ficha:

Edad: _____

Género: _____

Plataforma(s): _____

Estas fichas se completan con rangos. Por ejemplo, ¿qué experiencias de juegos suele preferir cada una de las siguientes audiencias? ¿Son siempre las mismas?

Edad: _____ 15-18 años

Género: _____ Femenino

Plataforma(s): _____ Celulares, Facebook, Wii

Edad: _____ 3-5 años

Género: _____ Ambos

Plataforma(s): _____ PC

Edad: _____ 16-25 años

Género: _____ Masculino

Plataforma(s): _____ Xbox 360, PS3

Edad: _____ 16-25 años

Género: _____ Masculino

Plataforma(s): _____ PC (Redes sociales), iPhone, Wii

Como herramienta básica, resulta útil porque nos ayuda a ver algunas diferencias iniciales, inferir varias cuestiones de esos datos básicos, e incluso aplicar “ingeniería invertida” (*reverse engineering*) para ciertos problemas: Para el juego que ya planteamos antes sobre un astronauta náufrago en un planeta perdido, ¿cuál de estas audiencias sería más probable que constituya nuestra audiencia núcleo? ¿Hacia cuál orientaríamos principalmente el *gameplay*? ¿Qué tipo de *gameplay* resultaría al elegir otras audiencias?

El *gameplay* es justamente una de las pistas más importantes que nos da el definir una audiencia *específica*. Cuando se incluye “a todo el mundo” como audiencia, nos estamos perdiendo la posibilidad de generar una experiencia puntual, dirigida, efectiva la cambiamos por una experiencia generalizada.

Existen otros datos que podríamos agregar, aunque son datos menos “duros” que los anteriores:

Necesidades puntuales del jugador: _____

Contexto de uso: _____

Aspiraciones (de un tipo de audiencia): _____

Pueden ser muchas, y es necesario usar un buen criterio para identificar cuáles son útiles y cuáles no, y cómo “leerlas”.

Un caso general podría ser que tuviésemos que hacer un juego para usuarios de iPhone. Más allá de una plataforma, este dispositivo tiene un contexto propio (su uso móvil y casual en lugares públicos, por ejemplo), necesidades puntuales (más allá de los controles, que serían parte de lo que incluiríamos en “plataforma(s)”, puede ser deseable un juego de niveles breves, guardar con frecuencia, reglas de juego minimalistas para empezar rápido, o la capacidad de personalizar el juego) y, sobre todo, una serie de aspiraciones (que pueden estar relacionadas con una estética minimalista y muy cuidada que consumen los usuarios de productos Apple).

Tal vez podamos refinar nuestra audiencia encontrando subgrupos aún más relacionados entre sí. Por ejemplo, a esta audiencia hipotética, ¿qué otros intereses podrían unirlos? ¿Tienden a desear cosas en común? ¿Qué clase de gente es? ¿Practican hobbies, deportes u otras actividades en común? ¿Los une una conciencia compartida sobre temas sociales, sobre el mundo del espectáculo o sobre ser parte de un mismo país? Cualquier dato sobre una audiencia nos sirve (o debería servirnos) para entender la experiencia que hace *diferente y específica* a una audiencia, para ajustar el *gameplay* hacia ella.

En nuestro próximo level, y ahora que contamos con un conocimiento básico sobre lo que es y constituye una audiencia, continuaremos profundizando sobre este tema para explorar cómo juegan distintos jugadores.