El reto

Trasladar la experiencia de vuelo a un navegador web con un alto nivel de realismo. Para ello usamos la tecnología *webGL*.

Conseguir nubes realistas y un color de cielo especial era esencial. Fueron también los aspectos más difíciles para dar con una solución técnica óptima.

Conseguir libertad de movimiento con diferentes cámaras ofreciendo los mejores ángulos del avión en ruta. Por ello, era imprescindible tener una solución flexible para realizar el *render* fuera cual fuera la interacción del usuario.

Capturas desde diferentes cámaras

Análisis técnico

Si quieres examinar algunas de las técnicas utilizadas, añade ?debug=true. Presionando la tecla (**o**) puedes activar el control orbital, y con la barra espaciadora serás capaz de cambiar de cámara.

Nubes

Una de nuestras primeras sorpresas fue la dificultad de procesado que implicaba dar volumen a las nubes. Para solucionarlo simulamos

nubes en 2D, basadas en partículas y siempre en posición perpendicular a la cámara.

Render de las nubes

Cielo

El cielo también representó un reto. Necesitamos recrear diferentes escenarios según la luz (amaneceres, atardeceres, mediodías, noches…). Después de testar *cubemaps* estáticos y *renders* de vídeo, encontramos un estupendo sky shader desarrollado por zz85. Simula los ciclos de día de una manera increíble. Para reforzar el *look and feel* que queríamos conseguir, coordinamos el sombreado con una textura diferente para cada momento del día (*glimmering*) adaptada para los cielos estrellados por la noche.

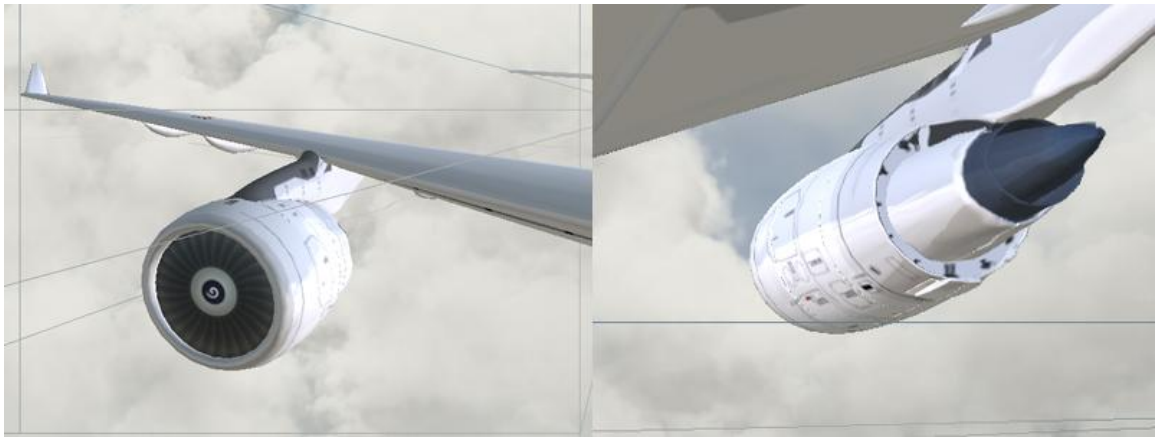
Otro elemento clave para el realismo del cielo fue representar el sol y la luna en el escenario y sincronizarlos con la hora del vuelo. Para ello usamos la librería SunCalc. Nos permitió saber la posición del sol y la luna dependiendo de la latitud y longitud en la que se encontraba el avión.

Diferentes horas del día

Post-producción y efectos especiales

Sombreados

Fueron esenciales para mejorar la estética de los *render*. Usamos un fxaa para suavizar la imagen, dado que evitamos usar el *antialiasing* en el *WebGLRenderer* para un mejor funcionamiento. Para distorsionar la imagen simulando la quema de combustible en los motores usamos una neblina centrada sólo en la parte trasera de cada turbina.


Sombreado para el combustible

Sonido 3D

Como un extra, la cámara más baja del avión permite a los usuarios aproximarse a los motores. Además de mostrar la quema del combustible, introducimos dos emisores de audio en cada uno de los motores. De esta manera se puede escuchar cada turbina –especialmente con auriculares- de manera diferente según la cámara elegida.

Animación del avión

Para que el movimiento del avión fuera suave, usamos un autonomous agent. Necesitábamos similar un puntero imaginario que el usuario manejaba, parecido a como si el usuario tirara de una cuerda. Por eso, en el caso de las cámaras usamos un comportamiento de llegada para conseguir un movimiento más natural, como si nos estuviéramos inclinando hacia adelante.

Sonidos de audio y voces

Grabamos la voz de un piloto y de una TCP (tripulante de cabina de pasajeros) para acompañar al usuario en su vuelo, como si fueran los contadores de la experiencia. También usamos sonidos típicos de aeropuerto y ruidos del interior y exterior del avión para enriquecerlo. Además, mezclamos sonidos de motores diferentes (más o menos intensos) dependiendo de la cámara elegida.

Estructura del proyecto

El desarrollo fue ágil gracias al uso de herramientas como **Grunt,** usadas para automatizar las áreas de publicación para la versión *desktop* y la móvil, y **Bower**, para manejar las librerías.

Usamos librerías generales como jQuery, TweenMax, SoundJS o PreloadJS para facilitar la programación. Y nos beneficiamos de js-signals para manejar los eventos y de handlebars para las plantillas.

En la parte de aplicación en tiempo real, un único archivo con más de 200 líneas con la librería socket.io programada en node.js nos permitió crear diferentes espacios para cada experiencia usando un único código fuente.

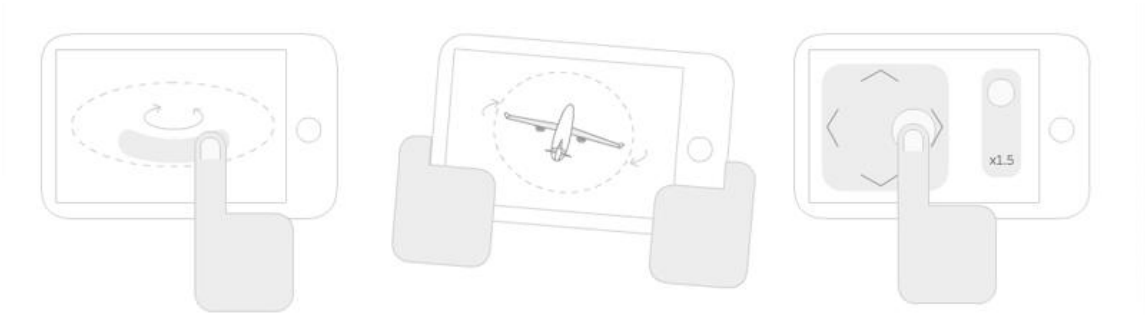
3D Three.js

Por supuesto, la base de todo fue la 3D de la librería de Javascript three.js. Además, fragmentamos la WebGLen numerosos archivos para cada clase para conseguir un proyecto muy estable.

Arte y usabilidad

Interacción del usuario

Tuvimos que imaginar las formas de interacción que tendría el usuario y que permitiríamos durante el vuelo, evitando posiciones indeseadas del avión. Al final optamos por permitir cambios de cámara y variar ligeramente la posición del avión.



Proposiciones de interacción con el avión

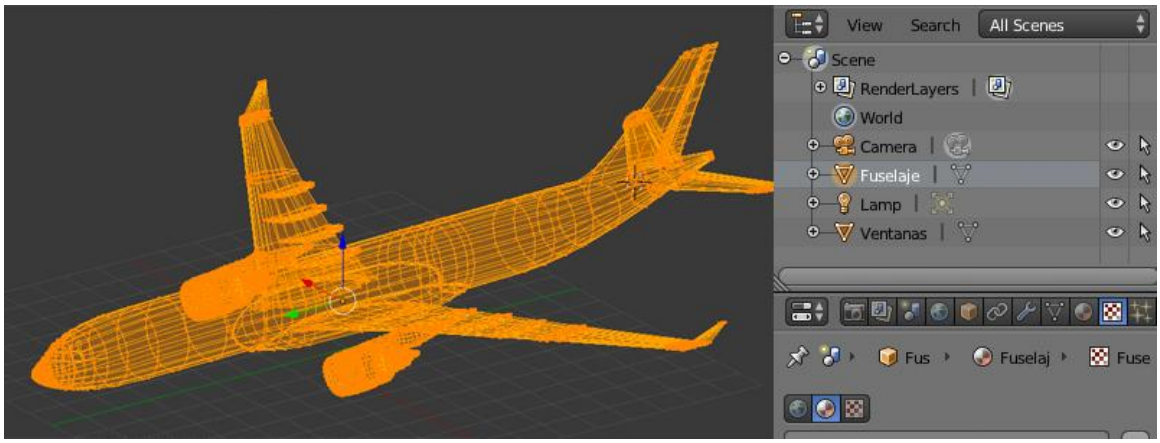
Usabilidad móvil

Debíamos crear una interfaz móvil que fuera fácil de manejar, adaptada a las limitaciones técnicas –especialmente en el caso de iOS, ya que estos dispositivos no permiten pantallas completas como sí lo hacen los Android-.

Evolución del mobile UI

Menos es más cuando hablamos de polígonos

Usamos Blender para trabajar los visuales en 3D. Nuestro punto de partida era el avión *renderizado* que usamos en nuestros spots de televisión, pero trabajando con 195.743 caras y millones de mallas no era viable para mantener las texturas originales de UV; retopologizamos la malla hasta conseguir en torno a 10.000 caras, una única malla y 2 texturas, una para el fuselaje y otra para las ventanas, que nos permitiera iluminar de manera independiente.

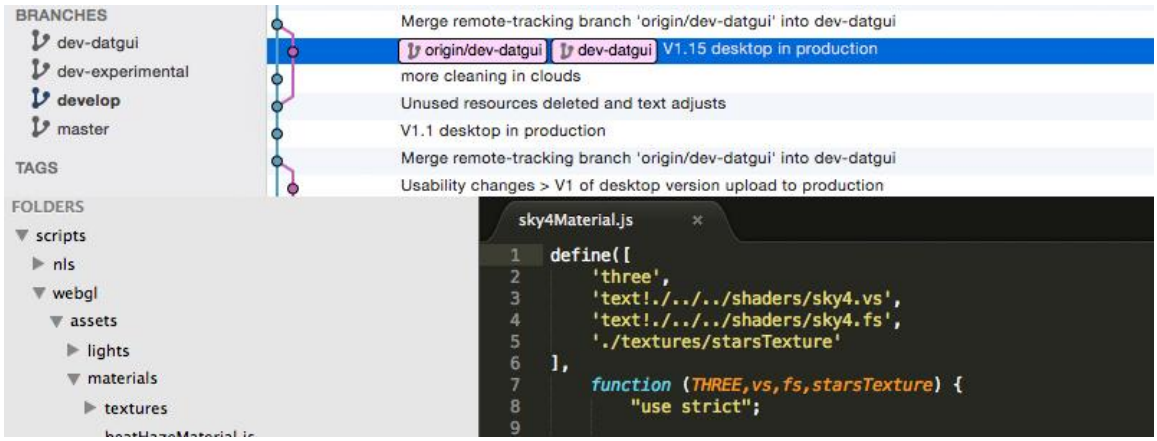


Retopologized A330

Producción

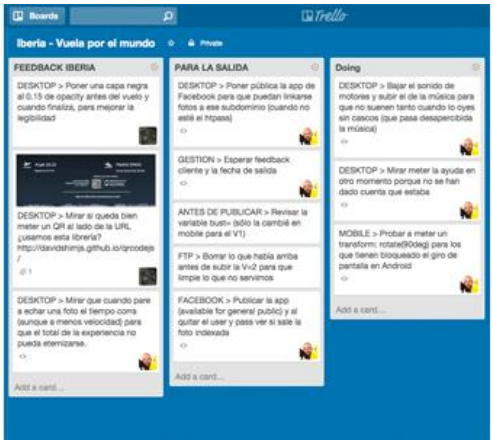
Otra de las herramientas clave fue **Git**, con la que conseguimos trabajar la rama de desarrollo teniendo una rama "máster" sólo para lo que debía ser subido a producción. También gracias a Grunt

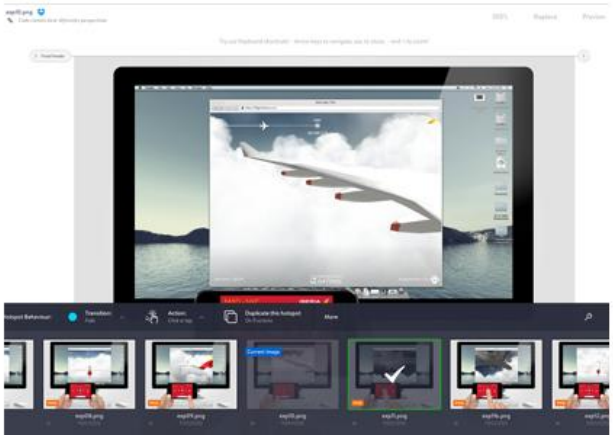
conseguimos reducir el peso de la producción y comprimir el archivo *javascript* con más de 100 clases para escritorio y móvil.



Git and Sublime Text

Para coordinar, Trello fue básica para asignar tareas y prioridades. Otro gran descubrimiento al prototipar y compartir fue Marvel's service. Nos permitió compartir bocetos interactivos para poder entender en qué estábamos trabajando.




Trello and Marvel.app

Resultado final

El resultado puede comprobarse desde flight.iberia.com. Puedes disfrutar de la experiencia desde tu ordenador usando un móvil o tablet como mando.

Para hacerlo más social, tiene la funcionalidad de capturar imágenes del vuelo y descargarlas, o compartir en las redes sociales.

Ejemplos de capturas

¡Esperamos que te guste!