

Memoria Trabajo de Fin de Grado

Creación de un juego para “Smartphone”, Iride

Autor: Oriol Núñez Vilaseca

Director: Antoni Castejón García

Titulación: Grado en multimedia.

Índice

Contenido

CREACIÓN DE UN JUEGO PARA “SMARTPHONE”, IRIDE	1
1. INTRODUCCIÓN.	4
1.1 ANÁLISIS DE LA COMPETENCIA.....	4
1.1.1 <i>Otros juegos</i>	4
1.1.2 <i>Puntos en común</i>	6
1.1.3 <i>Valor añadido</i>	7
1.1.4 <i>Modelo económico y objetivos</i>	7
1.2 PÚBLICO OBJETIVO	7
1.3 METODOLOGÍA DE TRABAJO.	8
2. FASE DE DISEÑO.	10
2.1 GAME DESIGN	10
2.2 DISEÑO DE MENÚS	11
2.2.1 <i>Navegación</i>	11
2.2.2 <i>Diseño gráfico</i>	12
2.3 DISEÑO DE PERSONAJES.....	18
2.3.1 <i>Idea y bocetos</i>	18
2.3.2 <i>Modelado</i>	18
2.3.3 <i>Texturizado</i>	19
2.3.4 <i>Rig y Animación</i>	20
2.4 DISEÑO DE LONGBOARDS.....	22
2.5 DISEÑO DE ESCENARIOS.....	23
2.6 DISEÑO DE ADVERSARIOS.....	24
3. USABILIDAD.....	26
3.1 PRIMERA PRUEBA.....	26
3.1 SEGUNDA PRUEBA.....	31
4. DESARROLLO.....	33
3.1 ¿POR QUÉ UNITY?	33
3.2 ASPECTOS TÉCNICOS DE PROGRAMACIÓN	34
3.2.1 <i>Clases del menú</i> :.....	34
3.2.1.1 <i>Esquema de clases del menú</i> :.....	35
3.2.2 <i>Clases del juego</i> :	36
3.3 TEST EN DISPOSITIVOS MÓVILES	39

5. CONCLUSIONES	40
6. BIBLIOGRAFÍA.....	41
7. ANEXO	42
7.1 GAME DESIGN.....	42
7.2 PROJECT.....	50
7.2 PROJECT.....	51

1. Introducción.

Este proyecto consiste en la creación de un videojuego para dispositivos móviles.

En el proyecto se han realizado tanto los procesos de concepción de la idea, como el diseño y la beta del juego.

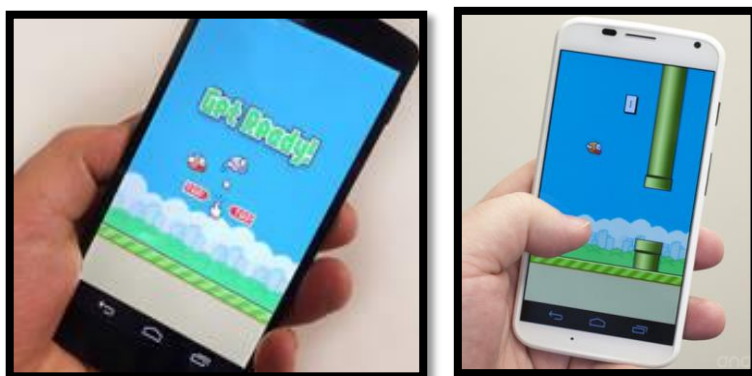
1.1 Análisis de la competencia

Las tiendas de aplicaciones que tenemos disponibles hoy en día en nuestros smartphones, principalmente “AppStore” en el caso de Apple o “PlayStore” en el caso de Android, están llenas de juegos que se suelen llamar “free time-wasters”. Los podríamos definir como juegos para el tiempo muerto, es decir juegos pensados para que sean usados en períodos de tiempo en que las personas están esperando, en el metro, en una cola, etc. Por lo tanto tienen un planteamiento muy diferente al de los videojuegos tradicionales, para ordenadores o consolas, los cuales están pensados para pasar tiempo jugando con ellos.

Son juegos sencillos, a los que se juega fácilmente, suelen ser repetitivos y no esconden un guion literario, ni tratan de dar mensajes o contar historias. Sus principales virtudes son proporcionar diversión rápidamente, ser sencillos sin estar sobrecargados pero estar muy bien elaborados y también es muy importante que sean adictivos.

1.1.1 Otros juegos

Flappy Bird



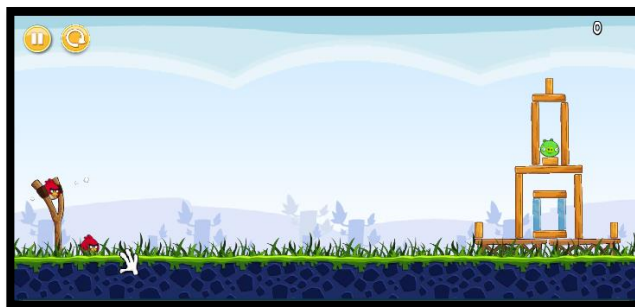
En la parte superior vemos un popular juego que fue un fenómeno y salió en todos los telenoticias llamado “FlappyBird”. Es un juego sin fin, si tocamos la pantalla movemos el pájaro hacia arriba y sino cae, tenemos que esquivar los obstáculos sin tocar el suelo y conseguir aguantar el mayor tiempo posible. El juego, a pesar de tener gráficos muy sencillos sin un gran nivel de detalle, fue capaz de generar una adicción muy grande y eso fue lo que la hizo triunfar. Además jugó un papel clave el boca a boca entre la gente que se quedaba enganchada y animaba a sus amigos a descargar el juego. Es un claro ejemplo de “free time waster” claro, simple y adictivo.

Angry birds

Ahora vamos a ver un juego que ya lleva mucho tiempo entre nosotros. Va más allá que “*flappyBird*” con su trama principal, liberar los huevos que han robado los cerdos salvajes. Armado con tus compañeros, los pájaros enfadados, tienes que destruir la fortaleza de los cerdos usando los pájaros como proyectiles. Éste juego está organizado por niveles, conforme los superemos, desbloquearemos otros niveles cada vez de mayor dificultad. En la primera imagen vemos el primer nivel, muy fácil, ya que normalmente con un buen golpe puedes liberar los huevos. En cambio en la segunda vemos un nivel mucho más avanzado, el cual para superarlo tendremos que definir una estrategia y atacar por distintos lados para conseguir la liberación. Los controles del juego son simples arrastrar el tirachinas con el dedo para definir la fuerza y el ángulo de salida. Hasta aquí podemos ver puntos en común con el juego anterior, pero hay una diferencia muy grande. En el primer juego destacaban las personas con mayor habilidad, en este juego en cambio destacarán los usuarios que desarrollen una mejor estrategia utilizando el menor número de pájaros para conseguir el mayor número de puntos.

Además este juego destaca por una gráfica simple pero muy completa y con detalles, también goza de unas dinámicas de fuerzas muy realistas que hace que los choques que realices sean muy reales.

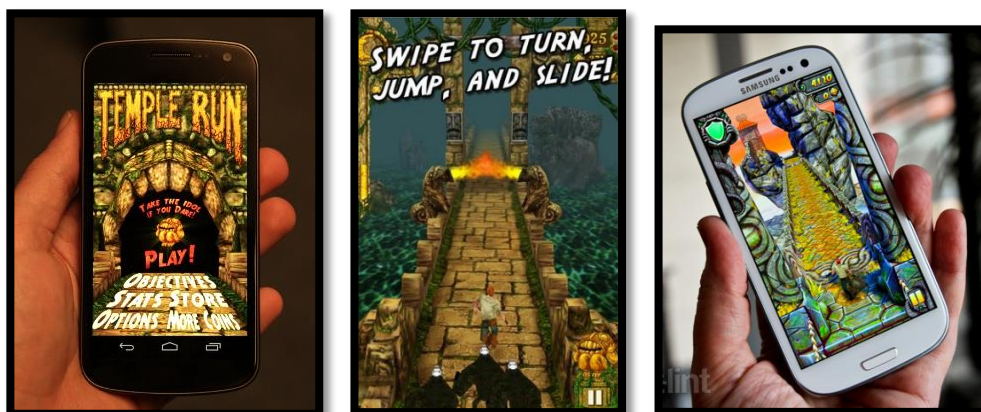
Gracias al éxito de la primera versión la empresa ha podido crecer mucho, además de realizar un sinnúmero de versiones del juego, como por ejemplo una en el espacio, con físicas adaptadas, u otra para promocionar una película de animación sobre pájaros, la película “Rio”.



Temple Run

Este juego consiste en escapar del laberinto sin que nos atrape el monstruo que nos persigue. Este juego destaca entre los otros dos por tener más opciones de control disponibles, ya que tendremos que usar tanto el acelerómetro como la pantalla táctil. Mediante la inclinación moveremos al corredor de derecha a izquierda por el camino y con los gestos en la pantalla lo haremos saltar, deslizarse por el suelo o girar a izquierda y derecha en las esquinas.

Este juego también enganchó mucha gente ya que ofrece la posibilidad de utilizar los puntos que se recojan para poder comprar personajes basados en personas reales, como por ejemplo Usain Bolt o también escenarios o poderes especiales para facilitar el juego y hacer que el usuario aumente el tiempo de partida, con el cual se hacen ránquines ya que el objetivo es conseguir un tiempo mayor al de nuestros amigos.



1.1.2 Puntos en común

- Curva de aprendizaje muy pequeña, se aprende a jugar rápido.
- Son adictivos debido a la motivación por superar a nuestros amigos y conocidos. Con mucha presencia en redes sociales.
- No tienen un argumento fuerte.
- Las partidas o niveles son de corta duración, para así asegurar que el usuario pueda hacer una partida en sus 5 minutos libres sin tener que dejarla a medias.
- Gráficos simples para que puedan ser ejecutados en un mayor número de dispositivos con menos recursos.
- Juegos gratuitos. Se financian con compras dentro de la aplicación y/o publicidad.

1.1.3 Valor añadido.

El valor añadido del juego que se realiza en este proyecto es su temática, el “longboard”. Es un deporte que está de moda hoy en día y hay cada vez más usuarios y no hay una gran variedad de juegos del deporte disponibles para dispositivos móviles, además los que hay no son de los denominados “FreeTimeWasters”.

1.1.4 Modelo económico y objetivos.

El modelo económico que se plantea con este juego es el de la gratuidad para los usuarios. Los objetivos que tengo con este desarrollo son didácticos, aprender a crear un juego. No se crea con el objetivo de venderse, sino con el objetivo de crear un juego el cual pueda estar en la tienda de aplicaciones y destacar en ella como juego, puesto que el beneficio de que me conozcan considero que es más valioso que las retribuciones económicas que me pudiera aportar. Por tanto el hecho de que no tenga ningún coste para el usuario será un punto clave para lograr un mayor número de jugadores.

Además también tenemos que tener en cuenta que es un segmento con muchos competidores cuyos juegos son gratuitos, lo único que ofrecen es la posibilidad de pequeñas mejoras por unos céntimos de euro, las llamadas compras “in App”, dentro de la aplicación o también juegos en los cuales hay espacio a anunciantes.

1.2 Público objetivo

El juego va dirigido al público joven que está interesado en el propio deporte “longboard”. Actualmente está creciendo el número de usuarios entre la gente de 15 a 25 años aunque hay usuarios de múltiples edades repartidos por todo el mundo. Sin embargo, el target del juego se centrará entre los 16 y 24. Para conseguir esto el diseño gráfico del juego tiene como objetivo mantener la simplicidad que necesita el juego pero sin ser infantil o demasiado sencillo. Un problema con el que tendremos que lidiar es que el mismo target suele ser usuario de consolas de última generación, por lo que están acostumbrados a jugar a juegos con entornos 3d muy elaborados, como por ejemplo los del juego “Watch Dogs”, en imagen.



Debido a que el juego es para dispositivos móviles y estos no tienen suficiente potencia como para soportar gráficos tan realistas como los que vemos en el ejemplo anterior, la gráfica tendrá que resultar graciosa para el usuario para que le guste, ya que no podrá tener un nivel de detalle tan alto.

1.3 Metodología de trabajo.

En un proyecto de estas características, la mejor metodología de trabajo es tener un equipo de personas que se dediquen al proyecto y utilizar la ayuda de algún software de gestión de procesos y / o de gestión de tareas. De este modo se consigue coordinar a un equipo con la seguridad de recoger todas las tareas con sus fechas y demás detalles como tiempo empleado. Si somos ordenados podemos recoger todos estos datos con algún programa como Microsoft Project, con el que se generan árboles de tareas con fechas de entrega, previsión de horas invertidas y todas las herramientas para gravar los movimientos hechos y los que quedan por hacer. Además también se pueden usar gestores de tareas para ir repartiendo las tareas entre los miembros del equipo.

Sin embargo, en mi caso, realicé el proyecto solo, por lo que opté por una estrategia un poco distinta para la organización. Primero establecí un número de horas semanalmente para dedicar al proyecto, aun sabiendo que el proceso se podía alargar mucho. Para ello he utilizado la herramienta Projectc dividiendo todo el proyecto en tareas y sus respectivas subtareas. Esto lo he complementado con mi gestor de tareas personal, "Wunderlist", ahí cada semana me anotaba pequeñas tareas para cumplir objetivos semana a semana. He utilizado esto para substituir la gestión del tiempo del proyecto de Project. Al no tener una dedicación a tiempo completa con el proyecto, usar la gestión del calendario con este proyecto no hubiera sido un caso real, ya que las tareas se han ido realizando en paralelo junto a otras tareas. Si hubiera utilizado la herramienta para esto, los datos obtenidos serían tareas extendidas en períodos de tiempos muy largos que no creo que aporten nada al proyecto. Al contrario que utilizar la herramienta como mapa de tareas, que si resulta muy útil para cumplir con éxito el proyecto sin dejar cabos sueltos. Además nos permite ver en que parte del proceso estamos. Y la posibilidad de dividir las tareas en subtareas si estas son complejas o se van complicando durante el proceso.

En las siguientes imágenes podemos ver el árbol en el que se ha dividido el proyecto.

<ul style="list-style-type: none"> ▸ Game Design <ul style="list-style-type: none"> <i>Idea inicial</i> diseño jugabilidad Relación usuario juego Relación juegos usuario 	<ul style="list-style-type: none"> ▸ Pruebas de usabilidad <ul style="list-style-type: none"> Prueba de navegación a partir de las maquetas Detección de errores de diseño
<ul style="list-style-type: none"> ▸ Creación de arte <ul style="list-style-type: none"> ▸ Diseño "look and feel" <ul style="list-style-type: none"> Bocetos Diseño maqueta final Corrección de errores de diseño 	<ul style="list-style-type: none"> ▸ Proceso de desarrollo <ul style="list-style-type: none"> ▸ Desarrollo de menús <ul style="list-style-type: none"> ▸ Vistas <ul style="list-style-type: none"> Importar arte a Unity ▸ Controlador <ul style="list-style-type: none"> Creación de funciones principales Funciones específicas de cambio de escena
<ul style="list-style-type: none"> ▸ Diseño Personajes <ul style="list-style-type: none"> Bocetos Diseño maqueta final 	<ul style="list-style-type: none"> ▸ Desarrollo juego <ul style="list-style-type: none"> ▸ Vistas <ul style="list-style-type: none"> Importar arte a Unity ▸ Controlador <ul style="list-style-type: none"> Controlador tiempo y partida Controlador controles Controlador escenario Controlador objetos Controlador personaje
<ul style="list-style-type: none"> ▸ Diseño Escenarios <ul style="list-style-type: none"> Bocetos Diseño maqueta final 	<ul style="list-style-type: none"> ▸ Test en dispositivos
<ul style="list-style-type: none"> ▸ Pruebas de usabilidad <ul style="list-style-type: none"> Prueba de navegación a partir de las maquetas Detección de errores de diseño 	<ul style="list-style-type: none"> ▸ Test en usuarios finales

En el apartado de anexos se puede encontrar todo el documento del Project en pdf donde se puede ver con más detalle la organización de las tareas.

2. Fase de diseño.

El diseño del juego se ha realizado en diferentes etapas. La primera, llamada "Game Design", comprende todos los datos referentes a la idea del juego, por qué y cómo será el funcionamiento de los elementos dentro del juego.

Definimos todos los aspectos teóricos del juego, nos tiene que servir para que los demás entiendan el juego. En él no puede haber ningún cabo suelto, ya que cuando vayamos a pasar de la idea al juego ese será el guion que sigan los artistas que realicen la gráfica y los programadores que lo hagan funcionar.

En cuanto al diseño gráfico del propio juego se realizó después de tener el concepto del juego claro. La parte gráfica se ha dividido en 4 tareas.

- Diseño de menús y navegación.
- Diseño de personajes.
- Diseño de longboards.
- Diseño de escenarios.

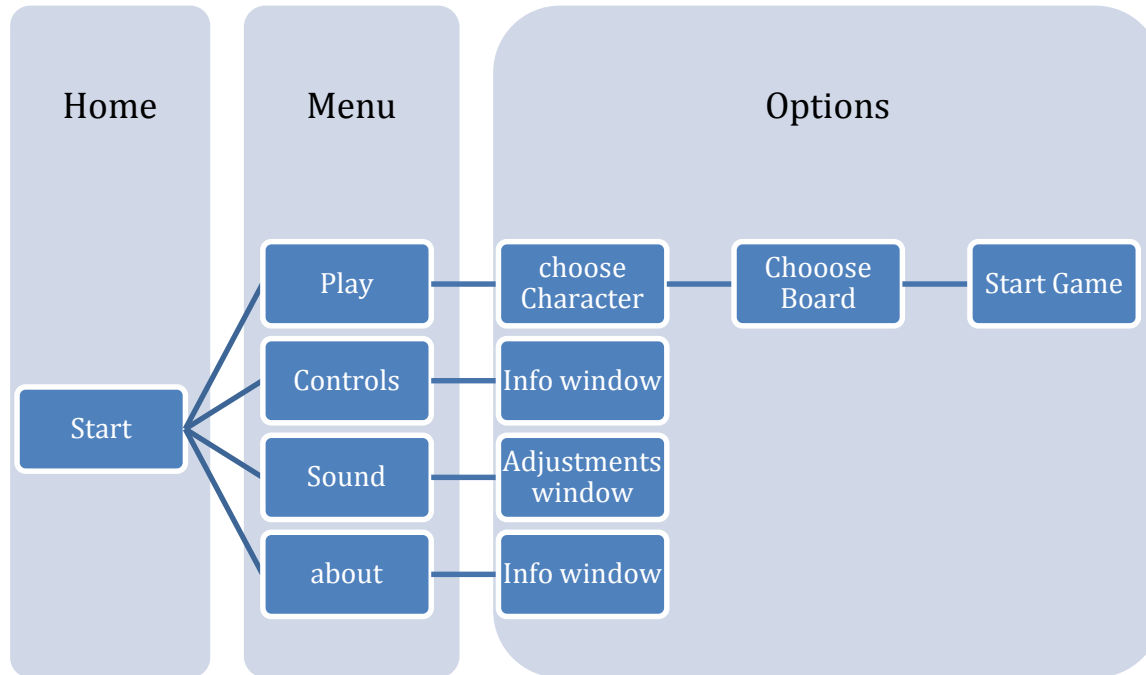
En esta fase del desarrollo del juego es importante tener en mente la tecnología que se va a usar para crear el juego en sí. En nuestro caso vamos a usar Unity, un motor gráfico para juegos. Para buscar una mayor eficiencia en el desarrollo del juego todos los materiales que se han ido creando se han creado ya en 3D para que después se puedan usar en Unity sin tener que volver a generar la gráfica.

2.1 Game Design

Este documento lo podemos encontrar en el anexo. En él podemos encontrar todos los aspectos relacionados con la idea del juego

2.2 Diseño de menús

2.2.1 Navegación



Podemos ver el esquema de navegación de botones de los menús del juego.

2.2.2 Diseño gráfico.

Después de tener el mapa de funcionamiento de los menús definí estos criterios básicos que debía cumplir el juego tanto a nivel de usabilidad móvil para evitar futuros problemas como con los objetivos del tipo de juego que estoy haciendo.

- Gráfica simple, sin ser infantil ni con un nivel de detalle excelente.
- Conceptos de usabilidad:
 - Botones grandes, ya que con el dedo no se puede ser tan preciso como con un ratón.
 - Identificar en que parte del menú estamos.
 - *Menos es más.*
 - *Un buen diseño trata de simplificar y aclarar, no añadir capas de complejidad.*



Diseño de la pantalla de inicio. En esta pantalla tenemos la opción de registrarnos mediante el uso de Facebook y la opción de empezar a jugar, "Start". Esta es una de las pantallas más importantes del juego ya que es lo primero que ve el usuario al ejecutar la aplicación y por tanto si no está bien podría causar que algunos usuarios llegarán a cerrar el juego sin apenas haberlo visto. Tenemos el logo del juego en la parte superior, el patinador para que se sepa de qué va el juego, el botón "Start" en la parte central y debajo la opción de registrarnos con Facebook.

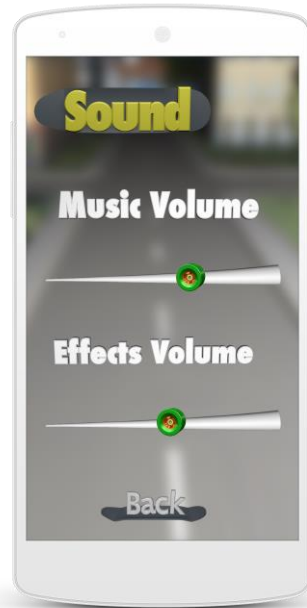


Este es el menú principal del juego. Tenemos las principales opciones comunes en todos los juegos. Usamos el logo en la parte superior para saber que estamos en la aplicación, letras grandes que son los botones, hay que tocar las palabras. También se ha añadido la rueda al lado del botón “Play” para llamar la atención del usuario y este vaya directamente a jugar.



En esta pantalla podemos ver los 5 controles que necesitamos conocer para poder jugar. Además, la primera vez que juguemos tendremos una pequeña introducción donde se presentarán los controles y cuando usarlos. En la parte superior el logo

cambia y se utiliza un marcador para indicar la opción del menú en la que se ha entrado. También hay la opción atrás en la parte inferior.



En esta pantalla podemos controlar el volumen del juego. Tenemos dos sliders que al moverlos se ajustará el volumen al nivel deseado. Con el ancho de la barra se simboliza el lado en el que el volumen será mayor. Por lo tanto, arrastrar la rueda a la izquierda reducirá el volumen mientras que llevarla a la derecha incrementará el volumen.



En esta pantalla podemos ver información sobre el proyecto, el autor, etc.



En esta pantalla elegimos el patinador que queremos. Veremos a un personaje en el medio, que estará dando vueltas sobre sí mismo, para que se vea desde todos los ángulos. Usando las flechas que podemos ver, una a cada lado del patinador,

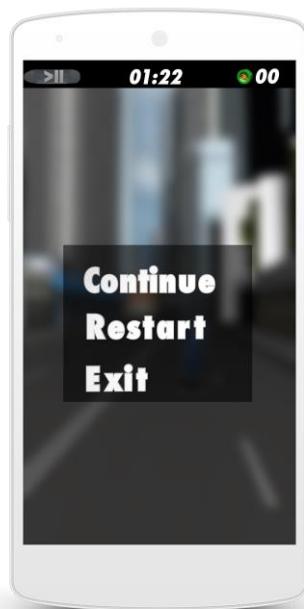
seleccionaremos un muñeco u otro. Finalmente para avanzar tendremos que pulsar el botón "Next".



En esta pantalla elegimos el "longboard" que queremos. Veremos un "longboard" en el medio, que estará dando vueltas sobre sí mismo, para que se vea desde todos los ángulos. Usando las flechas que podemos ver, una a cada lado, seleccionaremos uno u otro. Finalmente para empezar tendremos que pulsar el botón "Play".



En esta pantalla vemos ya como será el “Gameplay”, puesto que esta pantalla muestra como sería el juego cuando se esté ejecutando.



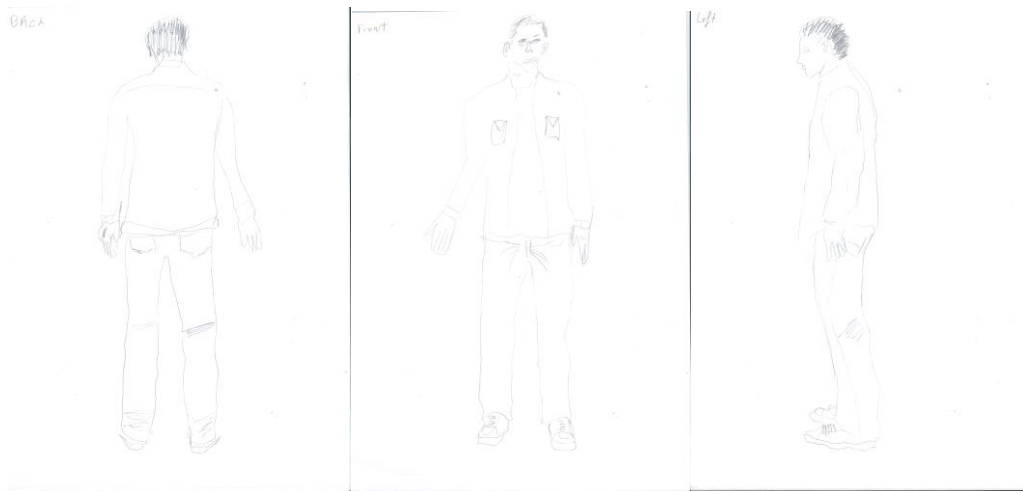
En esta pantalla vemos como será el menú que se verá al pausar el juego. En el fondo tendríamos la imagen del juego congelada y desenfocada y encima nos quedaría el menú con las opciones.

2.3 Diseño de personajes

2.3.1 Idea y bocetos.

El primer paso para decidir como quería hacer el personaje fue fijarme en otros juegos, fijarme en la cantidad de detalles que tienen los modelos, cantidad de colores, sus formas y sus estilos y realizar a partir de ahí unos bocetos a mano alzada.

A pesar de que en el propio juego el personaje solo se ve de espaldas, decidí que era mejor realizar todo el modelado completo, ya que además de aportar más realismo, me permitiría la utilización de los modelos para los menús.

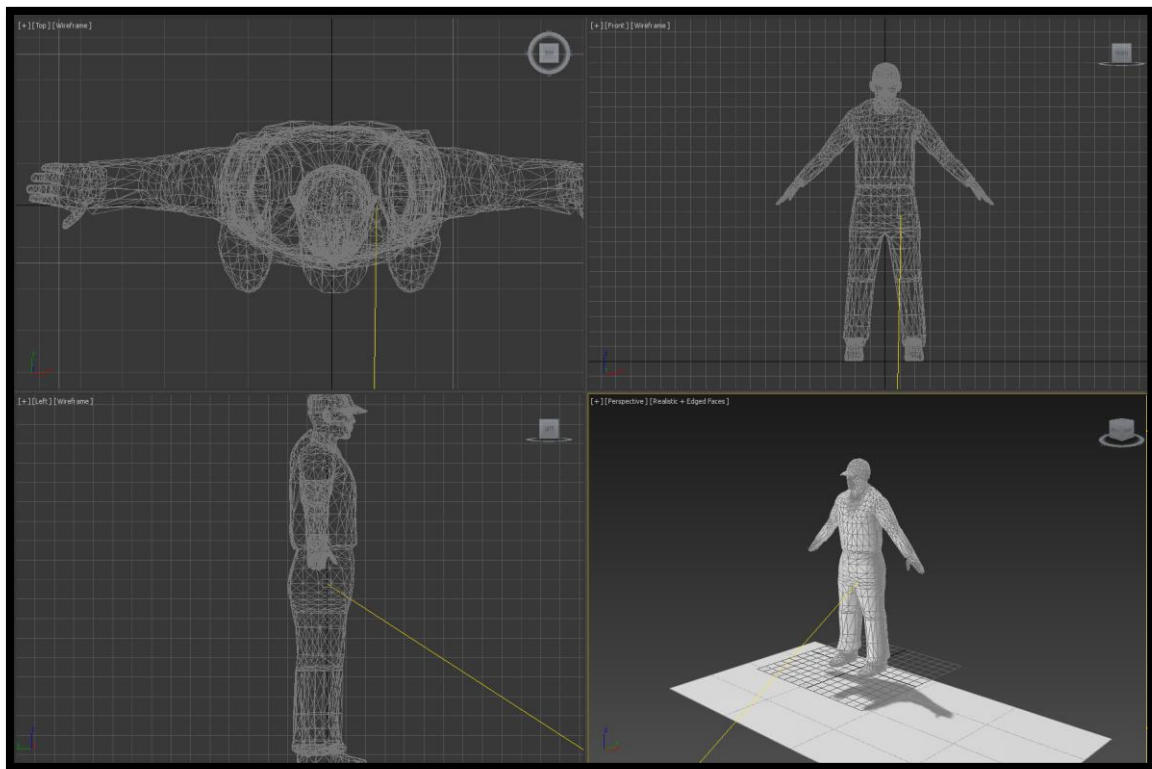


2.3.2 Modelado.

En general para el proceso de creación de los modelos he utilizado tres programas, 3dsMax, Mudbox y cinema4d.

En concreto 3ds max lo he utilizado para generar la forma humana del modelo, la malla. Para ello he utilizado un modo tradicional, usando capturas de pantalla del personaje de un juego como fondo en el editor para usarlas de guía de modelado. De esta forma tuve una referencia en cuanto a formas y proporciones. Hubiese podido utilizar un fotografía o incluso mis bocetos, pero creo que la mejor referencia que podía tener era otro modelado 3d, ya que ni era tan detallado como una fotografía, ni tan poco detallado como los bocetos.

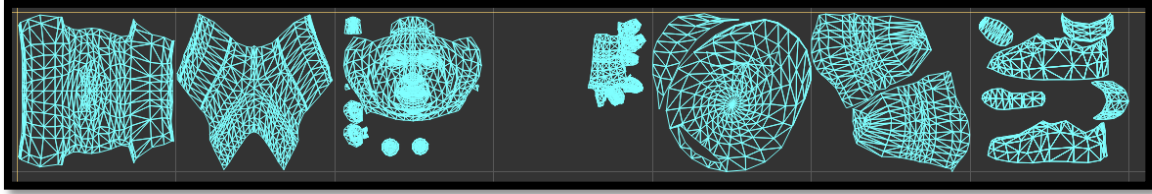
Aquí podemos ver en imagen la malla final del modelo antes de pasar al siguiente punto, la texturización.



2.3.3 Texturizado.

En este proceso generé los materiales para el modelo. Lo hice con un programa llamado Mudbox. Con este programa se han hecho dos procesos, la generación de mapas bump y la creación de las texturas.

Los mapas bump son mapas que sirven para simular relieves que no existen en la malla, de este modo las mallas pueden ser menos densas y menos pesadas, ya que si hay muchos detalles no serían aptas para juegos y menos para móviles. Los mapas en si se guardan en formato de imagen y contienen información para los programas. Una vez añadidos estos detalles al modelo se procedió a dar color y textura a la malla con el método "Unwarp". Este método crea imágenes de la malla en un plano 2d. Se seleccionan las partes de la malla que comparten textura hasta que se ha dividido toda la malla en distintas partes, tronco, pantalones, zapatos, cara, gorra, etc. Entonces el programa transformará la superficie 3d de la malla a 2d en imagen. Con esto definido podemos empezar a usar las herramientas de pintura del programa. Tenemos una serie de pinceles y demás funciones para dar color a la mala. Entonces con "photoshop" u otro programa de edición de imagen se modifica la textura para darle los acabados que le queramos dar. Finalmente el programa aplica la textura del mapa 2d a la malla 3d para que se pueda llevar a otros programas y seguir con el proceso.



En esta imagen podemos ver el resultado de “Mudbox” de la malla extendida, Se pueden observar la cara, los pies, las manos, el torso, los brazos y las piernas.



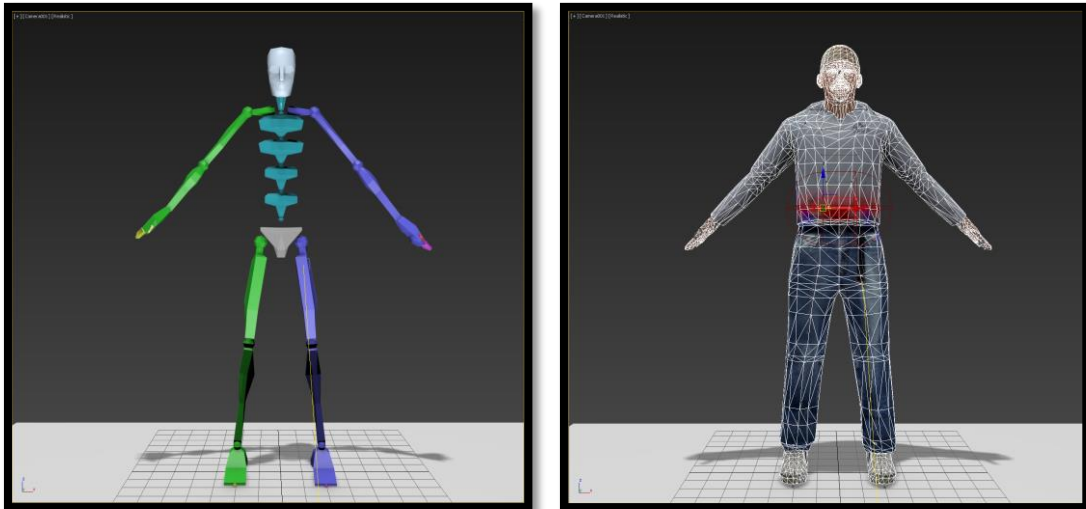
En esta imagen vemos como quedaron las texturas finales.

2.3.4 Rig y Animación

El “rig” es un proceso que sirve para facilitar la animación posterior. Para que una malla deje de ser una malla y se convierta en lo que representa, un cuerpo, un objeto mecánico, un árbol, etc, esta tiene que cumplir unas propiedades físicas, moverse bajo ciertas reglas y así podrá parecer real. En nuestro caso aplicamos una estructura basada en un sistema de huesos. Estos huesos están anidados de cierto modo que al mover una mano, por herencia, se mueve el resto de elementos del brazo, si el movimiento lo requiere, como en nuestro cuerpo.

Una vez se ha adaptado el esqueleto a la forma del modelado, tenemos que vincular las partes de la malla al hueso correspondiente. Es un proceso laborioso en el cual para cada hueso hay que crear una zona de influencia con distintos niveles de influencia, para que el hueso mueva la parte de la malla correspondiente y adapte la de los lugares donde no afecta directamente o afecta menos.

En estas imágenes podemos ver a la izquierda la estructura de huesos del personaje y a la izquierda el proceso de vinculación de huesos con malla y la delimitación de deformación en la malla causada por el hueso. En este caso las partes rojas es donde el hueso tendrá mayor influencia y las azules menos.

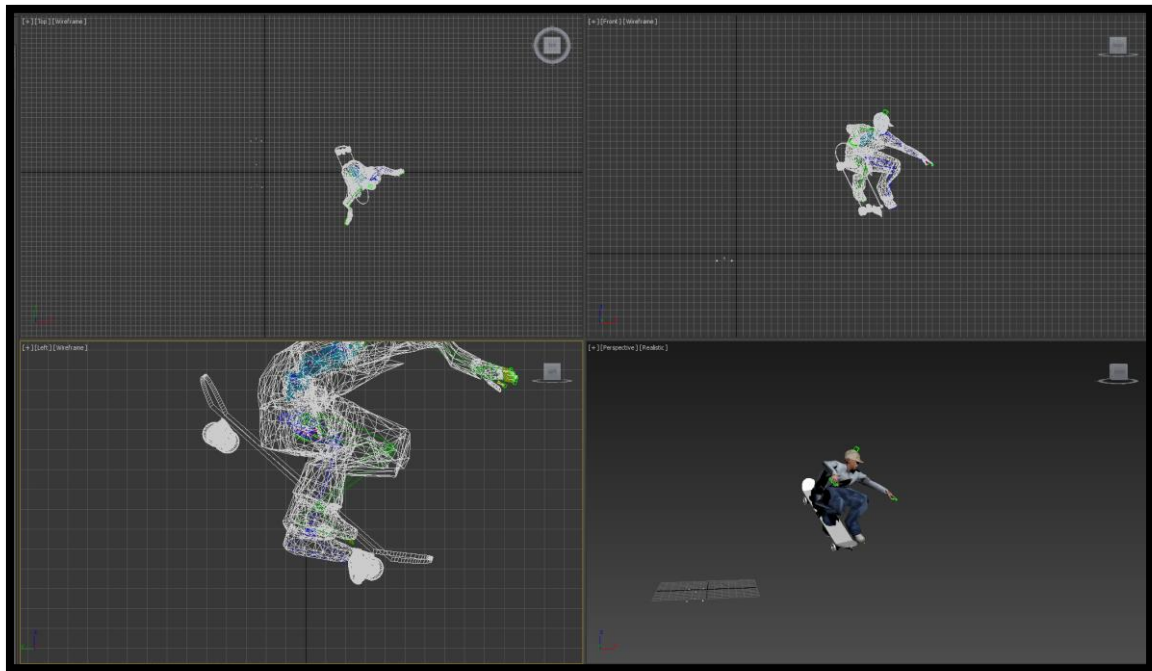


Una vez terminados estos procesos podemos empezar a animar los personajes.

En nuestra aplicación se necesitan tres secuencias de animación, la “idle” estado por defecto del personaje patinando, una que se combinaría con esta en la que el personaje se impulsa y otra para el salto. Podríamos añadir más animaciones, pero al tratarse de un juego para móviles nos obliga a simplificar aspectos para no sobrecargar el sistema y así garantizar que más dispositivos podrán ejecutar el juego.

Para este tipo de animaciones se suelen usar herramientas que permiten gravar con sensores de movimiento a deportistas y aplicarlos a los modelos 3d previamente preparados. Pero como no tenemos el material para poder capturar estos movimientos en personas reales, el proceso de animación se ha tenido que hacer moviendo los huesos imitando los movimientos de los patinadores.

En la imagen podemos ver una captura de pantalla del proceso de animación de salto.

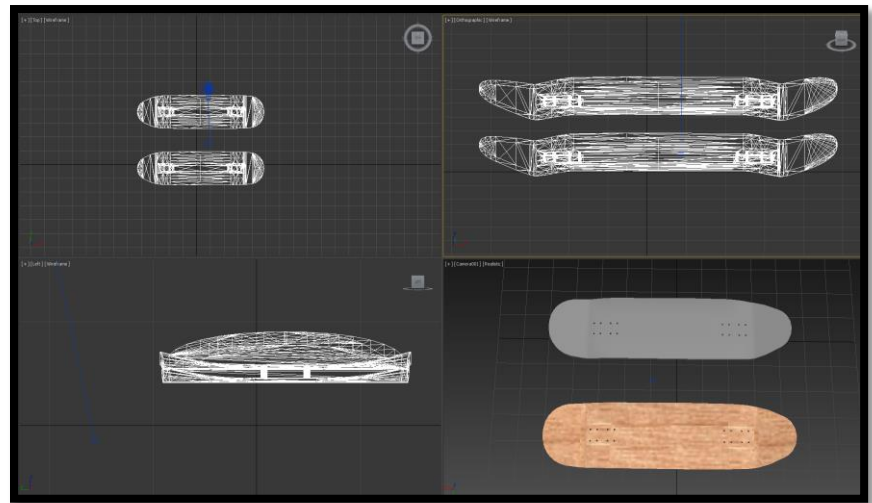
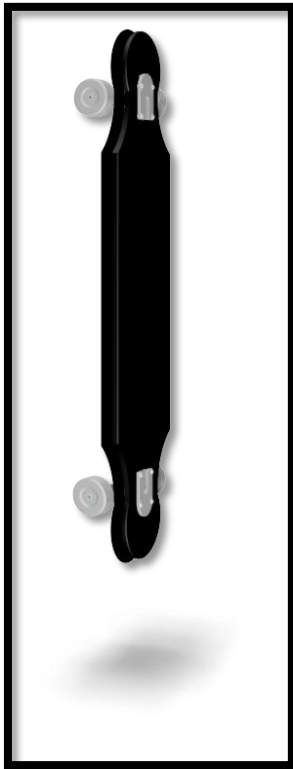


Por ahora solo se ha desarrollado un personaje, pero más adelante, aunque sea fuera del proyecto, se repetirán estos procesos para crear más personajes y añadir opciones al juego.

2.4 Diseño de longboards

Para la creación de los longboards se ha seguido un procedimiento parecido al del personaje. Se han creado en base a modelos reales a partir de imágenes de estos.

El procedimiento tubo la parte de modelado y la parte texturizado, ambas más fáciles que en el caso del cuerpo del personaje. En el caso del modelado las formas mucho más regulares y el poder tener fotos exactas de los modelos ha simplificado mucho el proceso. De igual manera el proceso de texturizado ha resultado más sencillo ya que hay menos cantidad de materiales y además, ya que estos no tendrán mucho peso visual en el juego, se ha optado por simplicarlos al máximo para reducir la carga gráfica del juego.



En las imágenes superiores podemos ver dos modelos de longboard distintos. A la izquierda tenemos el “render” de un longboard finalizado y a la derecha como podemos ver tiene un nivel de detalle menor que el modelo.

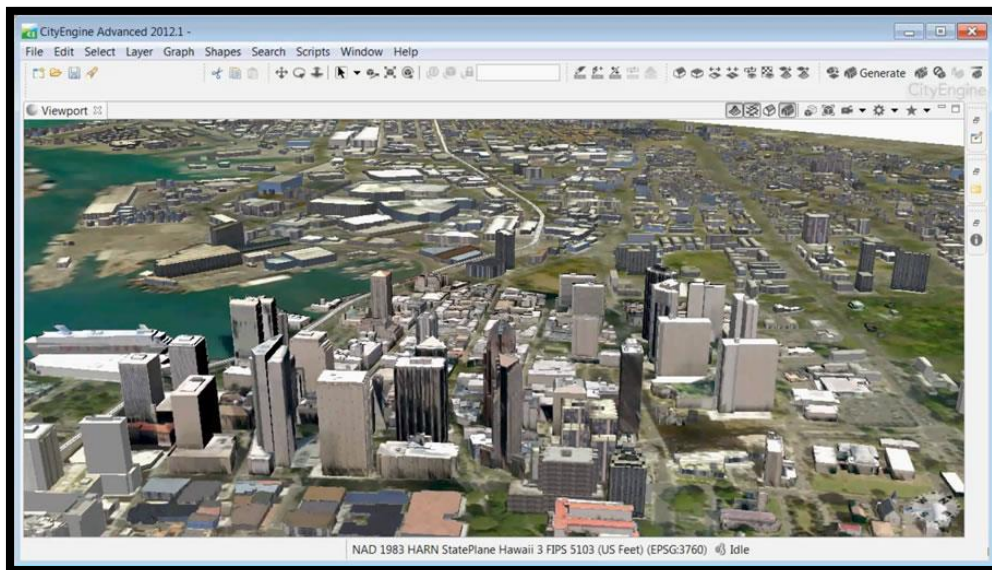
Por otro lado a la derecha tenemos una captura de pantalla donde se puede ver el proceso de creación de un segundo modelo que se he hecho.

2.5 Diseño de escenarios

El diseño de las ciudades para el juego se ha hecho mediante un programa llamado “CityEngine” este programa se utiliza para crear ciudades aleatorias a medida. Podemos definir una multitud de parámetros, desde la forma que queremos que tenga la ciudad, el tipo de edificios, la disposición de estos, la densidad de edificios, opciones en cuanto a texturas, anchuras de calles, etc.

Con el uso de este programa ganamos mucho tiempo ya que el esfuerzo que se tiene que hacer para modelar una parte de una ciudad desde cero no tiene sentido para esta aplicación.

En la imagen vemos el editor del programa, con un ejemplo de ciudad.

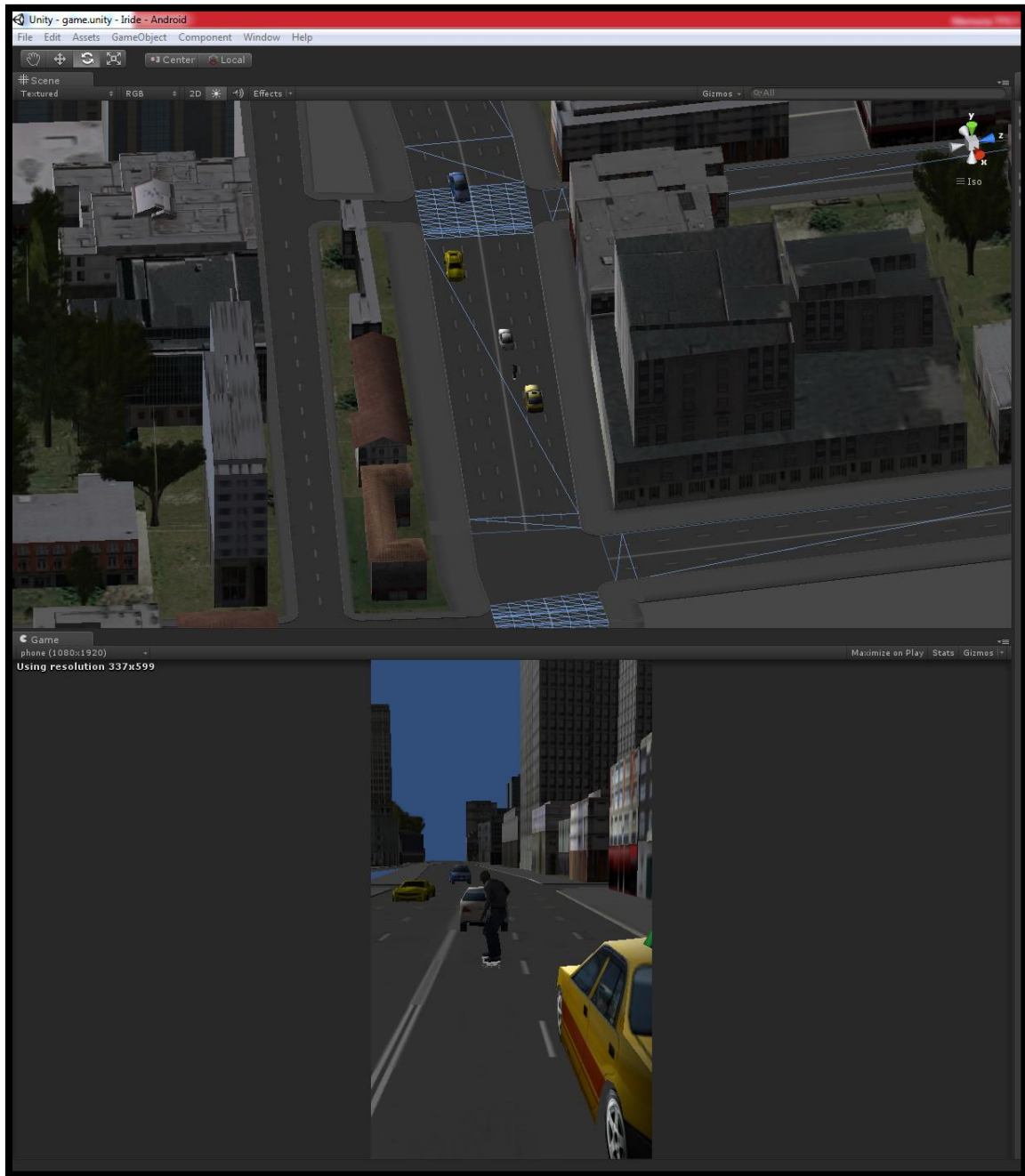


Después de crear la ciudad a nuestro gusto el siguiente paso fue exportarla a 3dsmax. A pesar de haber configurado el programa para obtener una ciudad con el menor número de polígonos posibles, la ciudad era demasiado grande. Por lo tanto la edité con 3dsMax para personalizar el modelado de la ciudad, ajustar como quería y eliminar todas aquellas partes generadas que no necesito para el proyecto, donde lo que necesito es el modelado de una calle larga y bien construida. Todo lo demás no lo necesitamos y no lo usaré porque solo empeoraría el funcionamiento del juego, ralentizándolo debido al alto número de polígonos que eso supone.

2.6 Diseño de Adversarios

De la misma forma que he generado el personaje y los "longboards", hubiese podido generar vehículos para ponerlos a circular por las calles. Pero debido a que ello conlleva una gran inversión de tiempo he decidido seleccionar distintos modelos ya hechos, gratuitos y libres de licencias, para usarlos en el juego.

En la imagen podemos ver algunos de los modelos seleccionados, en la parte de arriba de la imagen con una vista superior y en la parte de debajo con una vista del juego real.



3. Usabilidad.

Antes de empezar el desarrollo del juego decidí asegurar que tanto el diseño del menú, como la navegación y los controles del juego estaban bien pensados y los usuarios finales lo aprobarían.

Realicé un test a cinco usuarios, de edades entre 17 y 23 años, 3 chicos y 2 chicas. El test consistía de dos partes.

En la primera parte se buscaba detectar si la organización de menús y botones en el juego era correcta. Para ello y con los usuarios sin haber visto el juego, se les asignó unas tareas distintas a cada persona. A continuación se les enseñó la maqueta del juego y ellos tenían que hacer lo que sus tareas les indicaban. Después de haber realizado esta prueba les expliqué la segunda parte. En esta parte se les dejó el tiempo que desearan para ver el resto de la aplicación y después respondieron a un cuestionario.

Además aproveché la ocasión para probar distintos modelos de “Smartphone”, se usaron dos modelos, uno de gran tamaño, 5,5” y otro de gama media más pequeño, de 3.8”. De este modo podía ver si el tamaño de la pantalla afectaba de algún modo, aunque esta aplicación no se adapta al tamaño de pantalla, sino que se escala al tamaño de la pantalla.

Después de realizar el test se extrajeron conclusiones y se hizo una modificación de diseño.

3.1 Primera prueba.

Objetivos de la prueba:

- Comprobar que la curva de aprendizaje del juego era pequeña.
- Verificar que la navegación de menús era correcta.
- Confirmar que el título de las entradas del menú no daba pie a errores y por tanto definía bien su contenido.
- Detectar la falta de algún botón o funcionalidad.
- Corroborar que los usuarios podían llevar a cabo las tareas sin errores.

Tareas que recibieron los usuarios:

1. Iniciar el juego, empezar a jugar, pausar el juego y salir.
2. Iniciar el juego, ajustar el volumen en el menú, empezar a jugar, reiniciar la partida, pausar y salir.
3. Iniciar el juego, ver los controles en el menú, empezar a jugar, pausar, reiniciar la partida y salir.
4. Iniciar el juego, ver todos los personajes, seleccionar uno, empezar a jugar, y salir.

5. Iniciar el juego, ver todos los “longboards”, seleccionar uno, empezar a jugar, y salir.

Definición de los usuarios:

1. Chico 1: 22 años, usuario de Android.
2. Chica 2: 17 años, usuaria Android
3. Chica 3: 22 años, usuaria iPhone.
4. Chico 4: 23 años, usuario iPhone.
5. Chico 5: 23 años, usuario Android.

Resultados prueba 1.

Chico 1 y tarea 1:

Iniciar juego	Empezar a jugar	Pausar el juego	Salir
SI	SI	SI	SI
Observaciones: El usuario realizó las tareas rápidamente sin preguntar nada. Salió del juego mediante el botón del móvil. Realizó la prueba con el móvil pequeño.			

Chica 2 y tarea 2:

Iniciar juego	Ajustar Volumen	Empezar juego	Reiniciar partida	Pausar el juego	Salir
SI	SI	SI	SI	SI	SI
Observaciones: Al final pregunto si podía salir directamente por el botón. Realizó la prueba con el móvil pequeño, con botones “Android” físicos.					

Chica 3 y tarea 3:

Iniciar juego	Ver controles	Empezar juego	Pausar juego	Reiniciar el juego	Salir
SI	SI	NO	SI	SI	SI

Observaciones: La usuaria no supo salir sola de la pantalla de ver los controles. Esto se debe a que no está acostumbrada al botón de atrás del sistema "Android", además el móvil con el que realizó la prueba tiene estos botones algo escondidos. Realizó la prueba con el móvil grande, sin botones "Android" físicos.

Chico 4 y tarea 4:

Iniciar juego	Ver personajes	Seleccionar personaje	Empezar juego	Salir
SI	SI	SI	SI	NO

Observaciones: La usuaria no tuvo problemas durante la prueba, pero necesitó ayuda para salir porque no sabía cómo funcionaba el móvil. Realizó la prueba con el móvil grande, sin botones "Android" físicos.

Chico 5 y tarea 5:

Iniciar juego	Ver "longboard"	Seleccionar "longboard"	Empezar juego	Salir
SI	SI	SI	SI	SI

Observaciones: El usuario no tuvo problemas durante la prueba, ni tampoco necesitó ayuda para salir porque ya sabía cómo funcionaba el móvil. Realizó la prueba con el móvil grande, sin botones "Android" físicos.

Análisis de resultados:

1. En la primera tarea se ha realizado un recorrido básico en el cual el usuario no ha tenido ningún problema. Ha cumplido con todo lo que

se le ha pedido con eficacia y rapidez. En este caso no se han detectado anomalías en el juego.

2. En este caso en concreto he puesto a drede primero la tarea de reiniciar la partida antes de la tarea de poner pausa, aun sabiendo que para reiniciar partida se tiene que pausar el juego. Ya que me interesaba saber si los usuarios buscarían la opción reiniciar a partir del menú de pausa. La prueba ha resultado satisfactoria ya que se han completado todas las tareas con éxito.
3. En este caso he querido comprobar si la interfaz de pantallas de información del juego eran funcionales, en este caso la pantalla de instrucciones. Aquí los resultados no han sido buenos, ya que el usuario no ha sabido salir de la pantalla de instrucciones y volver al menú. El juego se ha diseñado pensando en que los usuarios conocerían el botón atrás de su teléfono, para volver al menú principal. Pero en este caso que el teléfono de la prueba funcionaba de distinta forma que el teléfono de la usuaria. Esta no ha pensado en usar un botón que no tiene en el teléfono de "Apple".
4. En este caso el usuario cumplió con las pruebas dentro de la población pero fallo en el salir de la aplicación. Esto se debe a la poca familiaridad del dispositivo que esconde los botones en la pantalla, hay que activarlos, y también a la falta de conocimiento del dispositivo del usuario. Este caso en particular no tendría que repetirse en un caso real en que el usuario sea el dueño del teléfono, ya que se da por supuesto que cada uno sabe usar su móvil.
5. Este caso ha sido cien por cien satisfactorio ya que el usuario, con sus conocimientos, ha desarrollado tanto las tareas de dentro el juego, como la tarea de salir.

Conclusión primera parte:

Después de analizar los resultados detecté dos posibles problemas. El más importante, fue el de no saber volver al menú principal desde las pantallas de información, el otro, no saber salir del juego, que le doy menos importancia ya que creo que en un caso real no sucedería, ya que el problema ha recaído más en el uso del hardware que no del software.

Ante la preocupación de que algunos usuarios tengan ese problema en el menú, decidí añadir el botón atrás en las pantallas de menú, además de poder seguir usando el botón propio del dispositivo. Añadiendo este botón no se añade ninguna nueva funcionalidad, solo se duplica el botón atrás, para así asegurarnos que todos los usuarios puedan salir de las pantallas de información para volver al menú principal. Además esto también ayuda

a perfilar el juego si se decide exportarlo a la plataforma del iPhone, IO'.



Antes



Después

Aquí podemos ver el antes y después como resultado de la prueba.

3.1 Segunda prueba.

Esta prueba se realizó después de que los usuarios terminaran su lista de tareas. Solo se les dijo que se tomaran el tiempo necesario para ver toda la aplicación y que después debían completar un pequeño test.

El objetivo de la prueba era conocer las posibles dudas que pudiesen tener los usuarios en el resto de la aplicación, ya que no habían visto la aplicación en su totalidad.

¿Crees que los títulos del menú describen su contenido?	¿Navegando por los menús te has sentido perdido?	¿Te ha resultado difícil aprender a jugar?	¿La información que da el juego durante la partida es suficiente?	¿Qué añadirías o eliminarías del juego?

Esta es el cuestionario que respondieron los usuarios.

Resultados de los usuarios:

Chico 1

¿Crees que los títulos del menú describen su contenido?	¿Navegando por los menús te has sentido perdido?	¿Te ha resultado difícil aprender a jugar?	¿La información que da el juego durante la partida es suficiente?	¿Qué añadirías o eliminarías del juego?
SI	NO	NO	SI	Botón atrás en algunas pantallas.

Chica 2

¿Crees que los títulos del menú describen su contenido?	¿Navegando por los menús te has sentido perdido?	¿Te ha resultado difícil aprender a jugar?	¿La información que da el juego durante la partida es suficiente?	¿Qué añadirías o eliminarías del juego?
SI	SI	NO	SI	Botón atrás en algunas pantallas

Chica 3

¿Crees que los títulos del menú describen su contenido?	¿Navegando por los menús te has sentido perdido?	¿Te ha resultado difícil aprender a jugar?	¿La información que da el juego durante la partida es suficiente?	¿Qué añadirías o eliminarías del juego?
SI	NO	NO	SI	Botón atrás.

Chico4

¿Crees que los títulos del menú describen su contenido?	¿Navegando por los menús te has sentido perdido?	¿Te ha resultado difícil aprender a jugar?	¿La información que da el juego durante la partida es suficiente?	¿Qué añadirías o eliminarías del juego?
SI	NO	NO	SI	Me parece bien.

Chico 5

¿Crees que los títulos del menú describen su contenido?	¿Navegando por los menús te has sentido perdido?	¿Te ha resultado difícil aprender a jugar?	¿La información que da el juego durante la partida es suficiente?	¿Qué añadirías o eliminarías del juego?
SI	NO	SI	SI	Está bien.

Conclusiones segunda parte:

Si analizamos las respuestas que han dado los usuarios a las preguntas podemos ver que siguen en el mismo camino que las conclusiones de la primera prueba, los usuarios aprueban el juego pero han tenido dudas con el sistema de navegar con el botón atrás del móvil. Por este motivo he optado por añadir este botón y así erradicar las posibles dudas que pudieran surgir en el futuro en los usuarios finales.

Conclusiones finales:

Gracias al uso de tests de usabilidad mediante “mockups” he obtenido las sensaciones de los usuarios sobre el producto y he podido mejorarlo corrigiendo errores cometidos en el proceso de diseño.

Por lo tanto queda demostrado la eficacia de estos test.

4. Desarrollo.

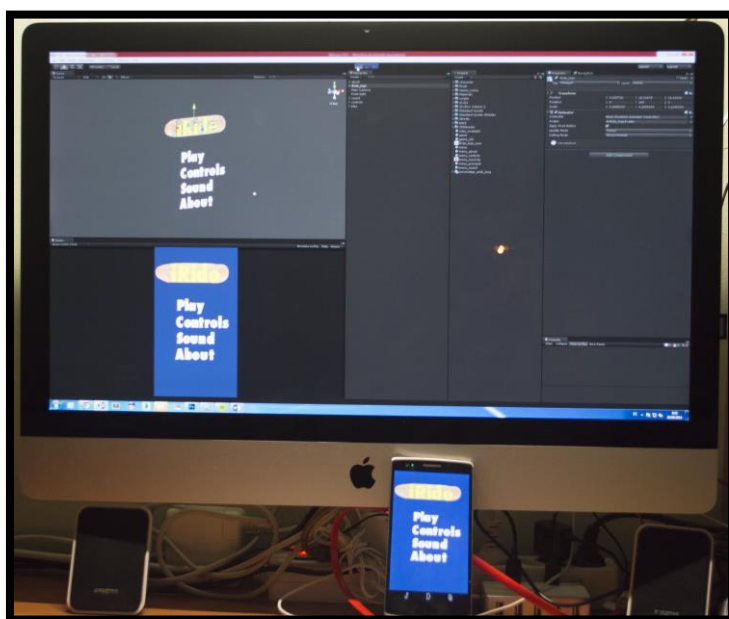
3.1 ¿Por qué Unity?

Unity es un motor gráfico principalmente pensado para juegos. Nos permite desarrollar juegos a partir de un entorno gráfico mucho más amigable. Este editor nos permite desarrollar proyectos tanto en entornos 2d como en entornos 3d. En mi caso lo he utilizado en modo 3d para gestionar todo el material generado en 3d. Además del entorno gráfico, también hay herramientas para crear las escenas definitivas a partir de los modelos importados, como luces y cámaras.

Además puedes crear los scripts para controlar las funciones del juego en 3 lenguajes distintos, javascript, Boo o C#, en este caso he utilizado el último, C#.

Otra de las herramientas que ofrece Unity es la posibilidad de usar un dispositivo móvil durante el desarrollo sin tener que exportar la aplicación ni instalarla en el dispositivo en cuestión. Para que esto funcione solo tenemos que instalar el paquete de desarrollo de android para ordenador, Google Android SDK, y también tenemos que instalar una aplicación en el móvil llamada Unity Remote, ambas descargas son gratuitas. Esta utilidad tiene la peculiaridad que para hacer estas pruebas en un dispositivo android, tienes que ejecutar Unity en Windows, y si quieres hacerlo en un dispositivo Apple, tienes que hacerlo desde un sistema operativo mac osx.

Por otro lado, a causa de la política de Apple, solo Unity desde mac puede generar archivos “.ipa”, el archivo correspondiente a la aplicación finalizada. Esto es un problema para todos los usuarios de Unity que no desarrollen desde un mac, ya que no podrán lanzar aplicaciones en IOS, el sistema operativo móvil de Apple. En cambio los usuarios que si lo hagan desde mac, podrán finalizar aplicaciones tanto en IOS como en Android, los dos sistemas operativos móviles más importantes en el mercado.



En esta imagen podemos ver Unity Remote funcionando. Tenemos el “Smartphone” conectado mediante usb al ordenador, y cuando ejecutamos el programa dentro del editor para comprobar los resultados, vemos como también se ve en el móvil. Además podemos usar la pantalla táctil y los demás controles como si la aplicación fuera real.

Poder realizar esta tarea con Unity ha sido mucho mas cómodo, si no lo hubiésemos usado, para hacer estas pruebas, lo más normal hubiese sido utilizar el emulador de Android o IOS que vienen por defecto con los packs de desarrollo de aplicaciones correspondientes, Android SDK o Xcode.

3.2 Aspectos técnicos de programación.

La programación del juego se ha planteado mediante dos estructuras principales, una para la navegación de menús y otra parte que controlará las funciones propias del juego.

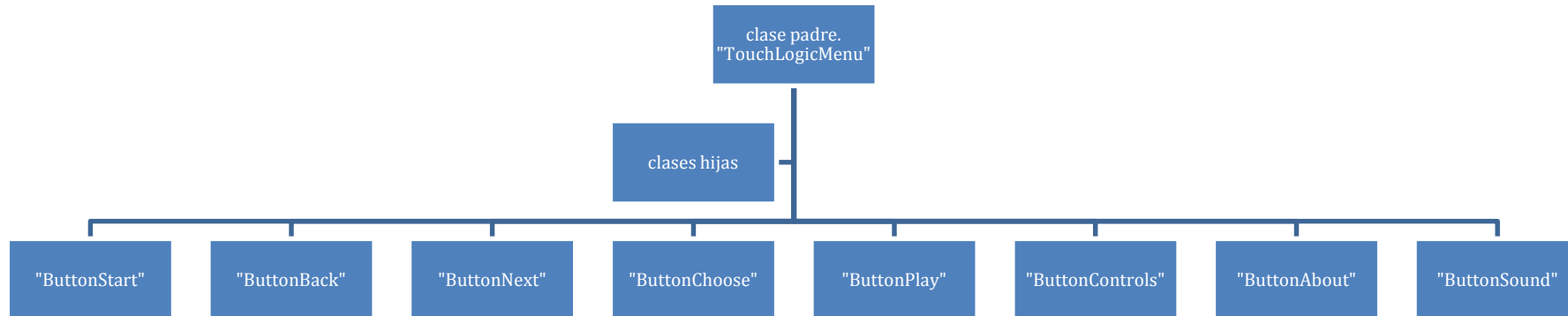
3.2.1 Clases del menú:

Los menús del juego funcionan en base a una clase padre llamada "TouchLogicMenu", esta función se ejecuta en el entorno llamado "Update", lo que significa que se ejecutará una vez por fotograma. La función tiene la tarea de controlar los toques en la pantalla y detectar si es el caso que el dedo se ha colocado encima de un botón u objeto tocable. Esto se lleva a cabo mediante el sudo del sistema "RayCast", esto se puede traducir como lanzamiento de rayo, su función es lanzar un rayo imaginario desde el punto donde se ha tocado la pantalla hacia dentro de la pantalla. Si el rayo toca algún objeto preparado para ser tocado, lo sabremos y podremos obtener información del objeto tocado.

En nuestro caso práctico cuando el rayo toca un objeto nos guardamos su nombre. Después el siguiente procedimiento lanza un mensaje que contiene "Touched" más el nombre del objeto con el que el rayo ha chocado. Este mensaje será recibido por alguna clase hija, que al ver el mensaje ejecutará sus funciones.

Existe una clase hija para cada botón, que extienden de la clase padre "TouchLogicMenu", de esta manera podemos definir la funcionalidad de cada botón. Estas funciones se podrían haber compactado en un solo script pero prefiero tener un esquema más grande compuesto de pequeñas partes que no todo en un script muy largo, de este modo es más fácil detectar y solucionar errores.

3.2.1.1 Esquema de clases del menú:



3.2.2 Clases del juego:

El juego en sí se plantea con dos clases principales, una que se encarga de gestionar todo lo relacionado con los movimientos del personaje y otro que controla la jugabilidad.

La parte que controla el jugador controlará los tres movimientos que puede hacer el jugador, el avance automático que acelera en función del tiempo que pase, el salto, el giro en esquinas y el deslizamiento lateral.

El avance automático será un valor de velocidad que irá aumentando exponencialmente, ya que su valor se multiplicará por el tiempo de la partida. A pesar de que este valor esté en otra clase, será una variable pública para que pueda ser accesible desde otras clases.

El salto y el giro en las esquinas tendrán un controlador con una parte en común, pues las dos acciones se realizan mediante gestos táctiles. El controlador común se ejecutará una vez en cada fotograma y detectará los gestos, en función del tipo de gesto detectado se llamará a la función específica del gesto, es decir a la función de salto o la de giro, para que ellas se encarguen de dar respuesta al gesto con la acción que toque.

Por último en esta clase habrá otro controlador que también se ejecutará una vez cada fotograma y será el encargado de detectar la oscilación del terminal en el eje "x", este movimiento es más conocido como "Tilt". Cuando se mueva el terminal este multiplicará el valor de ángulo por una variable fija, a la que llamaremos velocidad de horizontal, y el valor resultante indicará la nueva posición en la que debe colocarse el personaje.

Por otro lado habrá el otro conjunto de clases que controlará los demás elementos del juego. Dicha clase controlará:

- El tiempo de reloj.
- Las colisiones con paredes y vehículos.
- La creación de los elementos recolectables.
- La creación y control de los vehículos enemigos.
- El menú flotante de pausa.

```

1 using UnityEngine;
2 using System.Collections;
3
4 public class accelerometerInput : MonoBehaviour
5 {
6     public float TiltSpeed = 5; //velocidad de desplazamiento
7     private float Zspeed;
8     public Vector3 forward ;
9
10
11     public float Speed = 50; //variable velocidad de avance
12     public float heigh =1;
13
14     public Vector3 tilt;
15
16     void Update() {
17
18         tilt.x = -(Input.acceleration.x); //detectar la inclinacion del acelerometro
19
20
21         Zspeed = tilt.z/Speed;
22         //Debug.Log (tilt.z+"/"+Speed+"/"+Zspeed);
23         tilt.z = (tilt.z+Speed);
24         //Debug.Log (tilt);
25
26         if (tilt.sqrMagnitude > 1)
27             tilt.Normalize();
28
29         transform.Translate((tilt.x * TiltSpeed), 0, -Zspeed); //Asignar valores a la posicion del personaje
30     }
31 }

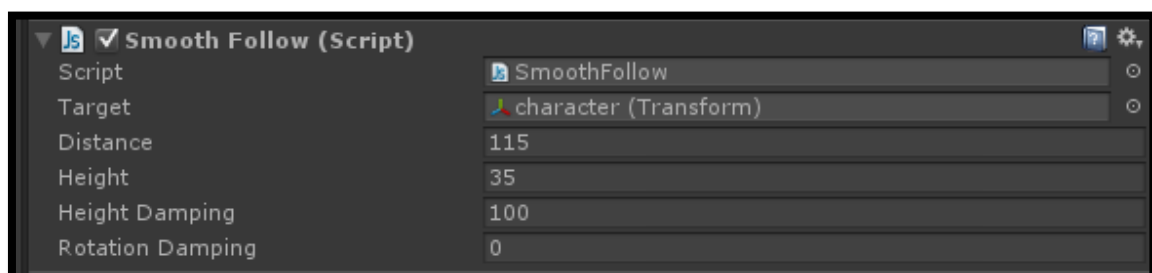
```

En la imagen superior vemos una captura de la clase que se encarga de controlar parte de los movimientos del personaje. La clase se llama “accelerometerInput”, está formada por distintas variables y acciones que se concentran en el “void Update ()”, para que las acciones se ejecuten una vez por fotograma.

Principalmente usamos la función “Input.accelerometer.x”, una función interna de Unity que nos devuelve los valores de inclinación, para almacenarlos en la posición “x” de un vector llamado “tilt”. Finalmente en la última línea de código, multiplicamos el valor del acelerómetro por una constante, para conseguir un movimiento más rápido, y finalmente también asignamos “Zspeed” a la posición “z” del patinador. “Zspeed” es una variable que contiene un número que se incrementa con el paso del tiempo y al asignarlo a esta posición representa el avance del jugador.

En el juego solo aplicamos los movimientos al personaje, ya que hemos aplicado a la cámara una clase llamada “Smooth Follow”, que se encarga de seguir los movimientos que haga el personaje para mantenerlo siempre en pantalla.

A continuación podemos ver una captura de imagen de la configuración de esta clase.



```
crashingControl ▶ OnCollisionEnter (UnityEngine.Collision hit)
1 using UnityEngine;
2 using System.Collections;
3
4 public class crashingControl : MonoBehaviour {
5
6     // Use this for initialization
7     void Start () {
8
9     }
10
11    // Update is called once per frame
12    void Update () {
13
14    }
15
16    private string hitobject;
17
18    void OnCollisionEnter(UnityEngine.Collision hit)
19    {
20        //Debug.Log ("collision");
21        hitobject = hit.gameObject.name;
22        if(hitobject == "skater")
23        {
24            //Debug.Log ("in_collision");
25            Application.LoadLevel("gameOver");
26
27        }
28    }
29 }
30
```

La clase que vemos en esta imagen se llama “crashingControl” y su función es controlar que el personaje no choque con objetos. La clase en cuestión se aplica tanto a coches como a las paredes. Para que funcione tanto el personaje como los elementos es necesario que estos tengan un objeto “Collider” en su composición. Este tipo de objeto predefinido en Unity se usa para marcar físicamente el volumen de un objeto. Tanto se puede hacer de manera perfecta mediante el modelado del objeto o mediante formas regulares como cajas o esferas. En estos casos un problema muy frecuente es que al usar un modelado con muchos detalles cargamos mucho el sistema y causamos lentitud.

En la clase comprobamos una vez cada fotograma si el “Collider” del objeto del escenario está chocando con el “Collider” de otro objeto, en la línea 18. Después con un “if” vemos si el nombre del objeto que ha tocado es “skater”, solo en ese caso se terminaría el juego y nos llevaría la pantalla de “Game Over”. En el juego el objeto “skater” es el patinador.

3.3 test en dispositivos móviles

Hoy en día hay una variedad de “Smartphones” muy elevada, los hay con precios y especificaciones muy distintas, de hecho parece mentira que sean productos de una misma categoría. Esto causa que sea imposible probar esta o cualquier otra aplicación en todos los dispositivos compatibles. Aun así podemos ayudarnos de emuladores para ordenador de sistemas operativos Android. Con ellos podemos probar el juego emulando las características de dispositivos de potencia y tamaños de pantalla diversos para ver como se comporta la aplicación en todos ellos.

Adicionalmente si no hay errores en pantalla detectados en los emuladores, se puede organizar un test final con usuarios. Este test consistiría en conseguir un grupo de voluntarios, representativos del target y dotados con sus propios móviles, para que se descargasen la aplicación, la instalen y la ejecuten. Después se les puede pedir que contesten un test para conocer a fondo si el procedimiento ha sido correcto, qué ha podido fallar y su opinión de la aplicación.

5. Conclusiones

Gracias a este proyecto he podido profundizar mis conocimientos alrededor de los videojuegos para “Smartphone”, ya que no cursé la materia optativa específica sobre creación de videojuegos, además, también el hecho de realizar el proyecto en solitario hace que las decisiones no se tomen con toda la confianza ya que no hay equipo. Una de las cosas que más me he dado cuenta durante el proyecto es que por pequeña que sea una idea, en este caso la creación de un pequeño videojuego, es mucho mejor contar con un equipo de gente.

En este caso en particular, donde se ha desarrollado el proyecto sin un equipo de diseño, estoy muy contento de haber podido realizar pruebas de usabilidad, ya que me ayudaron a conocer la aceptación real y a la detección de errores.

También tengo que comentar que empecé el trabajo con la idea de poder llevar a cabo toda la producción, incluyendo el diseño y la implementación, cosa que no ha sido posible. No tanto por un tema de incapacidad sino más bien por haber invertido las horas de dedicación del proyecto al diseño y demás pruebas. Y es que ante el posible miedo de que obtuviera un resultado final visualmente pobre, quise asegurarme demasiado bien el apartado de concepción y diseño.

Sin embargo la realización de este trabajo me ha permitido pensar y diseñar un juego y poner en marcha el proyecto en condiciones y con la seguridad de que, aunque el proyecto académico termine, el proceso de creación del juego seguirá adelante con la garantía de un buen diseño en todos los aspectos.

6. Bibliografia

Game Design:

http://badgeville.com/wiki/Game_Design

<http://gameaccessibilityguidelines.com/>

Guías de usabilidad:

<https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/>

<http://developer.android.com/design/get-started/principles.html>

<http://www.simpleusability.com/beinspired/2012/01/mobile-gaming-usability/>

<http://www.slideshare.net/apigee/mobile-app-design-best-practices-usable-interfaces-for-tiny-places>

7. Anexo

7.1 Game Design

DOCUMENTO DE DISEÑO

1. Fecha

1.1. Inicio:

1.2. Última modificación:

1.3. Última modificación:

2. Nombre:

iRide

3. Nombre de la unidad de contenido:

Juego principal

4. Tipología de juego:

Arcade, Sports

4.1. Vista del usuario:

3D

4.2. Modo de juego:

Tercera persona.

4.3. Estilo de grafismo:

Entorno 3d con detalles como edificios conocidos para ubicar el juego.

5. Jugabilidad

5.1. Breve definición de la historia

El juego nace para llevar el placer del “longboard” a tus tiempos libres. Es un juego del tipo “*free time-wasters*” para llenar los tiempos de espera. Será un juego sin fin que irá aumentando la dificultad y la velocidad a medida que se avance. El objetivo será aguantar el mayor tiempo posible con vida y como habrá objetos para recolectar que nos darán puntos también se tendrán en cuenta en caso de empates en el tiempo.

A medida que el usuario consiga puntos podrá desbloquear personajes, “longboards” y nuevas ciudades.

5.2. Definición de las reglas i los retos

El usuario tendrá tres opciones de control táctiles y también el acelerómetro del Smartphone. Con las opciones táctiles se podrá girar al final de la calle a derecha o izquierda mediante gestos, *slide to left* o *slide to right*, y también podremos hacer el gesto hacia arriba para hacer saltar al personaje, *slide to top*. Además inclinándolo el Smartphone a derecha o izquierda moveremos al personaje a un lado u otro de la calle, pero será necesario el gesto para girar la calle.

El juego es infinito pero tendremos dos modos de llegar al game over, la primera ocurrirá si no giramos la calle a tiempo y caemos o chocamos contra la pared, después también terminaremos antes de tiempo si colisionamos con objetos que habrá repartidos por el escenario.

Durante el juego podremos recoger objetos recolectables que nos darán puntos. Para recoger los objetos bastará con cruzarlos con nuestro personaje. Estos objetos tendrán un espacio en el hud para poder ver la cantidad recogida. Los puntos obtenidos servirán para desbloquear niveles y nuevos personajes.

Los niveles del juego se generarán aleatoriamente a partir de distintos modelos 3D de las calles. El juego unirá las intersecciones entre calles con modelos de calles que se ordenarán de forma aleatoria para conseguir que el juego sea distinto cada vez que lo ejecutamos.

Como reto añadido tendremos la posibilidad de publicar, en la web del juego o en nuestra red social favorita, nuestro tiempo de juego, que como más grande sea mejor ya que querrá decir que hemos seguido corriendo más tiempo. De este modo el juego ganará jugadores ya que si publicamos el juego en la red social, nuestros amigos querrán jugar para obtener un tiempo mejor.

5.3. Control de los personajes

Controlaremos al patinador mediante la pantalla táctil y el acelerómetro del móvil. Tocando en la parte superior de la pantalla, haremos saltar al jugador, y si tocamos a izquierda o derecha, giraremos en una esquina.

El acelerómetro nos permite desplazarnos a izquierda y derecha ya sea en mitad de un salto o en una plataforma. De manera que si el jugador inclina el móvil en el eje horizontal a la izquierda o a la derecha el patinador se desplaza en esa dirección.

También tendremos que realizar gestos encima de la pantalla para girar al final de las calles. Es decir, deslizar el dedo a izquierda o derecha activará el cambio de dirección del personaje, ya que con el acelerómetro nos desplazaremos a izquierda o derecha pero manteniendo siempre la misma dirección. Por lo tanto al llegar en el punto en que el mapa nos obligue a girar a izquierda o derecha para no chocar y así poder seguir jugando, tendremos que realizar el gesto en la pantalla.

Por último para llegar a algunos objetos que al cogerlos nos darán puntos, será necesario hacer saltar al personaje. Para hacerlo usaremos un gesto parecido a los anteriores, pero en la dirección vertical, deslizar hacia arriba.

Con estos controles sencillos basados en una interacción natural con el dispositivo a la cual los usuarios ya están acostumbrados, conseguiremos una curva de aprendizaje del juego muy pequeña. Esto es muy importante en juegos como este y otros del mismo estilo, ya que facilita mucho la manera de interactuar con el juego. Como ya sabemos que este tipo de juegos son para ratos cortos, es muy importante reducir el tiempo que el usuario necesita para aprender a jugar ya que el objetivo es entretener al usuario y que empiece a jugar cuando antes. Para así aumentar el tiempo de juego en el corto período de tiempo en el que la aplicación se está ejecutando.

Además, al no usar una interfaz complicada facilitamos el uso del juego a personas que tengan problemas de motricidad o movilidad reducida.

5.4. Definición de los enemigos:

Los enemigos del patinador serán los mismos que se encontraría un patinador en el día a día, los coches y demás vehículos, que son los que causan más problemas a los patinadores. De este modo en el juego podremos usar el acelerómetro para movernos a derecha o izquierda para esquivar coches y camiones y también en caso de que nos encontremos un coche deportivo o “bajo”, lo podremos saltar realizando en el gesto correspondiente en el momento adecuado.

5.6. Definición de los objetos

El objetivo del juego es estar el máximo tiempo posible jugando si chocar con ningún coche o pared, ya que cuando una de estas dos cosas pase se terminará la partida actual y tendremos que volver a empezar desde el principio. Además en el juego habrá objetos para recolectar lo que será una motivación extra para el usuario.

5.7. Orientación del usuario

El usuario siempre tiene una vista de tercera persona sobre el patinador, esta vista será fija y no se podrá cambiar ni mover. El usuario siempre tendrá a la vista la espalda del personaje y el entorno, para poder ver los coches, la calle y los objetos colectables.

5.8. Interfaz

Hud superior con los siguientes datos:

- Reloj de tiempo que llevamos jugando, parte superior en el centro de la pantalla.
- Contador de objetos recolectados, parte superior derecha.
- Botón de pausa, en la parte superior izquierda.

5.9. Efectos de audio

- Música de fondo de menús.
- Música de fondo en el juego.
- Sonido recogida de objeto.
- Sonido rodadura longboard.
- Sonido coches al pasar.
- Sonido ambiente ciudad.
- Sonido de queja del personaje al chocar.

5.10. Curva de dificultad

El juego dispone de una curva de dificultad progresiva. La dificultad se mide con la velocidad a la que se mueve el patinador y esta dependerá del tiempo que llevemos jugando, siendo poca al principio y se irá multiplicando a medida que pase el tiempo de forma gradual.

A medida que aumente el tiempo aumentará la velocidad del juego, nos encontraremos antes con los coches y con las paredes y cada vez será más difícil no chocar con nada, con lo que se requerirá una mayor habilidad por parte del usuario.

5.11. Curva de aprendizaje

La curva de aprendizaje del juego será pequeña, con pocos controles y basados en gestos naturales y cotidianos en la interacción con dispositivos móviles. El jugador verá los controles en pantalla la primera vez que juegue al juego, además estos también estarán disponibles en el menú del juego para poder ser consultados cuando el usuario quiera.

6. Relación usuario-juego:

6.1. Interactividad física:

a. pantalla táctil:

- La interacción en los menús se realiza tocando las palabras.
- En el juego usaremos el gesto deslizar a la derecha o a la izquierda, depende del lado que escoja él usuario, para cambiar la dirección.
- Deslizar el dedo en dirección vertical hacia arriba para hacer saltar al personaje.
- Pulsando el botón situado en la parte superior izquierda de la pantalla se entrara en el menú de pausa.

b. acelerómetro:

- Girando el móvil se desplaza al patinador a izquierda y derecha, ya sea en mitad de un salto o en el suelo de forma indistinta.

c. botones físicos:

- Los botones, físicos o no, del dispositivo no se utilizarán de manera específica. Se podrán utilizar gestionados directamente por el sistema operativo y no tendrán ningún papel fundamental en el juego. Para salir del juego o dejar de utilizar la aplicación, se utilizarán los sistemas provistos por el SO, normalmente el botón home o el de multitarea para acceder a otras aplicaciones que ya se están ejecutando.

6.2. Interactividad lógica:

- Enemigos: Consideraremos el tráfico urbano como enemigos, si el patinador toca algún vehículo, chocará, y se dará la partida por terminada. Para seguir jugando tendremos que volver a empezar.
- Si al final de una calle no giramos, chocaremos contra la pared al tocarla, y se terminará la partida del mismo modo como si chocamos con un coche.
- Objetos: Si tocamos un objeto coleccionable lo recogeremos.

7. Relación juego-usuario:

7.1. Diseño del nivel

7.1.1. Ambientación

Los niveles del juego consistirán en distintas ciudades que el usuario podrá escoger a medida que vaya obteniendo más puntos recolectables

El color del edificio cambia cada nivel y el tipo de edificio cambia cada mundo.

El juego no tendrá un número de ciudades fijas, se irán añadiendo ciudades en función de la acogida del juego.

7.1.2. Dimensiones

Todos los niveles serán iguales, generados aleatoriamente a partir de mapas 3d de calles que se juntarán de manera también aleatoria. Lo que cambiara es la ambientación de la calle para situarse a una ciudad u otra.

7.1.3. Puntos de guardado de partida

No se podrá guardar la partida, solo quedarán guardados los puntos recogidos y un histórico de tiempos, pero tendremos que seguir jugando hasta que choquemos y terminamos la partida.

7.1.4. Pausa

El juego tendrá un botón de pausa por si surge algún imprevisto, además este menú ofrecerá la posibilidad de reiniciar el nivel, salir al menú principal o continuar pero, al cerrarse la aplicación se perderán los datos pausados.

7.2. Objetos

7.2.1. Estructurales

7.2.1.1. Que decoran y no son colisionables

El suelo y las paredes que dan la ambientación de la ciudad.

7.2.1.2. Que son colisionables:

La pared frontal que indica el fin de la calle.

7.2.2. Interactivos

Los vehículos y las paredes frontales, que nos harán reiniciar la partida.

Las ruedas que nos darán los puntos recogidos

7.3. Enemigos:

7.3.1. Breve definición literaria

Los enemigos son los vehículos que nos encontraremos por la calle, ya que el patinador circulará por entre el tráfico.

7.3.2. Ubicación

Los vehículos circularán por la calzada.

8. Extras:

8.1. Opciones de guardar partida

La partida se guarda automáticamente cuando se superé un nivel.

8.2. Relación historial

A medida que el jugador completa niveles le aparecen como desbloqueados en el menú principal.

9. Limitaciones tecnológicas

Juego para dispositivos con el sistema operativo de Google, android.

7.2 Project

A continuación anexo el documento de organización de Project exportado como pdf para que se pueda ver toda la organización de tareas.

ID	Task Mode	Task Name	Duration	Start	Finish												
						S	T	08 Jul '13	M	F	T	05 Aug '13	S	W	02 Sep '13	S	T
1																	
2		Game Design		Tue 02/07/13													
3		<i>Idea inicial</i>	46 days?	Mon 01/07/13	Mon 02/09/13												
4		diseño jugabilidad	45 days?	Tue 03/09/13	Mon 04/11/13												
5		Relación usuario juego	45 days?	Tue 03/09/13	Mon 04/11/13												
6		Relación juegos usuario	45 days	Tue 03/09/13	Mon 04/11/13												
7																	
8		Creación de arte	190 days	Tue 05/11/13	Mon 28/07/14												
9		Diseño "look and feel"	190 days?	Tue 05/11/13	Mon 28/07/14												
10		Bocetos	45 days	Tue 05/11/13	Sun 05/01/14												
11		Diseño maqueta final	96 days	Mon 20/01/14	Mon 02/06/14												
12		Corrección de errores de diseño	36 days	Mon 09/06/14	Mon 28/07/14												
13		Diseño Personajes	190 days?	Tue 05/11/13	Mon 28/07/14												
14		Bocetos	45 days?	Tue 05/11/13	Sun 05/01/14												
15		Diseño maqueta final	96 days?	Mon 20/01/14	Mon 02/06/14												

Project: Iride_project_developm
Date: Tue 07/10/14
















7.2 Project

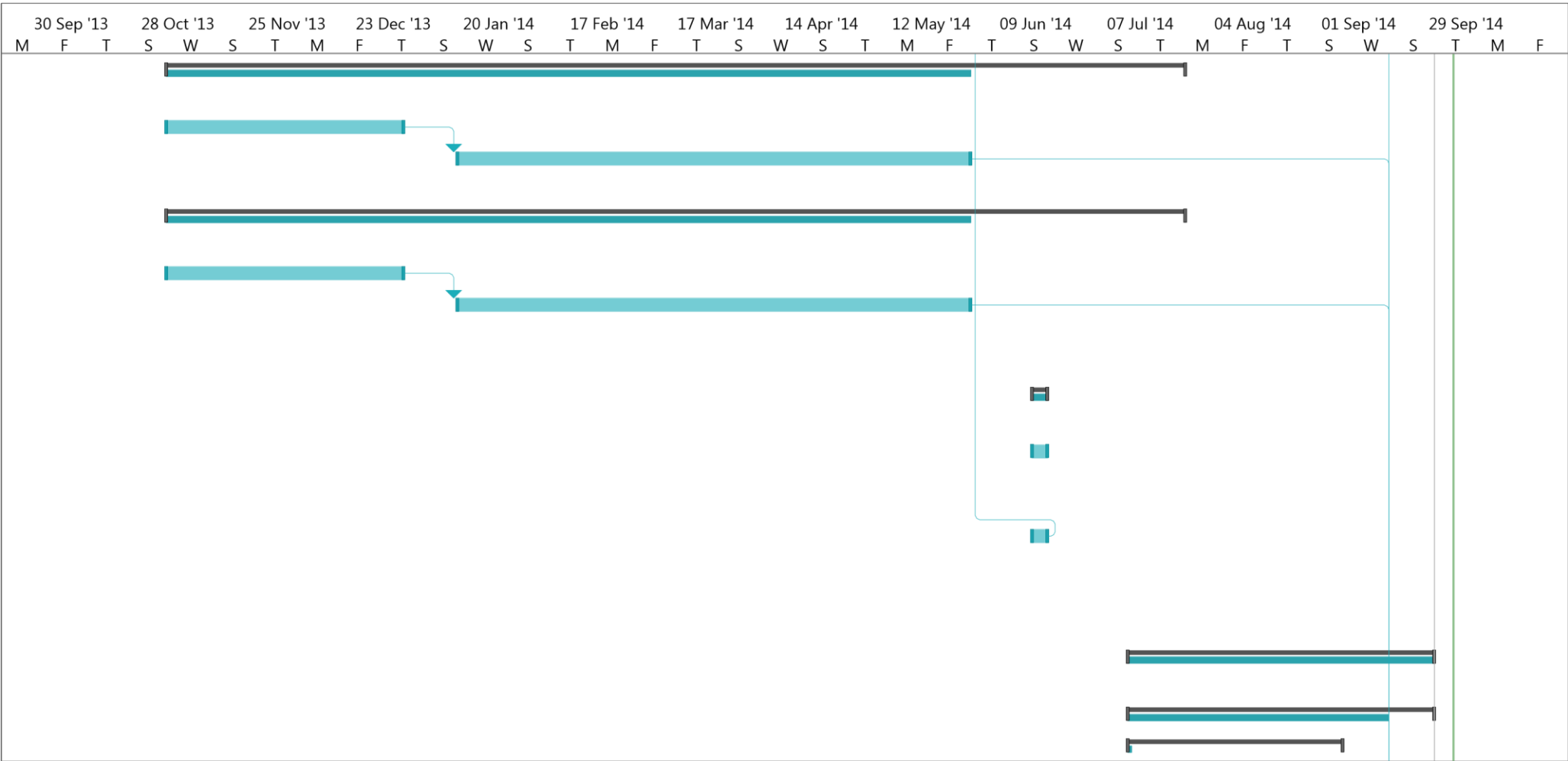
Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	
Milestone		Duration-only		Deadline	
Summary		Manual Summary Rollup		Progress	
Project Summary		Manual Summary		Manual Progress	
Inactive Task		Start-only			
Inactive Milestone		Finish-only			

36		Desarrollo juego	58 days?	Mon 14/07/14	Wed 01/10/14
37		Vistas	58 days?	Mon 14/07/14	Wed 01/10/14
39		Controlador	39 days?	Sat 09/08/14	Wed 01/10/14
46					
47		Test en dispositivos	9 days	Sun 21/09/14	Wed 01/10/14
48					
49		Test en usuarios finales	9 days?	Sun 21/09/14	Wed 01/10/14

Project:
Iride_project_developm

Date: Tue 07/10/14

Task	Inactive		SummaryExternal Tasks			
Split	Manual Task		External Milestone			
Milestone	Duration-		only Deadline			
Summary	Manual		Summary Rollup		Progress	
Project	Summary		Manual Summary Manual Progress			
Inactive Task	Start-only					
Inactive Milestone	Finish-only					



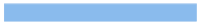

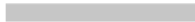





Project: Iride_project_developm

Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	
Milestone		Duration-only		Deadline	

30 Sep '13 28 Oct '13 25 Nov '13 23 Dec '13 20 Jan '14 17 Feb '14 17 Mar '14 14 Apr '14 12 May '14 09 Jun '14 07 Jul '14 04 Aug '14 01 Sep '14 29 Sep '14
M F T S W S T M F T S W S T M F T S W S T M F T S W S T M F



Project: Iride_project_developm

Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	
Milestone		Duration-only		Deadline	