



Pirueto's Race: Videojuego 3D en Unity destinado al aprendizaje del Alfabeto dactilológico

Sonia Alvarez Campá

Memoria del Proyecto Final del Grado Multimedia

Narrativas 2D/3D

Consultor: Andreu Gilaberte Redondo

Professor: Antoni Marín Amatller

20/06/2016

Créditos/Copyright

Esta obra está sujeta a la siguiente licencia:



Creative Commons Reconocimiento – NoComercial – SinObraDerivada

(by-nc-nd): No se permite un uso comercial de la obra original ni la generación de obras derivadas. <http://creativecommons.org/licenses/by-nc-nd/4.0/>

Tipografías utilizadas:

Porky's Font Autor: Fontalicious. Licencia gratuita para uso personal.

<http://www.dafont.com/es/font-comment.php?file=porkys>

Anja Eliane Font. Autor: Edson & Eliane. Licencia gratuita para uso no comercial.

http://www.dafont.com/es/font-comment.php?file=anja_eliane

Dissaramas CRU Fingerspelling Font. Autor: CRU-Dissaramas Dungkratok (2008).

Licencia de Dominio Público y uso Comercial. Version 1.0 Marzo 2015

<http://www.fontspace.com/cru-dissaramas/dissaramas-cru-fingerspelling>

Efectos de Sonido FX y Música del Juego:

FX's Sound: Taira Komori's Japanese Free Sound Effects:

<http://taira-komori.jpn.org/freesounden.html>

Music Game 1: Cookie Island. Free Loop Short Edit. Licencia CC-BY 3.0.

<http://www.playonloop.com/2015-music-loops/cookie-island/>

Music Game 2: Across the skies. Free Loop Short Edit. Licencia CC-BY 3.0.

<http://www.playonloop.com/2015-music-loops/across-the-skies/>

FICHA DEL TRABAJO FINAL

Título del Trabajo:	<i>Piruetto's Race: Videojuego 3D Destinado al Aprendizaje del Alfabeto Dactilológico</i>
Nombre del Autor:	<i>Sonia Alvarez Campá</i>
Nombre del Consultor:	<i>Andreu Gilaberte Redondo</i>
Nombre del PRA:	<i>Antoni Marin Amatller</i>
Fecha de entrega:	<i>06/2016</i>
Titulación o programa:	<i>Grado Multimedia</i>
Área del Trabajo Final:	<i>Narrativas 2D/3D</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave:	<i>Videojuego Interactivo, Lengua de Signos, Kawaii</i>
Resumen del trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo</i>	
<p>El presente Trabajo de Final de Grado se enmarca dentro de la Especialidad de Comunicación Visual y Creatividad del Grado Multimedia de la <i>Universitat Oberta de Catalunya (UOC)</i> y consiste en el desarrollo de un videojuego educativo en 3D, con estética <i>Kawaii</i>, destinado al aprendizaje del lenguaje de signos.</p> <p>La realización del proyecto engloba todas las fases del desarrollo, desde su concepción inicial hasta la obtención del producto final, siendo su objetivo principal aportar una función educativa además de generar entretenimiento. El proyecto pretende concienciar y normalizar el uso la Lengua de Signos entre todos los usuarios, ya que son pocas las iniciativas de productos y servicios que tienen en cuenta esta Lengua.</p> <p>El alcance de este proyecto incluye el aprendizaje del motor gráfico de Videojuegos Unity y el lenguaje de programación <i>C Sharp Script</i>, modelado de elementos 3D y Arte 2D para proponer un escenario creativo, original, personal y sensible. Un proyecto que pretende ir más allá de la concepción inicial de los videojuegos y cumplir una función social, aplicando el uso de las nuevas tecnologías al ámbito educativo.</p>	

Abstract (in English, 250 words or less):

This work is framed within the specialty of Visual Communication and Creativity Grade Multimedia at the Open University of Catalonia (UOC) and it is about the development of an educational game in 3D, with Kawaii style, destined to sign language learning.

The project encompasses all phases of development, from initial conception to obtain the final product, its main objective is to provide an educational function and generate entertainment. The project aims to raise awareness and standardize the use sign language between all users, because there are few initiatives of products and services that consider the Sign Language.

The reaching of this project includes the learning of Unity Game Engine and the language of programming C Sharp Script, 3D modeling and 2D art elements to propose a creative, original, personal and sensitive stage. A project that to go beyond the initial conception of video games and fulfill a social function, applying the use of new technologies to education.

Dedicatoria/Cita

"La lengua de signos está llena de plasticidad, belleza y es capaz de crear la magia de la poesía y de envolver a las personas en un mundo onírico lleno de imágenes fantásticas. Sirve para confesarse, para la filosofía, para discutir o hacer el amor. Está llena de fuerza simbólica... El alma que se escapa por sus dedos es para ellos la vida misma".

Oliver Sacks (Escritor y neurólogo británico).

Agradecimientos

Deseo dedicar el presente Trabajo de Final de Grado a todas las personas que me han aconsejado, apoyado, y por qué no decirlo, soportado, a lo largo del presente trabajo y durante todo el Grado Multimedia.

Sin ellos no hubiera sido posible. ¡Muchas gracias!

Índice

1. Introducción/Prefacio.....	12
2. Descripción/Definición/Hipótesis.....	14
3. Objetivos.....	16
3.1 Principales.....	16
3.2 Secundarios.....	16
4. Marco teórico/Escenario.....	17
5. Contenidos.....	19
5.1 Animation.....	19
5.2 Fonts.....	19
5.3 Materials.....	20
5.4 Objects.....	20
5.5 Prefabs.....	21
5.6 Scenes.....	21
5.7 Scripts.....	21
5.8 Sounds.....	22
5.9 Sprites.....	23
5.10 Texturas.....	23
6. Metodología.....	25
8. Plataforma de desarrollo.....	27
8.1 Hardware.....	27
8.2 Software.....	28
9. Planificación.....	30
9.1 Fases.....	30
9.2 Tareas.....	30
9.3 Diagrama de Gantt.....	31
10. Proceso de trabajo/desarrollo.....	32
10.1 Pasos previos.....	32
10.2 Creación y Modelado del Personaje.....	33
10.3 Creación y Modelado del 'Cupcake Car'.....	36
10.4 Creación y Modelado de los elementos del escenario.....	37
10.5 Creación del Logotipo del Juego.....	38
10.6 Interfaz Gráfica.....	38
10.7 Lenguaje de Signos y Flat Design.....	39
10.8 Accesibilidad: <i>Closed Caption</i>	41
10.9 Instalación Unity y Ubicación.....	42
10.10 Iluminación y Sombras.....	42
10.11 Cámaras.....	42
10.12 Creación del escenario.....	43
10.13 Mapa del juego.....	44
10.14 Entradas por teclado.....	45
10.15 Scripting.....	45
10.16 Físicas.....	47
10.17 Prefabs (prefabricados).....	48
10.18 Animación.....	48
10.19 Sonidos FX y Música del Juego.....	49
10.20 Optimización del juego.....	50
10.21 Resolución y publicación de proyecto.....	50
11. APIs utilizadas.....	52
13. Prototipos.....	53
13.1 Lo-Fi.....	53
13.2 Hi-Fi.....	56
14. Guiones.....	59
15. Perfiles de usuario.....	61
16. Usabilidad/UX y Accesibilidad.....	62
17. Seguridad.....	64
18. Tests.....	65
19. Versiones de la aplicación/servicio.....	81
19.1 Versión Alpha.....	81
19.1 Versión Beta.....	81
20. Requisitos de instalación/implantación/uso.....	83
21. Instrucciones de instalación/implantación.....	84

22. Instrucciones de uso.....	85
23. Bugs.....	86
24. Proyección a futuro.....	88
25. Presupuesto.....	89
26. Análisis de mercado.....	90
27. Marketing y Ventas.....	91
28. Conclusión/-es.....	92
Anexo 1. Entregables del proyecto.....	93
Anexo 2. Código fuente (extractos).....	94
Anexo 3. Librerías/Código externo utilizado.....	99
Anexo 4. Capturas de pantalla.....	101
Anexo 5. Guía de usuario.....	107
Anexo 6. Libro de estilo.....	108
Anexo 8. Encuesta Previa.....	114
Anexo 9. Game Design Document.....	119
1. Introducción.....	121
1.1 Concepto del juego.....	121
1.2 Características principales.....	121
1.3 Género.....	121
1.4 Propósito y público objetivo.....	122
1.5 Jugabilidad.....	122
1.6 Estilo visual.....	123
1.7 Alcance.....	123
2. Mecánicas del juego.....	124
2.1 Jugabilidad.....	124
2.2 Flujo de juego:.....	124
2.3 Personaje.....	125
2.4 Movimiento y físicas:.....	126
2.5 Controles:.....	126
3. Interfaz:.....	127
4.1 Diagrama Flujo:.....	127
4.2 Diagrama Mecánicas de Juego:.....	128
4.3 Menú principal.....	129
4.4 Créditos:.....	130
4.5 Help:.....	130
4.6 Final Juego:.....	131
4. Arte.....	132
4.7 Arte 2D.....	132
4.8 Arte 3D.....	132
4.9 Animación.....	133
4.10 Sonido.....	133
Anexo 10. Glosario/Índice analítico.....	134
Anexo 11. Bibliografía.....	135

Figuras y tablas

Índice de figuras

Figura 1: Hello Kitty de Samrio.....	13
Figura 2: Ilustración de <i>Toki Doki</i>	14
Figura 3: Libro: <i>I Love Kawaii</i>	14
Figura 4: Libro: <i>Molly & Charuca Kawaii, Love and Cakes</i>	15
Figura 5: Juego/pelicula: <i>Sugar Rush</i>	14
Figura 6: Mundos de <i>Army of Trolls</i> de Gary J Lucken.	15
Figura 7: Iredia: El Secreto de Atram.	17
Figura 8: Software E-mintza.....	17
Figura 9: <i>Assets</i> Proyecto	19
Figura 10: <i>Assets</i> : Animation	19
Figura 11: <i>Assets</i> : Fonts	20
Figura 12: <i>Assets</i> : Materiales	20
Figura 13: <i>Assets</i> : Objects.....	21
Figura 14: <i>Assets</i> : Prefabs.....	21
Figura 15: <i>Assets</i> : <i>Scenes</i>	21
Figura 16: <i>Assets Scripts</i>	22
Figura 17: <i>Assets Sounds</i>	22
Figura 18: <i>Assets: Sprites</i> Interfaz Gráfica.....	23
Figura 19: <i>Assets: Sprites</i> Lengua Signos	23
Figura 20: <i>Assets</i> : Texturas	24
Figura 21: <i>Assets</i> : Texturas Lenguaje Signos	24
Figura 22: Boceto Piruetto.	34
Figura 23: <i>Turnaround</i> Piruetto.....	34
Figura 24: Variaciones de color Piruetto.....	35
Figura 25: Vistas Piruetto en 3DS Max.	34
Figura 26: Piruetto con texturizado previo en 3DS Max.....	35
Figura 27: Textura cara Piruetto.	35
Figura 28: Textura cara piruleta de Piruetto.	35
Figura 29: Textura cuerpo Piruetto. 36.....	35
Figura 30: Modelado Cupcake en 3DS Max.	35
Figura 31: Aspecto final del Cupcake Car con Piruetto.....	36
Figura 32: Modelado piruletas en 3DS Max.	35
Figura 33: Modelado Chupa-Chups en 3DS Max.	37
Figura 34: Modelado galleta en 3DS Max.Figura	36
Figura 35: Modelado nube en 3DS Max.....	36
Figura 36: Creación Logotipo Piruetto's Race.....	38
Figura 37: Aspecto gráfico de los elementos de interfaz.....	39
Figura 38: Ejemplo de fotografías realizadas para el lenguaje de signos.	39
Figura 39: Prueba de pictograma Lenguaje de Signos.....	40

Figura 40: Prueba de pictogramas finalmente descartados.....	40
Figura 41: Tipografía <i>Dissaramas CRU Fingerspelling Font</i>	40
Figura 42: Tipografía original.....	40
Figura 43: Tipografía modificada.....	40
Figura 44: Pictograma posicionado en Unity.....	41
Figura 45: Símbolo del Sistema Closed Caption. No debe confundirse con Subtítulo o Teletexto.....	41
Figura 46: Elementos IU (Interfaz Gráfica) de Unity.....	42
Figura 47: Vista Cámara y vista Minicámara.....	43
Figura 48: Aspecto final montañas en Unity.....	43
Figura 49: <i>Heightmap</i>	43
Figura 50: <i>Uso de plano en lugar de terreno</i>	44
Figura 51: Vista circuito en Unity.....	44
Figura 52: Circuito modelado en 3DS Max.....	45
Figura 53: Componentes <i>Box Collider, WheelColliders y centro de masas del Cupcake Car</i>	48
Figura 54: Proceso grabación letras alfabeto con Audacity.....	49
Figura 55: Pantalla <i>Build Settings</i> de Unity.....	50
Figura 56: Datos encuesta Sistema Operativo empleado para jugar.....	51
Figura 57: <i>Wireframe</i> Menú principal.....	53
Figura 58: <i>Wireframe</i> Pantalla Juego.....	53
Figura 59: <i>Wireframe</i> pantalla créditos.....	54
Figura 60: <i>Wireframe</i> Pantalla <i>Help</i>	54
Figura 61: <i>Wireframe</i> <i>Game Over</i>	55
Figura 62: Prototipo HI-FI Menú Principal.....	56
Figura 63: Prototipo HI-Fi pantalla de juego.....	56
Figura 64: Prototipo HI-Fi pantalla de Créditos.....	57
Figura 65: Prototipo HI-Fi pantalla Help.....	57
Figura 66: Prototipo HI-Fi pantalla Game Over.....	58
Figura 67: Mapa de Navegación del Juego.....	62
Figura 68: Opciones de Closed Captioning.....	63
Figura 69: Aspecto Juego en fase Alpha. Personaje, Cupcake Car sobre plano con física realizada.....	81
Figura 70: Aspecto Juego en fase Beta.....	82
Figura 71: Reproductor Web de Unity (tecnología obsoleta).....	84
Figura 72: Tecnología WebGL.....	84
Figura 73: Ampliación Imagen Boceto Piruetto.....	101
Figura 74: Ampliación Imagen Vistas 3D Piruetto.....	101
Figura 75: Ampliación Imagen modelado Cupcake en 3DS Max.....	102
Figura 76: Ampliación imagen aspecto Final Piruetto incorporado en Unity.....	102
Figura 77: Ampliación imagen aspecto final Piruetto & Cupcake Car incorporados en Unity.....	103
Figura 78: Ampliación imagen circuito en 3DS Max.....	103
Figura 79: Ampliación imagen modelado piruletas en 3DS Max.....	104
Figura 80: Ampliación modelado Chupa-Chups.....	104
Figura 81: Ampliación modelado nubes de fresa y nata.....	105
Figura 82: Ampliación modelado de galletas en 3DS Max.....	105
Figura 83: Ampliación modelado chocolate en 3DS Max.....	106

Figura 84: Aspecto entorno de trabajo Unity.....106

Índice de tablas

Tabla 1: Fases Planificación 30
Tabla 2: Tareas Planificación..... 30
Tabla 3: Presupuesto. 89

1. Introducción

La aparición de los videojuegos en el mundo ha supuesto una revolución cuyas dimensiones constituyen el objeto de estudio de muchos investigadores. Su historia comienza en años 40, cuando aparecen las primeras supercomputadoras programables y continua con los videojuegos modernos de hoy en día. Actualmente, los videojuegos son un producto de consumo de masas y según el DEV¹, es una industria que seguirá en crecimiento a lo largo de los años.

Dicha industria ha presentado diversos enfoques hasta nuestros días: en sus orígenes, en los años 60 fue el entretenimiento y actualmente nos encontramos enfoques más pragmáticos como: educación, salud, publicidad, cine, etc. Por ello, la realización de este proyecto no se enmarcará única y exclusivamente en el desarrollo de un videojuego cuyo objeto sea el entretenimiento sino que también pretende aportar un valor añadido, situándose en el sector educativo.

El potencial de los videojuegos en el sector educativo puede verse reflejado en numerosos estudios, como por ejemplo, *Digital Games in Education. The design of GamesBased Learning Environments*², *Learning with Digital Games: A Practical Guide to Engaging Students in Higher Education*³; y *Good Video Games and Good Learning*⁴ (por citar algunos), que reflejan como los videojuegos constituyen un método ideal para el aprendizaje ya que hacen uso del constructivismo⁵, para situar al alumno en el centro de su experiencia de aprendizaje. *"Todos los aprendizajes más importantes de la vida se realizan jugando"*⁶.

Según dichos estudios, los videojuegos generan un entorno íntimo, atractivo y relajado a través de su ambientación y elementos gráficos. Potencian la inmersión en el aprendizaje sin la presión de los errores, otorgan una respuesta automática a ciertas acciones, aportan retroalimentación, permiten aprender de una forma autónoma, ayudan a establecer conexiones y sobre todo, trabajan la motivación intrínseca. Por este motivo, se ha tenido especial cuidado con la ambientación del proyecto, que hará uso de una estética de tipo *Cartoon* basada en el estilo *Kawaii*.

¹ DEV. Libro Blanco de Desarrollo de videojuegos presentado por la Asociación Española de Productoras y Desarrolladoras de Videojuegos y Software de Entretenimiento. [en línea] <http://www.dev.org.es/> [Fecha de consulta: 06/03/2016]

² Gros, B. (2007). *Digital games in education. The design of games-based learning environments*. *Journal of Research on Technology in Education*.

³ Whitton, N. (2010). *Learning with digital games: A practical guide to engaging students in higher education*. New York, NY.

⁴ Gee, J. P. (2005). *Good video games and good learning*. *Phi Kappa Phi Forum*.

⁵ Constructivismo. Wikipedia. Constructivismo [en línea] [Fecha de consulta: 06/03/2016]

⁶ [https://es.wikipedia.org/wiki/Constructivismo_\(pedagog%C3%ADa\)](https://es.wikipedia.org/wiki/Constructivismo_(pedagog%C3%ADa))

⁶ Cita de Francesco Tonucci (pensador psicopedagogo italiano).

El fenómeno *kawai*⁷, proviene de la cultura japonesa y apareció por primera vez en 1960 con la aparición de juguetes de peluche con forma de animal. Se basa en la reducción de las formas y exageración de los elementos restantes. Sus formas son redondas y amables, los colores son cálidos y los personajes son simples, con el fin de evocar características infantiles y sentimientos de ternura. Un icono de referencia de este estilo es: *Hello Kitty* de Samrio⁸.



Figura 1: Hello Kitty de Samrio

El compendio de todo lo expresado anteriormente sirve de referente para conceptualizar y establecer los cimientos del que pretende ser un videojuego sensible y simpático. Un proyecto dirigido a todo tipo de usuarios interesados en el Aprendizaje de la Lengua de Signos o Alfabeto Dactilológico Español, pero especialmente dirigido al colectivo infantil. También pretende aportar una práctica de aprendizaje creativo.

La Lengua de Signos es la lengua natural de las personas sordas. En cada país surge una lengua de signos del mismo modo que surgen las lenguas orales, por ello, en España se utiliza la Lengua de Signos Española y en Cataluña, la Lengua de Signos Catalana. Las lenguas de signos poseen todas las características de las lenguas orales y la misma funcionalidad para pensar, comunicar y manejar conceptos. (Fersocam, 2011)⁹.

Por tratarse de un proyecto de concienciación, sensible y social, orientado tanto a personas con problemas de audición como a las que no padecen, se incluye también opciones de accesibilidad, concretamente el *Closed Caption*, que subtitula todo aquello que se escucha en el juego: desde diálogos hasta sonidos. La accesibilidad más fácil de implementar es aquella destinada a personas sordas total y/o parcial, ya que la base del videojuego es visual.

Para finalizar, destacar que para la realización del proyecto, se ha elegido el desarrollo de un videojuego por ser una actividad multidisciplinaria, con una gran plasticidad, donde se trabajan multitud de competencias adquiridas en el Grado Multimedia: programación, diseño gráfico, modelado 3D, animación, sonido, música etc. "*Los videojuegos presentan un gran potencial en cuanto a la adquisición de competencias digitales*" (Gros, 2007, 2009).

⁷ Wikipedia. Kawaii. <https://es.wikipedia.org/wiki/Kawaii>. [en línea] [Fecha de consulta: 06/03/2016]

⁸ Samrio. Compañía, según Wikipedia, en 2008 obtuvo 500 millones(\$) de beneficios y posicionamiento de 50.000 productos en más de 60 países. <http://www.sanrio.com/characters-hello-kitty/>

⁹ Fersocam.(2011) Confederación Estatal de Personas Sordas

2. Descripción

Del mismo modo que el ser humano está hecho de historias, los videojuegos también tratan de historias. Historias que se van desenredando lentamente a través del personaje principal, de su antagonista y de sus objetivos. Los juegos son herramientas para contar historias que hacen referencia a la narrativa. Por ello, el presente Trabajo de Final de Grado se enmarca dentro de la especialidad de Narrativas Visuales 2D/3D.

Dicho proyecto consiste en el desarrollo de un videojuego educativo en 3D, con estética tipo *Cartoon*, estilo *Kawaii*, destinado al aprendizaje del lenguaje de signos. El motor gráfico de videojuegos utilizado será *Unity* que trabaja bajo los lenguajes de programación: *C#*, *JavaScript* y *Boo*¹⁰. No obstante, para el proyecto se ha elegido *C#*¹¹ (*C Sharp*), ya que es un lenguaje orientado a objetos muy potente (superior a *C/C++*).

Unity contiene algunos objetos predefinidos llamados *GameObjects*¹² que presentan algunas formas básicas, pero no puede modelar objetos. Se necesita un programa de modelado 3D externo para crear los elementos necesarios del juego. Debido a ello, el modelado de los objetos se realizará con *3DS Max* y posteriormente se importará a *Unity* en formato *FBX*¹³. Este apartado engloba el diseño del personaje principal, un coche y los elementos del escenario.

La historia del juego o narrativa está basada en Pirueto (el personaje principal) que le encanta comer piruletas de fresa, y se moverá montado en un coche especial (un Cupcake), por un circuito repleto de piruletas. La vista será en perspectiva tercera persona. Durante el juego, se piden aleatoriamente letras del alfabeto tradicional y el objetivo es colisionar con aquellas piruletas que contengan la letra que se pide, pero éstas contienen imágenes del alfabeto dactilológico.

El proyecto también comprende una parte de sonido, donde se han realizado grabaciones voz de todas las letras del abecedario para su posterior uso en el juego. Por otro lado, comprende la elección y adaptación de los pictogramas necesarios de la Lengua de Signos que más favorecen su visualización dentro del juego y para usarse en las piruletas, que son el objeto principal del juego.

¹⁰ **Boo**. Es lenguaje de programación orientado a objetos, de tipos estáticos para la Common Language Infrastructure con una sintaxis inspirada en Python. Wikipedia [https://es.wikipedia.org/wiki/Boo_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Boo_(lenguaje_de_programaci%C3%B3n))

¹¹ **C#**. El nombre C Sharp fue inspirado por la notación musical, donde '#' (sostenido, en inglés sharp) indica que la nota (C es la nota do en inglés) es un semitono más alta, sugiriendo que C# es superior a C/C++. Además, el signo '#' se compone de cuatro signos '+' pegados. Wikipedia. https://es.wikipedia.org/wiki/C_Sharp.

¹² **Unity Documentation**. GameObjects. <http://docs.unity3d.com/es/current/Manual/GameObjects.html> [en línea] [Fecha de consulta: 06/03/2016]

¹³ **FBX**. Formato de archivo 3D independiente y gratuito, que proporciona acceso al contenido de cualquier software.

Las plataformas de publicación son las siguientes: Una versión para *Windows* .exe, una versión para *Mac OSx* .dmg (resoluciones pantalla: 1280x720), y por último una versión Web (resolución pantalla: 960x600). *Unity*, compila todo el contenido una sola vez y exporta a todas las plataformas elegidas: web, escritorio, consola y Mobile, entre las principales, ya que es una herramienta multiplataforma.

Y por último, el proceso creativo del proyecto es el siguiente: el diseño del personaje, el coche y la estética del proyecto, toman influencia de la ilustradora *Kawaii*, Charuca¹⁴. Los libros de esta autora, son fuente de inspiración. Otra referencia para el proyecto es la marca Toki Doki¹⁵ inspirada en personajes estilo japonés con diferentes temáticas. El diseño del escenario está formado por todo tipo de caramelos y dulces, inspirado en el juego/película *Sugar Rush*¹⁶. La creación de mundos también se ha inspirado en el portfolio del juego *Army of Trolls* de Gary J Lucken¹⁷.



Figura 2: Ilustración de Toki Doki



Figura 3: Libro: I Love Kawaii,



Figura 4: Libro: Molly & Charuca Kawaii, Love and Cakes



Figura 5: Juego/película: Sugar Rush.

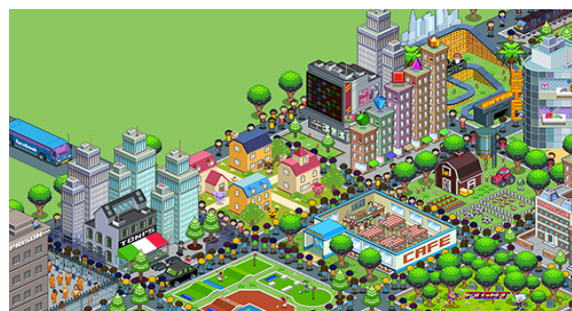


Figura 6: Mundos de Army of Trolls de Gary J Lucken.

¹⁴ Charuca. Illustrator and Character designer.. <http://www.charucashop.com/>

¹⁵ Toki Doki. Marca y estilo de vida de inspiración japonesa creada en 2005 por el artista italiano Simone Legno, Pooneh Mohajer e Ivan Arnold. Son productores de prendas de vestir, calzado, accesorios y otros productos que utilizan el arte y personajes icónicos. <http://www.tokidoki.it/>

¹⁶ Sugar Rush. Es un lugar ficticio donde empieza la aparición de Vanellope von Schweetz de la película de animación en 3D de Disney, Ralph el Demoledor. Es un reino de dulces y caramelos y donde sucede el videojuego arcade de carreras Sugar Rush Speedway. http://ralphdemoledor.wikia.com/wiki/Sugar_Rush

¹⁷ Army of Trolls de Gary J Lucken. Portfolio: <http://www.armyoftrolls.co.uk/website/html/portfolio.html>

3. Objetivos

3.1 Principales

- Desarrollo de un videojuego que cumpla una función educativa y/o social pero también aporte entretenimiento y diversión.
- Concienciar y normalizar el uso la Lengua de Signos entre todos los usuarios, ya que son pocas las iniciativas de productos y servicios que tienen en cuenta esta Lengua.
- Cumplir el objetivo para el que está diseñado: el aprendizaje del lenguaje de signos.
- Desarrollo de un producto con una estética *Kawaii* y un aspecto gráfico simpático, creativo y original.

3.2 Secundarios

- Investigar si es posible su publicación para dispositivos móviles.
- Superación de un reto personal, ya que prácticamente se desconocen todas las herramientas y ámbitos con los que se ha de trabajar para desarrollar el proyecto.
- Aprendizaje del programa Unity 3D.

4. Marco teórico y Escenario

En España, el desarrollo de videojuegos evoluciona a medida que lo hace el mercado internacional. Madrid y Barcelona son ciudades emergentes con una gran proyección de futuro. Por citar algunas cifras, en 2017 se esperan unos 723 millones de euros de facturación de empresas nacionales para este sector y el 78% proviene de la distribución online. (Wikipedia).

A pesar de estas cifras, no es tanta la industria de videojuegos que se centra en proyectos sensibles, accesibles y/o para personas discapacitadas, aunque en los últimos años el tema se está empezando a desarrollar. Lo podemos ver en proyectos como: "Iredia: El Secreto de Atram"¹⁸, reconocido como uno de los 50 mejores videojuegos *Indies*¹⁹ para la plataforma *Xbox Live*. El juego presenta un mundo en el que el sonido varía de diferentes formas para que el jugador pueda experimentar y aprender sobre los diferentes tipos de sordera.

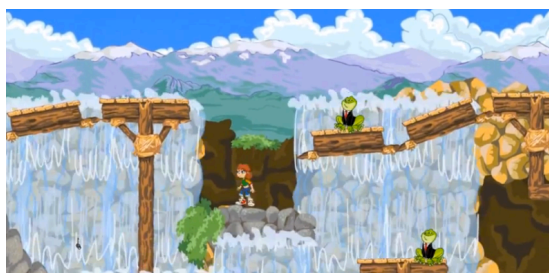


Figura 7: Iredia: El Secreto de Atram.

Otro proyecto sensible que podemos ver en el mercado español, es el software E-mintza²⁰ un programa de comunicación aumentativa desarrollado por la *Fundación Orange* y la *Fundación Dr. Carlos Elósegui de la Policlínica Gipuzkoa*, dirigido a personas con autismo o con barreras de comunicación escritas.



Figura 8: Software E-mintza.

¹⁸ **Iredia:** El Secreto de Atram. <http://www.iredia.es/iredia.html> [en línea] [Fecha de consulta: 06/03/2016]

¹⁹ **Indie.** Videojuego Independiente: https://es.wikipedia.org/wiki/Videojuego_independiente [en línea] [Fecha de consulta: 06/04/2016]

²⁰ **E-mintza.** <https://play.google.com/store/apps/details?id=air.com.orange.emintza&hl=es> [en línea] [Fecha de consulta: 06/04/2016]

Es un programa donde a través de pictogramas y sonidos se facilita la interacción del usuario trabajando con una pantalla táctil y es totalmente personalizable. Está apoyado por el plan "Avanza2" del Ministerio de Industria, Turismo y Comercio (lo que avala la calidad del proyecto).

Para la realización del proyecto se ha tomado como referencia el funcionamiento de dicho programa para adaptarlo al aprendizaje del alfabeto dactilológico, además de presentarlo con una interfaz gráfica más desarrollada, creativa y divertida.

5. Contenidos

El proyecto abarca desde la concepción de la idea inicial hasta el desarrollo del producto del final. Todos los elementos que componen la realización del videojuego (*Assets*) son de elaboración propia. El proyecto presenta la siguiente estructura en Unity:

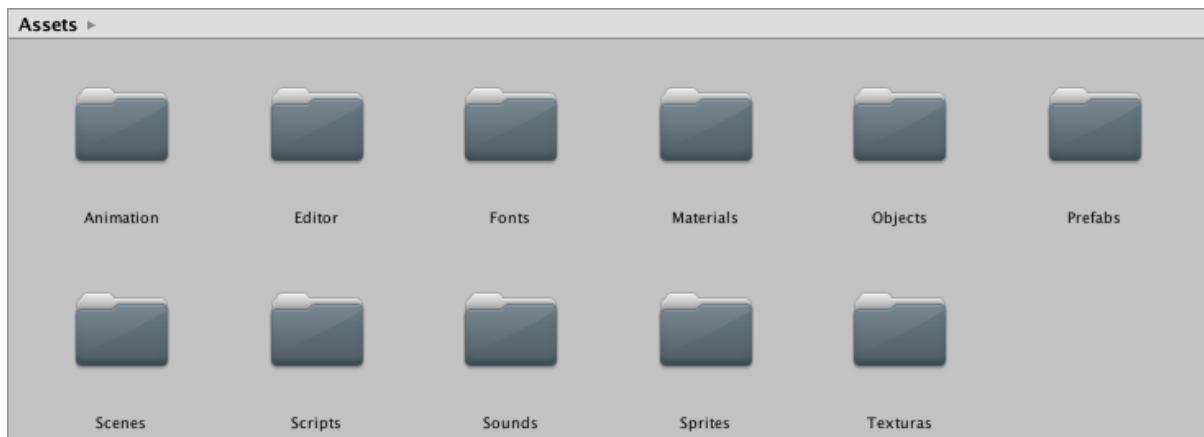


Figura 9: Assets Proyecto

5.1 Animation

Se han elaborado las siguientes animaciones dentro del software Unity:



Figura 10: Assets: Animation

5.2 Fonts

Se han importado 2 fuentes externas elaboradas por terceros, para conseguir el aspecto infantil deseado:

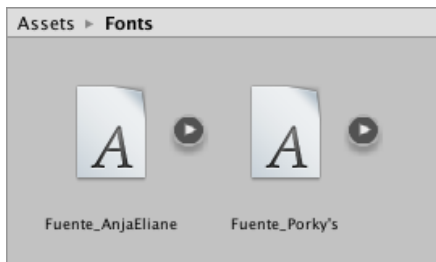


Figura 11: Assets: Fonts

5.3 Materials

Se han realizado los siguientes materiales dentro del software Unity:

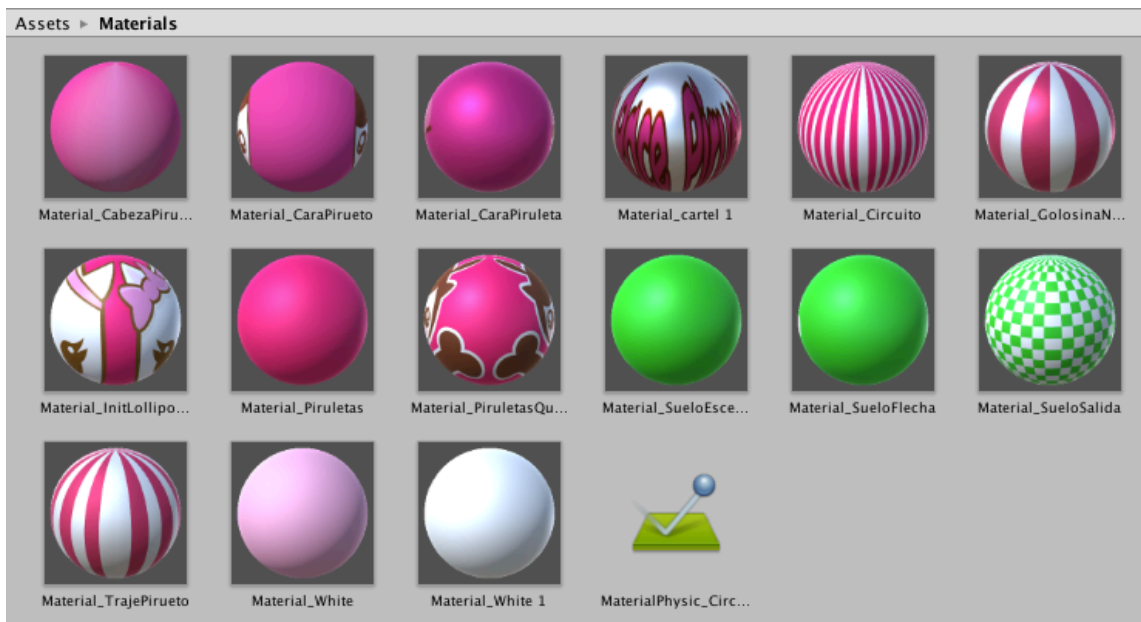


Figura 12: Assets: Materiales

5.4 Objects

Se han modelado los siguientes objetos con el software Autodesk 3DS Max y exportado en formato .FBX:

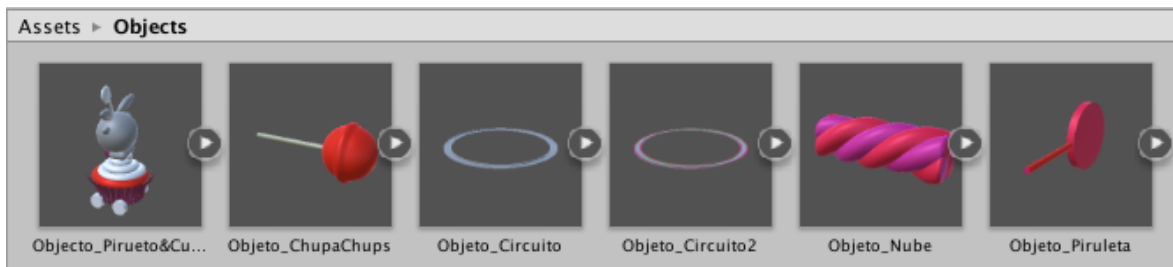


Figura 13: Assets: Objects

5.5 Prefabs

Se han creado los siguientes *prefabs*, ya que son elementos repetitivos, a partir de los objetos anteriores:

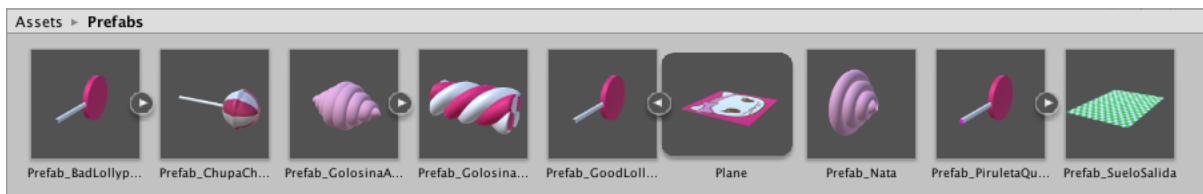


Figura 14: Assets: Prefabs

5.6 Scenes

Se ha realizado 2 escenas en Unity. Una para controlar el menú principal de juego y otra que contiene el desarrollo del juego.

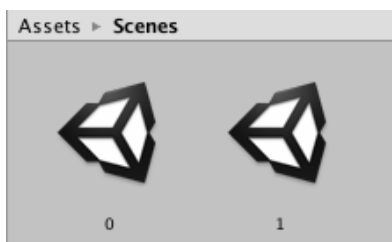


Figura 15: Assets: Scenes

5.7 Scripts

Se han realizado los siguientes scripts:

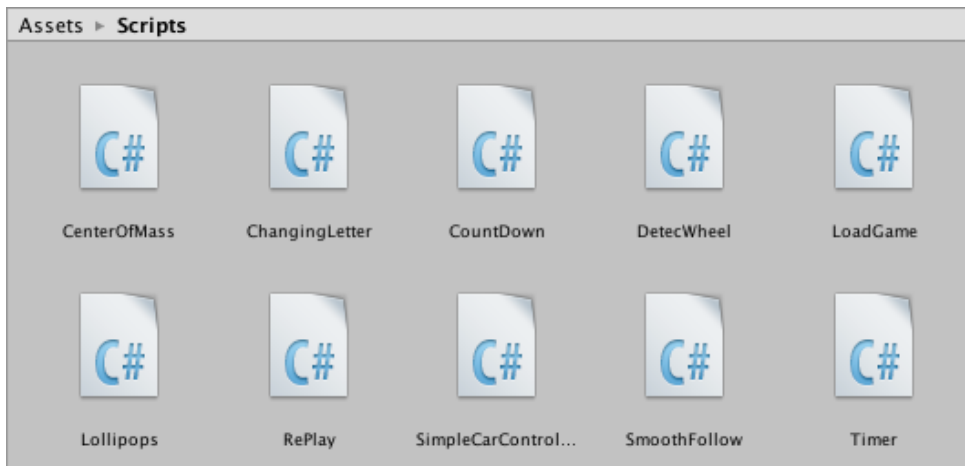


Figura 16: Assets Scripts

5.8 Sounds

El proyecto se compone de los siguientes sonidos:

- **Sonidos de elaboración propia:** Ha sido necesario grabar los sonidos de las letras del alfabeto. Formato .OGG ya que presenta un alto grado de compresión.
- **Sonidos de terceros:** Para la música y efectos especiales del juego. Formato .WAV, mp3.

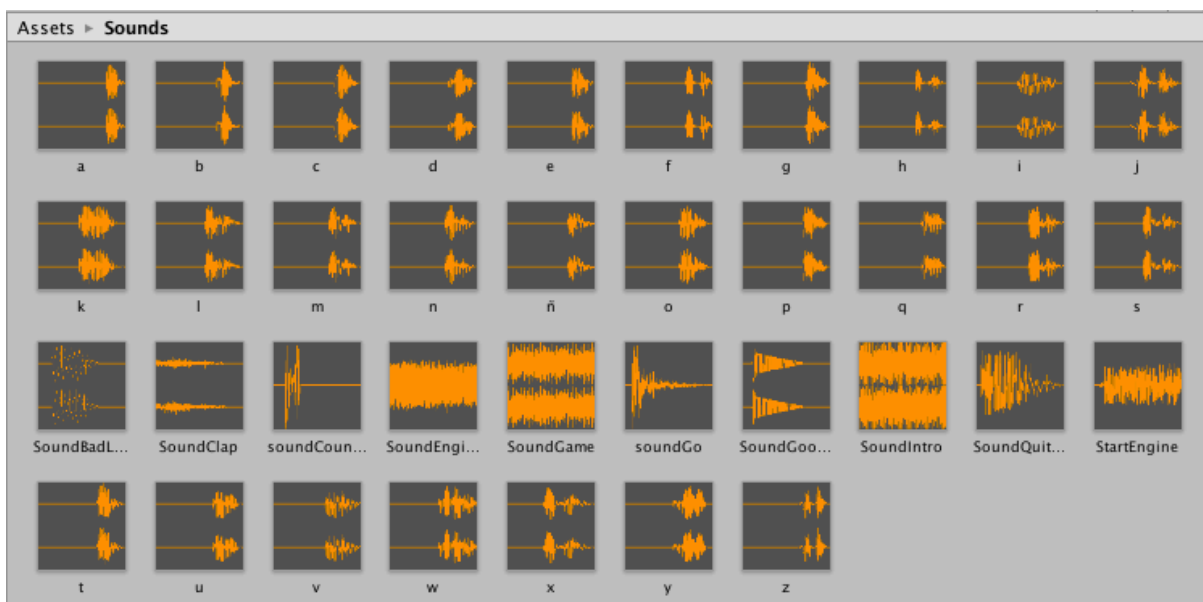


Figura 17: Assets Sounds

5.9 Sprites

Unity diferencia entre *Sprites* y texturas. Los *sprites* son elementos gráficos que no se utilizan como texturas, como por ejemplo, los elementos de la interfaz gráfica. Se han elaborado los siguientes *sprites* con Adobe *Illustrator* y *Photoshop*:

— Elementos de la interfaz gráfica:



Figura 18: Assets: Sprites Interfaz Gráfica

— Las letras del alfabeto dactilológico que aparecen en el panel de control cuando se pulsa la tecla ayuda (versión reducida 512x512):

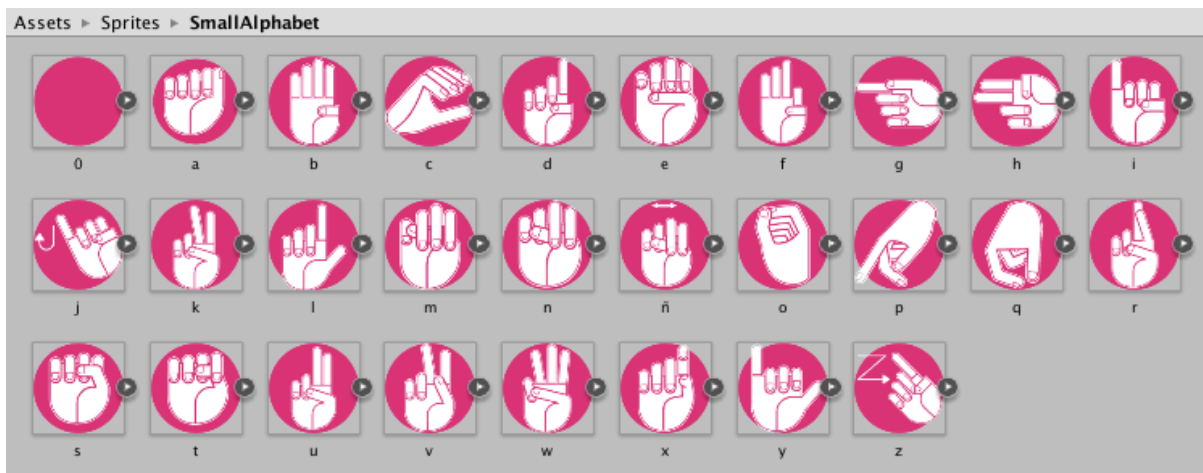


Figura 19: Assets: Sprites Lengua Signos

5.10 Texturas

Se han realizado todas las texturas que conforman el juego (*Adobe Illustrator*). Posteriormente se han utilizado para crear los materiales dentro del software Unity.

— Texturas del personaje y elementos del escenario:

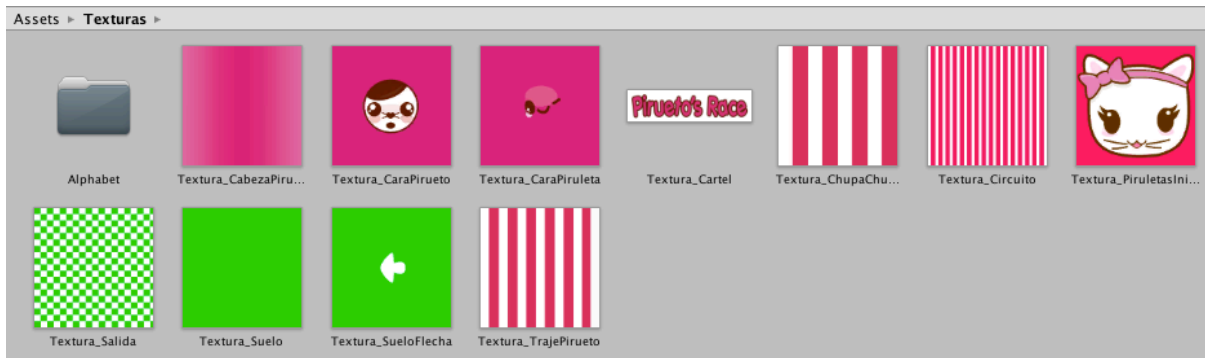


Figura 20: Assets: Texturas

— Texturas que van sobre las piruletas, para detallar los signos del alfabeto dactilológico que componen el objetivo del juego (versión ampliada 1024x1024) :

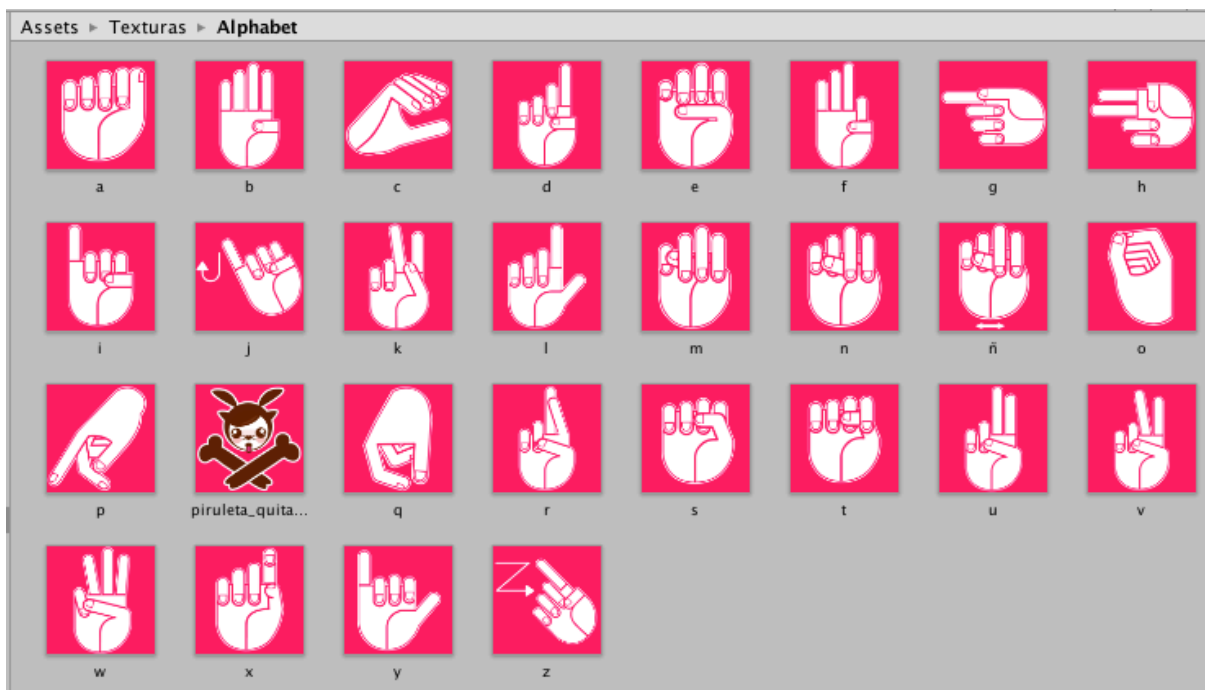


Figura 21: Assets: Texturas Lenguaje Signos

6. Metodología

Normalmente, en el proceso de creación de un videojuego se ven implicados diversos tipos de profesionales: animadores, programadores, diseñadores gráficos, músicos, etc.

Debido a esto, en el desarrollo de videojuegos se utilizan diversas metodologías ágiles en cuanto a la planificación del trabajo multidisciplinar, como por ejemplo:

- Waterfall process: En dicha metodología hasta que una fase no está terminada no se pasa a la siguiente, su principal inconveniente es que ante cualquier error se pierde excesivo tiempo.
- Rational Unified Process: Es una metodología iterativa y adaptable pero es muy pesada y hay mucho tiempo de documentación de los procesos.
- Essential Unified Process: Es una mejora de la anterior.
- OpenUp: es una metodología que solo incluye los pasos estrictamente necesarios y suficientes.
- Team Software Process: Se encarga de controlar que hay que hacer en cada fase del desarrollo del producto y las conecta entre sí. Para proyectos a gran escala.
- Microsoft Solution Framework: Es una infraestructura para realizar proyectos Microsoft.
- Scrum Framework: tecnología iterativa incremental muy utilizada en el desarrollo de videojuegos, solapa varias fases del desarrollo y presenta una gran flexibilidad a los cambios.

En este caso puesto que no se trabaja con un equipo multidisciplinar, no se ha visto necesario adoptar ninguna de las metodologías de planificación anteriores.

En cuanto a la metodología de trabajo del proyecto, en un primer término ha sido necesario realizar un proceso de documentación y aprendizaje previo de las siguientes temáticas:

- Lenguaje de Signos.
- Tecnologías TIC en el ámbito de aprendizaje de la lengua de signos
- Consulta del Manual, *Scripting* API, libros y tutoriales de Unity para el aprendizaje de programación en C#.
- Inventario visual sobre diversas estéticas de juegos y dibujos que podrían ser fuente de inspiración.
- Investigación de la técnica de dibujo *Kawaii*.
- Tutoriales de modelado 3D para el programa 3DS Max para realizar los diversos elementos del juego.
- Realización de un mini curso de Diseño de Art Toy 2D para diseñar el personaje principal del juego en 2D.

A partir de aquí, se trabaja con *placeholders* (objetos substitutivos) para realizar todas las pruebas que se realicen en Unity y especialmente la parte de programación hasta obtener un prototipo funcional del producto. Gradualmente, los *placeholders* se substituyen por los modelos definitivos.

En cuanto a las herramientas y tecnologías utilizadas en el proyecto son las siguientes:

- Unity Versión 5.3.2 Personal (Unity Free) con acuerdo de Licencia EULA, como motor gráfico del videojuego.
- Editor *Monodevelop* de Unity. Para la edición del código.
- *C# script*. C Sharp Script para el desarrollo de la programación del juego.
- Adobe *Illustrator* CS6 para la elaboración de la Ilustración, texturas y dibujos en 2D teniendo en cuenta la resolución y formato final.
- Autodesk 3DS Max para la animación y modelado 3D del personaje y objetos del escenario. Se exportaran en formato .FBX para su posterior importación a Unity.
- Adobe *Audition* CS6 y/o Audacity para la grabación de los sonidos del abecedario. Y manipulación del sonido y música del juego de terceros.

8. Plataforma de desarrollo

8.1 Hardware

MacBook Pro 15 pulgadas, principios de 2011:

- Sistema Operativo: OS X Versión 10.9.5.
- Procesador: 2GHz Intel Core i7.
- Memoria: 8GB 1333 MHz DDR3
- Tarjeta Gráfica: AMD Radeon HD 6490M 256 MG
- Sistema Operativo: OS X 10.9.5
- Disco Duro: 1GB.
- Partición *BootCamp* de 400MB y sistema operativo *Windows 7 Home Basic Service Pack1 64 bits* para la utilización de *3DS Max*.

Micrófono *Blue Yeti Gris USB*: Grabación de audio realizada en modo *Cardioid*, que es la mejor opción para grabaciones de voz, ya que el sonido llega directamente de frente al micrófono, mientras que ignora la parte trasera y laterales. Características:

- Frecuencia de muestreo: 48 kHz.
- Tasa de bits: 16 bits.
- Patrones polares: Modo Stereo, Cardioid, Onmnidireccional y Bidireccional.
- Respuesta de frecuencia 20Hz-20kHz.
- Peso 550g (micrófono) y 1 Kg (soporte).
- Fabricante: Blue Microphones.

Cámara Fotográfica Reflex Canon EOS 450E: Para las pruebas de las fotografías del lenguaje de signos. Características:

- Sensor CMOS de 12,2 Megapíxeles.
- Disparos en serie a 3,5 fps.
- AF de área amplia en 9 puntos.
- Compatible con los objetivos EF/EF-S y los flashes Speedlite EX.

8. 2 Software

Game Engine Multiplataforma Unity Personal versión 5.3.2 (Free) Acuerdo de Licencia EULA. Que incluye:

- *Tecnología OpenGL*: Motor gráfico empleado por Unity para el sistema operativo OS X.
- Motor de física: *PhysX* (para realizar el movimiento del coche).
- Plugin Web Unity para el desarrollo de la versión web.
- *MonoDevelop*: Depuración y edición de Scripts.
- *Mecanim*: tecnología de animación de Unity para realizar el movimiento fluido y natural del personaje y otros elementos.
- *Unity Asset Server*- para el control de versiones de los *assets* del juego y scripts.
- Sistema de audio construido con la biblioteca FMOD, para reproducir audio comprimido .Ogg (códec Theora).

Lenguaje C# (Sharp): Lenguaje de programación empleado para la realización de los Scripts de Unity. Características:

- Orientado a objetos y componentes.
- Representa una mejora respecto a C y C++.
- Incorpora las mejoras de JAVA y la flexibilidad de C++.
- Forma parte de la plataforma .NET pero es un lenguaje independiente.

Audacity 2.1.2 con Licencia GNU General Public License: Para la grabación, edición y compresión de sonido se ha utilizado. Características:

- Software de Grabación y Edición de vídeo
- Permite grabación desde el micrófono, entrada de línea, dispositivos USB y Firewire.
- Permite grabaciones multi-pista.
- Frecuencias de muestreo de hasta 192.000 Hz .
- Grabación a una profundidad de 24 bits en Windows (el uso de Windows anfitrión WASAPI), Mac OS X o Linux (usando ALSA o host JACK).
- Grabación de múltiples canales a la vez.

Adobe Suite CS6 64 bits, con Licencia de Estudiante proporcionado por la Universitat Oberta de Catalunya, e incluye:

- *Adobe Illustrator CS6* versión 16.0.0: Realización de la ilustración 2D (texturas, sprites, logos e Interfaz de usuario y diseño de personaje)
- *Adobe Photoshop CS6* versión 13.0: Edición de imágenes.
- *Adobe After Effects CS6 versión 11.0.4.2: Composición de video.*
- *Adobe Premiere CS6 versión 6.0.0. Edición de vídeo.*

Autodesk 3DS Max versión 2014 64 bits, Licencia de Estudiante. Software de renderización, animación y modelado 3D. Modelado de personaje y elementos de escenario. Características:

- Animación 3D
- Texturización y modelado 3D.
- Renderizado 3D.
- Dinámica y Efectos.
- Incluye Interfaz de usuario y entorno de producción.
- Scripts de Python.

Balsamiq Mockups versión 2.2.16 de prueba. Software de prototipado para la realización de los Prototipos, bocetos y *Mockups* del proyecto. Características:

- *Wireframes* de baja fidelidad que permite crear la funcionalidad
- Arrastrar y soltar: para crear la interfaz de usuario.
- *Quick Add for Speed*: construcción de *wireframes* utilizando el teclado.
- Símbolos reutilizables: Permite creación de plantillas.
- *IU* Componentes e Iconos: Incorpora controles de usuario de interfaz e iconos.
- *Click-Through* Prototipos: Generación de prototipos de clics para demostraciones.
- Exportación a PNG o PDF.

9. Planificación

9.1 Fases

FASES	INICIO	FIN	DÍAS
Pacs	24/02/2016	08/05/2016	75
Concepción	24/02/2016	08/03/2016	14
Diseño	09/03/2016	30/03/2016	22
Producción	31/03/2016	04/05/2016	35
Pruebas	05/05/2016	10/05/2016	6
Entrega	11/05/2016	20/06/2016	41

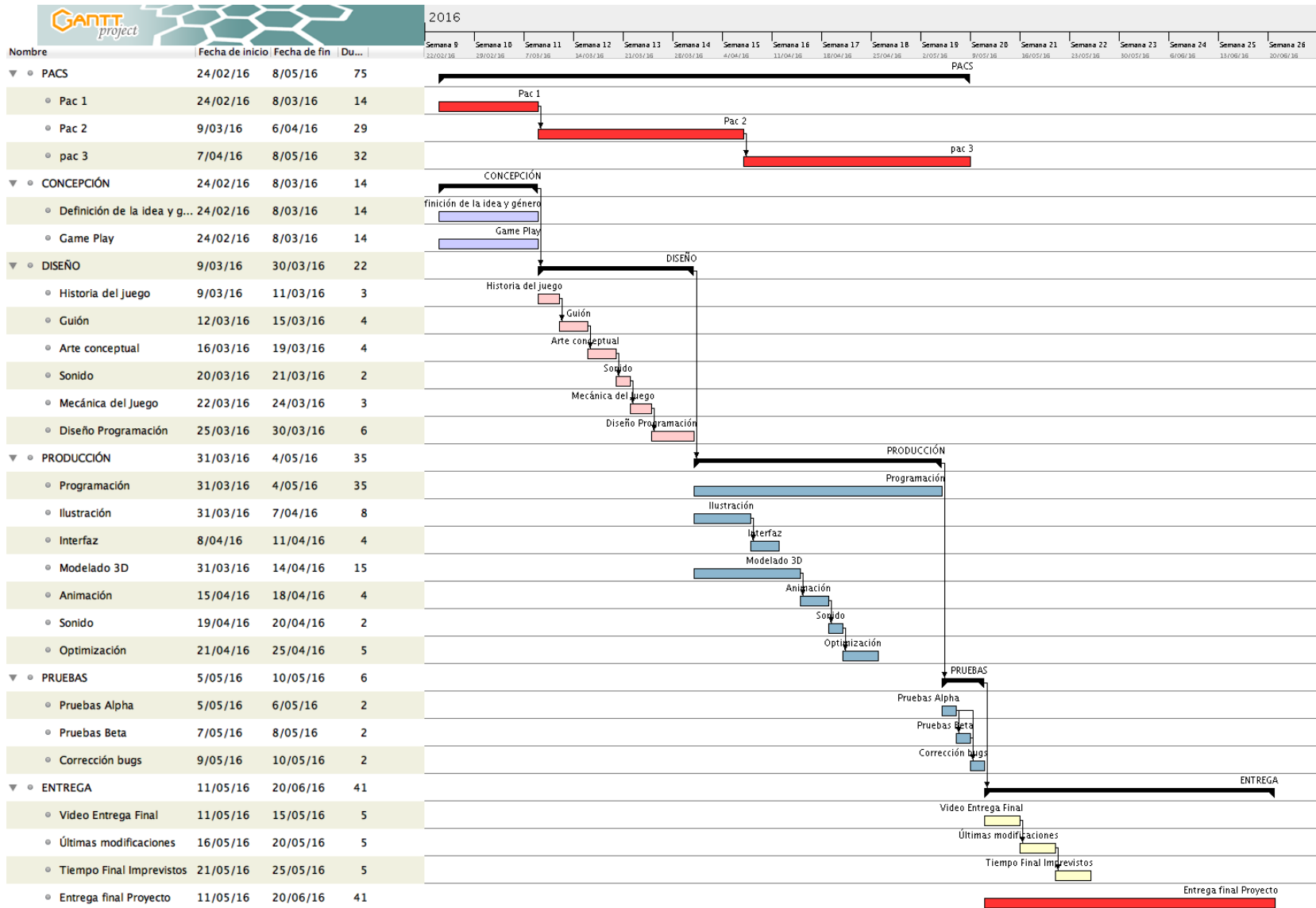
Tabla 1: Fases Planificación

9.2 Tareas

	TAREAS	INICIO	FIN	DÍAS
PACS	Pac 1	24/02/2016	08/03/2016	14
	Pac 2	09/03/2016	06/04/2016	29
	Pac 3	07/04/2016	08/05/2016	32
CONCEPCIÓN	Definición de la idea y género	24/02/2016	08/03/2016	14
	<i>Game Play</i>	24/02/2016	08/03/2016	14
DISEÑO	Historia del juego	09/03/2016	11/03/2016	3
	Guión	12/03/2016	15/03/2016	4
	Arte conceptual	16/03/2016	19/03/2016	4
	Sonido	20/03/2016	21/03/2016	2
	Mecánica del Juego	22/03/2016	24/03/2016	3
	Diseño programación	25/03/2016	30/03/2016	6
	PRODUCCIÓN	Programación	31/03/2016	04/05/2016
	Ilustración	31/03/2016	07/04/2016	8
	Interfaz	08/04/2016	11/04/2016	4
	Modelado 3D	31/03/2016	14/04/2016	15
	Sonido	19/04/2016	20/04/2016	2
	Animación	15/04/2016	18/04/2016	4
	Optimización	21/04/2016	25/04/2016	5
PRUEBAS	Pruebas <i>Alpha</i>	05/05/2016	06/05/2016	2
	Pruebas <i>Beta</i>	07/05/2016	08/05/2016	2
	Corrección <i>bugs</i>	09/05/2016	10/05/2016	2
ENTREGA	Video Entrega Final	11/05/2016	15/05/2016	5
	Últimas modificaciones	16/05/2016	20/05/2016	5
	Tiempo margen para imprevistos	21/05/2016	25/05/2016	5
	Entrega final del Proyecto	11/05/2016	20/06/2016	41

Tabla 2: Tareas Planificación

9.3 Diagrama de Gantt



10. Proceso de trabajo

10.1 Pasos previos

Primeramente se realiza un pequeño estudio online, elaborando encuestas a 80 personas, tanto del ámbito Multimedia como perfiles ajenos a él. El motivo de este estudio es poder establecer una mínima base informativa en cuanto a la orientación del proyecto en aspectos como:

- Porcentaje de gente que juega de forma habitual.
- Rango de edad y género tanto de los perfiles que juegan habitualmente como de los que no juegan.
- Dispositivos utilizados para jugar.
- Sistema Operativo.
- Conocer si los usuarios están interesados en juegos en Realidad Virtual a través del móvil.
- Conocer si los usuarios piensan que aprendizaje a través de los juegos es una buena idea.
- Conocer la opinión de los usuarios en cuanto si creen que existen el mercado suficientes juegos enfocados a personas que padecen alguna discapacidad.

Los resultados son concluyentes en cuanto a las plataformas de publicación del proyecto: Dispositivos Mobile y Windows como sistema operativo de sobremesa, son las plataformas que más despuntan. El formato web para jugar le gusta a la mitad de los usuarios. Y para finalizar, la mitad de los encuestados piensan que hay pocos juegos enfocados a personas con discapacidades, y sorprende que la otra mitad desconoce el tema.

Por ello, la finalidad de este proyecto además de la función didáctica, es normalizar y extender el uso del lenguaje de signos entre todos los usuarios y concienciar a la sociedad para que se ofrezca tal y como se ofrecen diferentes idiomas en demás productos, servicios y configuraciones.

La encuesta íntegra y la totalidad de sus resultados pueden consultarse en la sección de Anexos de la presente Memoria.

10. 2 Creación y Modelado del Personaje

10. 2. 1 Art Toy

El diseño del personaje se realiza pensando en una posible producción como juguete de coleccionista de edición limitada o "Art Toy"²¹. La producción se podría realizar con materiales como el vinil, ABS, madera, resina o *paper toys*. En los últimos años, con el desarrollo de las impresoras 3D, también podría plantearse la creación del Art Toy a través de las mismas.

Se realiza y se considera como Art Toy ya que en una etapa posterior, dicho elemento puede ser una extensión o complemento del juego. También se elabora una versión femenina del personaje.

10. 2. 2 Estilo Kawaii

Para la realización del personaje, se elige un estilo Kawaii, que como se ha comentado en otros apartados es un estilo minimalista basado en la reducción de los elementos a lo básico y a exagerar las proporciones. Se pretende elaborar un personaje dulce, que evoque sentimientos de ternura ('cute').

Los rasgos característicos del estilo Kawaii que se han utilizado para la elaboración del personaje son los siguientes:

- Cabeza grande y redonda, incluso es más grande que el cuerpo (exageración de las proporciones)
- Rasgos dulces y tierno.
- Grandes ojos separados.
- Nariz emplazada entre los dos ojos.
- Boca pequeña.
- La frente ocupa la mitad de la cara.
- Trazo grueso.
- Eliminación de brazos y piernas (reducción y/o eliminación de elementos para exagerar otros).

²¹ Art.Toy. Wikipedia. https://es.wikipedia.org/wiki/Art_toys [en línea] [Fecha de consulta: 06/03/2016]

Se trata de un personaje muy simple, pero al mismo tiempo es necesario transmitir una historia a través de él. Por ello, se utiliza otro minipersonaje (una minipiruleta) que forma parte de la estructura de su cuerpo para contar que le encantan las piruletas. Y así mismo, esta piruleta es otro personaje independiente.

10. 2. 3 Boceto y Slogan

Se realiza un dibujo a mano alzada, para su posterior escaneo y vectorización en Adobe *Illustrator*.

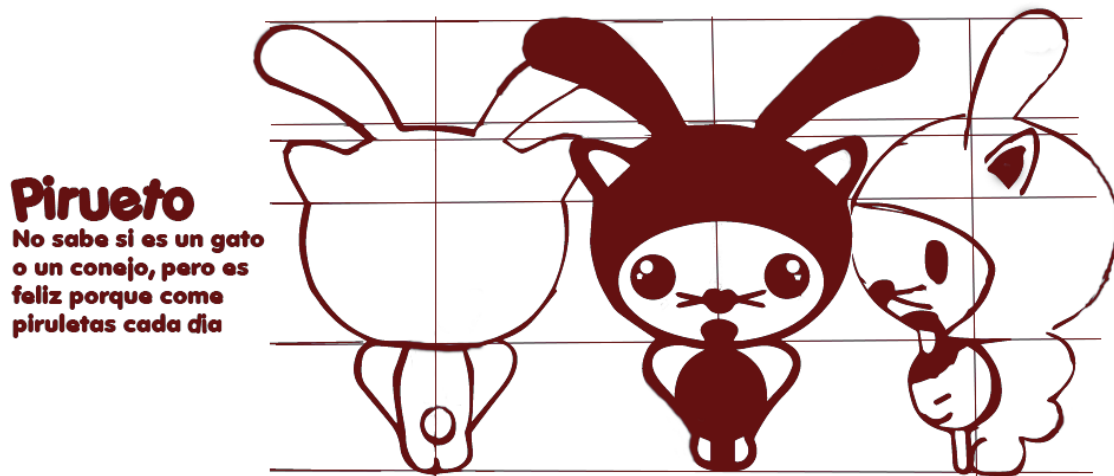


Figura 22: Boceto Piruetto.

10. 2. 4 TurnAround

A partir del boceto, se realizan las diferentes vistas del personaje, también se incluye una versión femenina del personaje.



Figura 23: Turnaround Piruetto.

10. 2. 5 Variaciones de color



Figura 24: Variaciones de color Piruetto.

10. 2. 6 Transformación de 2D a 3D

A partir del *TurnAround*, se realiza el modelado en *Autodesk 3DS Max*. Para ello se crean 2 planos, uno ajustado a la Vista Frontal y el otro ajustado a la Vista Lateral, ambos sobre los ejes 0. Se aplica como textura el *TurnAround* del personaje, se usa el modificador *UVWMap* para ajustar su mapeo y se congelan los planos. A partir de aquí, se trabaja con los dos visores para ir modelando el personaje.

El personaje se elabora a través de primitivas básicas (cilindros y esferas) sobre las que se van moviendo y escalando los vértices para ajustarlos al boceto (previa conversión a polígono editable). Los detalles del personaje se consiguen a través de las texturas, pero el texturizado final se lleva a cabo en *Unity*. Finalmente, para la exportación a *Unity* se guarda el archivo a *.FBX*.

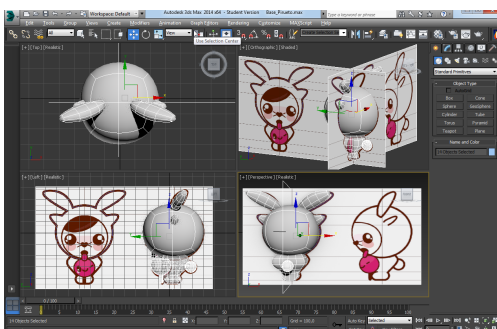


Figura 25: Vistas Piruetto en 3DS Max.

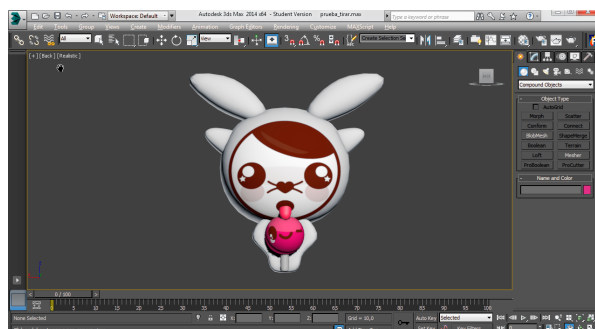


Figura 26: Piruetto con texturizado previo en 3DS Max.



Figura 27: Textura cara Piruetto.



Figura 28: Textura cara piruleta de Piruetto.

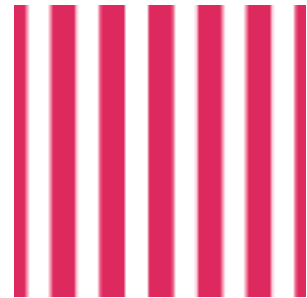


Figura 29: Textura cuerpo Piruetto.

10. 3 Creación y Modelado del 'Cupcake Car'.

Para la realización del coche del personaje se modela un 'Cupcake' sobre el que va montado el personaje. Ya que se pretende crear un mundo dulce de golosinas.

Tanto para el modelado del *Car Cupcake* como para las golosinas del escenario no es necesario crear un boceto en 2D, debido a la simplicidad de las formas.

El *cupcake* se modela a partir de 3 elementos: la base del cake, a partir de la primitiva cilindro; el relleno, a partir de una esfera; y la nata a partir de la *Spline Helix*. Todos los elementos se transforman en un polígono editable y se modelan a través del movimiento y escalado de los vértices y caras. Se suaviza la figura a través del activado de *Use NURMS Subdivisión*.

Las ruedas se añaden en Unity, para evitar conflictos con las físicas.

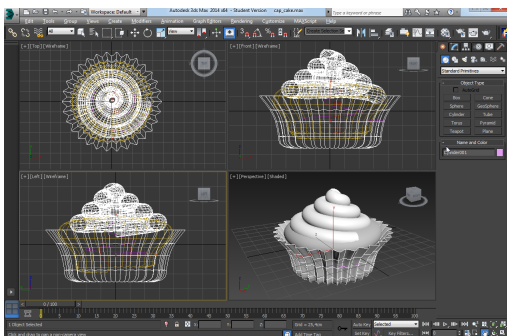


Figura 30: Modelado Cupcake en 3DS Max.



Figura 31: Aspecto final del Cupcake Car con Piruetto.

10. 4 Creación y Modelado de los elementos del escenario

Para la elaboración del escenario se modelan en 3DS Max, los siguientes objetos:

- Piruletas (elementos principales del juego).
- Chupa-chups.
- Nubes de nata y fresa.
- Galletas.
- Chocolate.
- Y aprovechamiento de otros elementos contenidos en el personaje (como la nata) para terminar la decoración final.

** Para su mejor visualización se presenta una ampliación en el Anexo de la Memoria correspondiente.

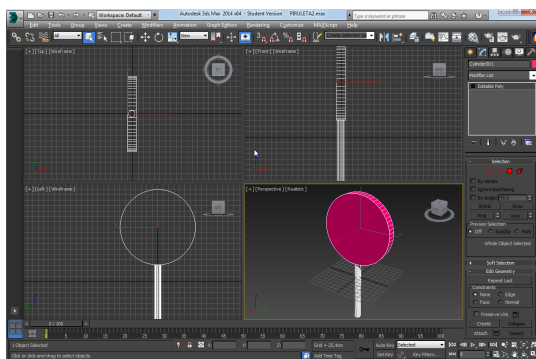


Figura 32: Modelado piruletas en 3DS Max.

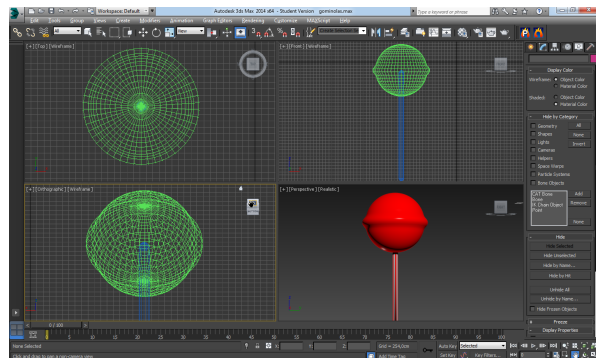


Figura 33: Modelado Chupa-Chups en 3DS Max.

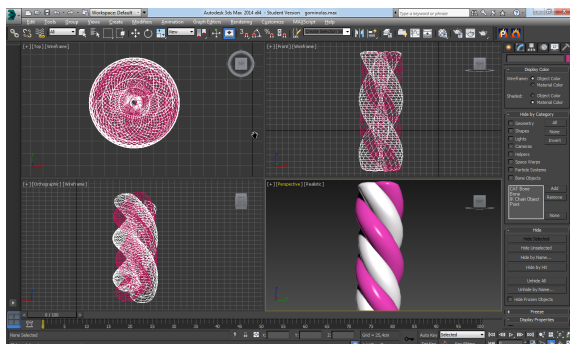


Figura 34: Modelado galleta en 3DS Max.Figura

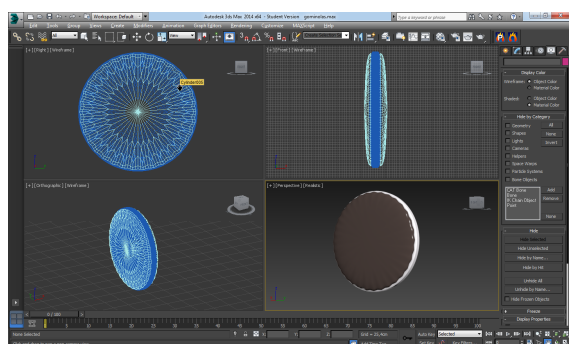


Figura 35: Modelado nube en 3DS Max.

10. 5 Creación del Logotipo del Juego

El logotipo, isotipo, isologo se realizan pensando en una tipografía divertida, con colores que recuerdan a las piruletas y en general se usa una estética que evoca las golosinas.

También se ha incluido un símbolo de 3 líneas horizontales que recuerdan los *rallys* o carreras de coches. Y se sigue el estilo *Kawaii* del juego. Se ha realizado también una pequeña leyenda que acompaña al logotipo: "Piruetto no sabe si es un gato o un conejo pero es feliz porque come piruletas cada día". Para más información acerca de formas, colores y tipografías puede consultarse el Manual de Estilo del Juego.



Figura 36: Creación Logotipo Piruetto's Race.

10. 6 Interfaz Gráfica

La Interfaz Gráfica se realizan en *Adobe Illustrator* siguiendo el color y la línea del proyecto. Puede verse el detalle de todos los elementos que componen la Interfaz Gráfica en el Libro de Estilo, en la sección de Anexos de esta Memoria.

Posteriormente, se importan todos los elementos a *Unity* y se transforman a *Sprites*, ya que por defecto, *Unity 3D* interpreta toda las imágenes como texturas. Los elementos de la UI se trabajan a través del *Canvas* de *Unity*. Se utilizan elementos de texto, *images*, *Raw Image* y botones. En la parte superior se puede observar el *Timer*, los puntos que suman o restan después de cada colisión con una piruleta y el sistema de vidas. En la parte inferior se observa el panel de control, donde aparece la letra que se pide cada x tiempo de forma aleatoria; el número de piruletas correctas e incorrectas; y finalmente la suma total de puntos.

Para la visualización de todos los elementos realizados para la interfaz gráfica puede consultarse el Libro de estilo del proyecto.



Figura 37: Aspecto gráfico de los elementos de interfaz.

10.7 Lenguaje de Signos y Flat Design

El objetivo principal del juego es el aprendizaje del lenguaje de signos, así que es necesario buscar las figuras que representan dichos símbolos. En un inicio se piensa en realizar su elaboración a través de las fotografías de cada signo, pero en la práctica después de realizar varias pruebas, no ha terminado de funcionar ya que este tipo de fotografía es un poco compleja. Se necesita conseguir que el signo se visualice de la mejor forma posible y posteriormente tratar las fotografías para que no desentonen con el juego.



Figura 38: Ejemplo de fotografías realizadas para el lenguaje de signos.

Como el resultado del proceso anterior no es satisfactorio, se busca otra alternativa más ágil y que pueda funcionar. Se piensa en la técnica del Flat Design²² para utilizar iconos minimalistas que potencien su visibilidad.

²² **Flat Design.** Es una técnica de diseño plano y minimalista. Se puede ver en proyectos como Windows Vista.

Unos pictogramas con los que inicialmente se realizan pruebas, son los siguientes, pero no terminan de convencer, ya que estos sólo presentan el contorno y dificultan la visualización:

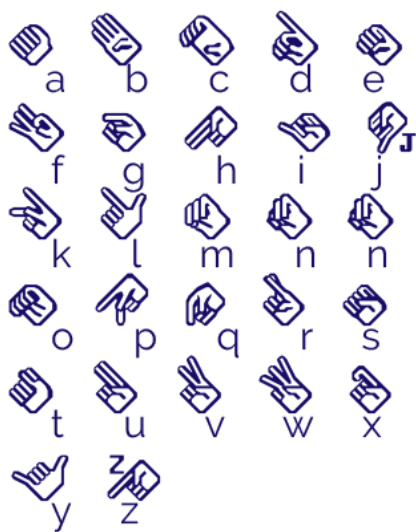


Figura 40: Prueba de pictogramas finalmente descartados.

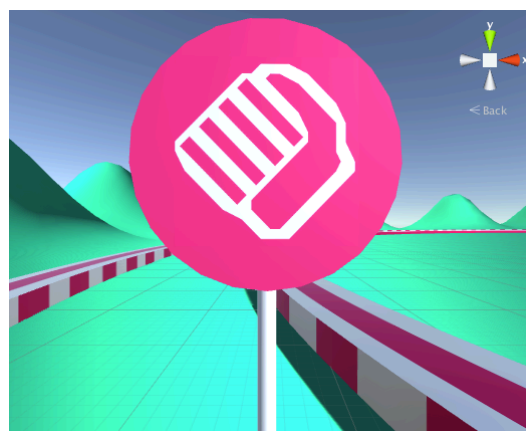


Figura 39: Prueba de pictograma Lenguaje de Signos.

Finalmente se selecciona una tipografía que trabaja el Lenguaje de Signos: *Dissaramas CRU Fingerspelling Font*²³. Es un proyecto de Dominio Público, incluso está permitido el uso comercial.

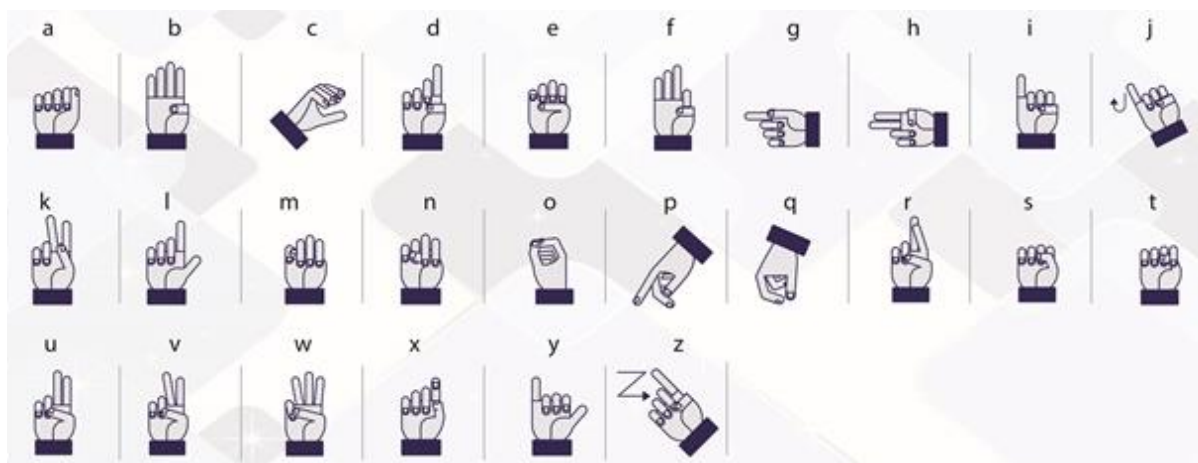


Figura 41: Tipografía *Dissaramas CRU Fingerspelling Font*.

Para su adaptación al juego, se realizan unos pequeños cambios:

- 1- Eliminación del puño de la tipografía, ya que es un elemento prescindible.

²³ *Dissaramas CRU Fingerspelling Font*. <http://www.fontspace.com/cru-dissaramas/dissaramas-cru-fingerspelling>

- 2- Redondeado de la parte inferior que ha quedado recortada por el puño. Esto dará una terminación más natural y se adaptará mejor, ya que todos los elementos del juego presentan formas redondas.
- 3- Cambio de color a blanco, para resaltar sobre el fondo de la piruleta y potenciar su visibilidad. Además, el blanco es 1 de los colores de trabajo establecidos en el Manual de Identidad del proyecto.



Figura 42: Tipografía original.

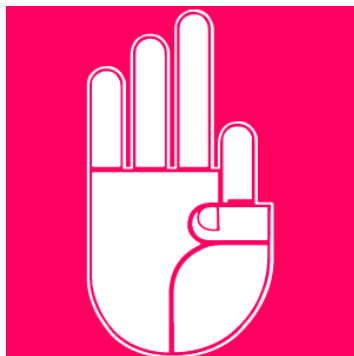


Figura 43: Tipografía modificada.

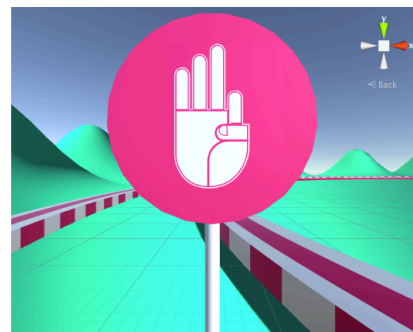


Figura 44: Pictograma posicionado en Unity.

10. 8 Accesibilidad: *Closed Caption*

El subtulado para sordos es una opción de accesibilidad de las más importantes para ser incluidas en los juegos. El *Closed Caption*²⁴ diferencia por un lado, los subtítulos que provienen del diálogo que se escriben debajo y que se han convertido en una característica estándar en la mayoría de juegos y por otro, los subtítulos que aportan un plus y mejoraran la audición, como mostrar en formato de texto los sonidos que se representan en el juego para una mayor inmersión del jugador. Para una persona con una discapacidad auditiva es importante conocer el ruido ambiental ya que ayuda a sumergirse en el juego y aumenta la experiencia de usuario.



Figura 45: Símbolo del Sistema Closed Caption. No debe confundirse con Subtítulo o Teletexto.

²⁴ **Closed Caption**. Wikipedia. https://es.wikipedia.org/wiki/Closed_Caption

10.9 Instalación Unity y Ubicación

El proceso de instalación es muy sencillo, únicamente hay que dirigirse al sitio web de Unity y descargar la aplicación, aceptar los términos de la licencia (que en este caso ha sido Unity Free), elegir los componentes e instalar. Al abrir la aplicación puede verse:

- Panel *Scene*: muestra el mundo virtual y lugar de trabajo que despliega un menú.
- Panel *Game*: donde se ejecuta el juego.
- Panel *Hierarchy*: objetos presentes en la escena, mostrando sus relaciones jerárquicas.
- Panel *Project*: Recursos que componen el juego.
- Panel *Inspector*: Muestra las propiedades de un objeto (*GameObject*)

La forma de trabajo se realiza modo online para recibir notificaciones sobre actualizaciones, código obsoleto y demás *feedback* por parte de Unity.

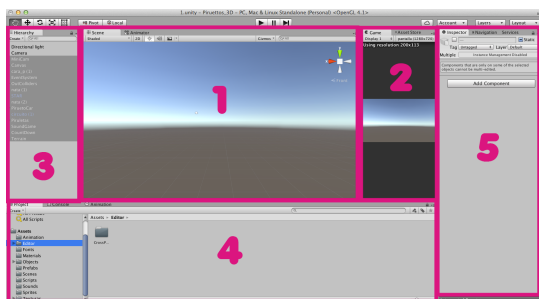


Figura 46: Elementos IU (Interfaz Gráfica) de Unity.

10.10 Iluminación y Sombras

La iluminación y las sombras son componentes que requieren un consumo elevado de recursos. Unity trabaja con 4 tipos de luz: *Ambient Light*, *Point Light*, *Directional Light* y *Spotlight*. La versión 5 de Unity incorpora por defecto una *Directional Light* al crear un proyecto nuevo. Dicha luz es similar a la del Sol, está formada por rayos paralelos entre sí, y permite iluminar los objetos de manera uniforme. Para la realización del juego se ha necesitado una lux direccional para todo el escenario. Y 3 más para enfocar hacia las piruetas, con el objetivo de mejorar su visibilidad. En cuanto a las sombras se utilizan Hard y Soft Shadows.

10.11 Cámaras

Se utilizan 3 cámaras. Una de ellas lleva adjunto un script de seguimiento (*Smooth Follow*) propio de Unity, para seguir al coche por todo el circuito. Dicha acción se podría haber realizado a través del uso de *Parenting*, asignando un objeto hijo a un objeto padre, simplemente arrastrando la

cámara dentro del objeto coche en el panel *Hierarchy*, pero se utiliza un script de cámara, para conseguir un movimiento más personalizado.

La vista de la cámara es en tercera persona, por lo que en todo momento se podrá observar al personaje. El modo de cámara utilizado es el de perspectiva, donde los objetos se ven más pequeños conforme se aleja la cámara.

Otra cámara se encarga de la animación inicial del juego. Y finalmente, también se incorpora una cámara secundaria en forma de mini cámara, en el lado izquierdo inferior para que el espectador pueda ver una vista frontal del personaje.



Figura 47: Vista Cámara y vista Minicámara.

10. 12 Creación del escenario

Unity incorpora un editor de terrenos que funciona a través de un conjunto de diferentes brochas que modelan el terreno. Esta forma de trabajar se deja al margen ya que se comprueba que hay que tener una cierta práctica con los pinceles para que el resultado sea satisfactorio del todo y se ha optado por realizar un sencillo *Heightmap*, para la creación de las montañas.

En *Photoshop* se ha creado un archivo RAW en escala de grises de 512x512, (Unity trabaja los tamaños en potencias de 2) donde las partes blancas son los picos más altos y las negras las partes planas, las escalas de grises son alturas intermedias entre el negro y el blanco. Una vez importado el terreno, se suaviza un poco, y se añade una textura. Esta técnica también se utiliza

para crear terrenos realísticos a través de *Heightmaps* de paisajes reales, pero no es el objeto de este proyecto.

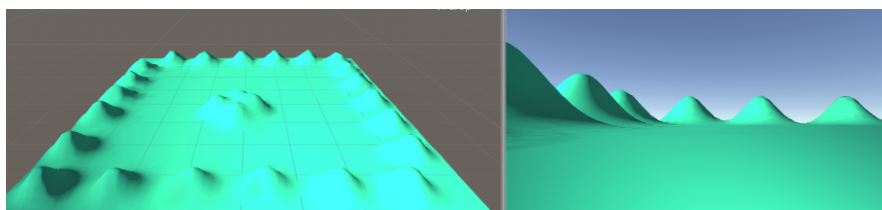


Figura 48: Aspecto final montañas en Unity.

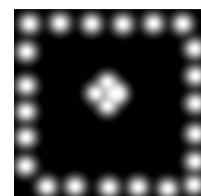


Figura 49: *Heightmap*.

Posteriormente, en la fase final del desarrollo del juego, concretamente en la fase de optimización del mismo, se decide eliminar dicho terreno, ya que incrementa a razón de unos 15 MG aprox. el proyecto y se considera un elemento prescindible. En su lugar, se utiliza un plano que realiza la misma función y apenas repercute en el peso final del proyecto.

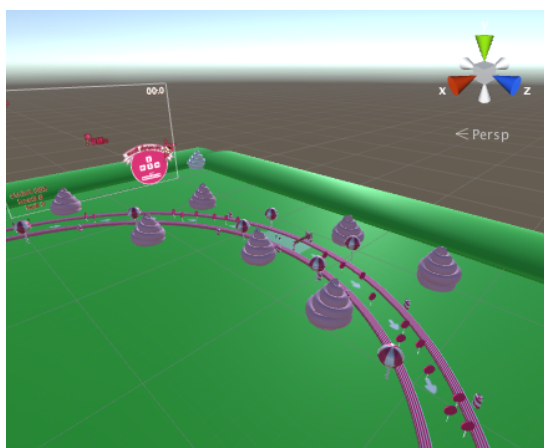


Figura 50: *Uso de plano en lugar de terreno*

10. 13 Mapa del juego

El mapa del juego se basa en un circuito circular por donde se moverá el personaje montado en un "Cap Cake" (coche). El personaje tiene total libertad de desplazamiento dentro del circuito. Dicho circuito se ha elaborado en Autodesk 3DS Max, a partir de una Shape Circular, a la que se le ha agregado mapa de coordenadas para que la textura quede correctamente emplazada, (Modificador UVW/Map). Finalmente, se exporta a .FBX y se importa a Unity. Las texturas finales se incorporan en Unity.

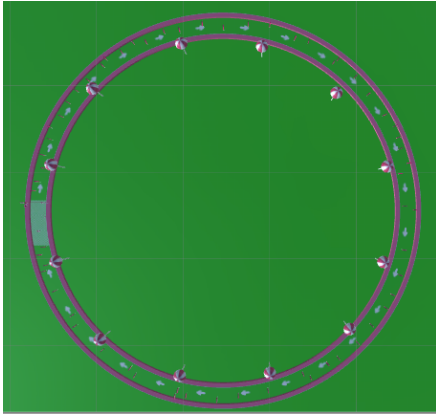


Figura 51: Vista circuito en Unity.

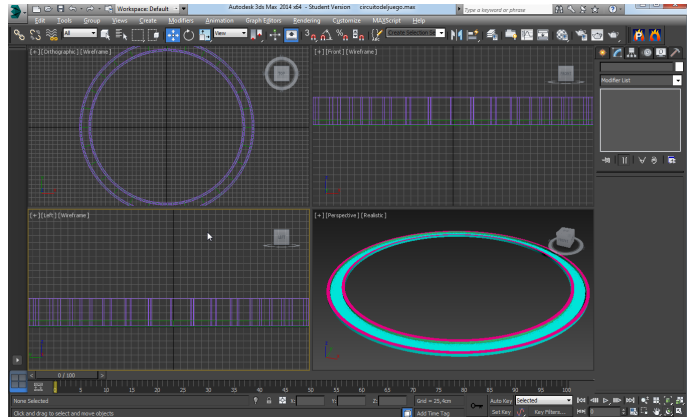


Figura 52: Circuito modelado en 3DS Max.

10. 14 Entradas por teclado

Es necesario controlar las entradas por teclado del usuario, ya que es lo que permite la interacción del jugador con la aplicación. El jugador necesita mover el coche para desplazarse, y lo realiza a través de las flechas de teclado. También es necesario obtener ayuda durante el juego, y esta se realiza a través de la barra espaciadora. Para leer las entradas se utiliza la clase Input. Desde el menú Edit> Project Settings> Input se configura el dispositivo.

10. 15 Scripting

En Unity es posible utilizar 3 lenguajes de programación: JavaScript, C# y Boo, incluso se pueden alternar los 3 al mismo tiempo, sin haber conflicto. La realización de este proyecto se centra únicamente en C#, por razones de claridad y por ser el lenguaje más potente y con más documentación en la red.

Hasta el momento, hay realizados 7 Scripts que controlan videojuego:

- **CenterOfMass.cs:** Dispone un centro de masas para el Cupcake Car.
- **ChangingLetter.cs:** Cambia la letra que Piruetto tendrá que buscar, de forma aleatoria cada x tiempo. Es el Script principal en el que se basa el juego.
- **CountDown.cs:** Maneja lo que ocurre al principio del juego. La cuenta atrás de salida, la rotación de la cámara, el inicio de la música del juego y los sonidos de efectos especiales, como el aplauso del público.
- **DetecWheel:** Detecta si las ruedas del Cupcake Car colisionan con el suelo (volcado), y si no es así acaba la partida.

- **SimpleCarController.cs:** Gestiona el manejo del 'Cupcake Car' (coche). Junto con el anterior, son los *Scripts* más importantes del juego. Se ha realizado siguiendo el Tutorial de Unity Script.
- **LoadGame.cs:** Gestiona el menú de inicio y la pantalla de *Game Over*.
- **Lollipops.cs:** Controla las colisiones del *Cupcake Car* con las piruletas, controlando si es correcta o incorrecta y activando los puntos y sonidos correspondientes.
- **Timer.cs:** muestra un contador de tiempo en la interfaz.
- **SmoothFollow.cs:** Script de seguimiento para la cámara propio de Unity. Se ha utilizado para que la cámara siga al coche cuando esté en movimiento.
- **Replay:** Script para volver a iniciar la partida cuando el juego finaliza.

10.16 Físicas

Para que el movimiento del coche tenga un comportamiento físico convincente y se vea afectado por colisiones, gravedad y otras fuerzas físicas, se usa el motor de física *NVIDIA Physx* de Unity. Sin el empleo de físicas, se puede mover el coche a través de su posición y rotación, pero no responde a un movimiento natural como se espera.

Se han utilizado los siguientes componentes:

- **Rigidbody:** permite que el objeto se convierta en un cuerpo rígido que responda a la gravedad. Con las siguientes propiedades principales: Masa del objeto, Resistencia al aire (Drag), Resistencia al aire en los giros (Angular Drag), es Kinematic (afectado por la gravedad).
- **Box Collider:** Colisionador de tipo cubo. Es un collider dinámico ya que está posicionado en un objeto con rigidbody. Este componente es necesario para que cuando el coche colisione con una piruleta se pueda detectar la colisión. Las piruletas, también poseen un Collider pero con el Trigger activado, para poder desencadenar un evento cuando ocurra una colisión. Esto interesa para que al colisionar se pueda detectar si la piruleta es correcta o no, se active el sistema de puntos positivos o negativos, y finalmente se destruya el objeto, desapareciendo del escenario.
- **Colliders de otros elementos:** Todos los elementos de la escena que no deben ser atravesados, como el suelo (sinó el personaje lo atravesaría) se les ha adjuntado un collider.
- **FixedUpdate():** Al trabajar con rigidbody y fuerzas, ha sido necesario sustituir la función Update() por FixedUpdate(), para que los cálculos del motor de física se basen en actualizaciones fijas de tiempo
- **Time.deltaTime():** Al utilizar FixedUpdate(), también ha sido necesario trabajar con esta función para convertir el frame rate del juego en independiente. Al aplicar esta función se trabaja por segundos y no por frames. Las acciones se ejecutarán cada x segundos en lugar de cada x frames.
Con las últimas 2 funciones se pretende mantener una constancia en los diferentes dispositivos.

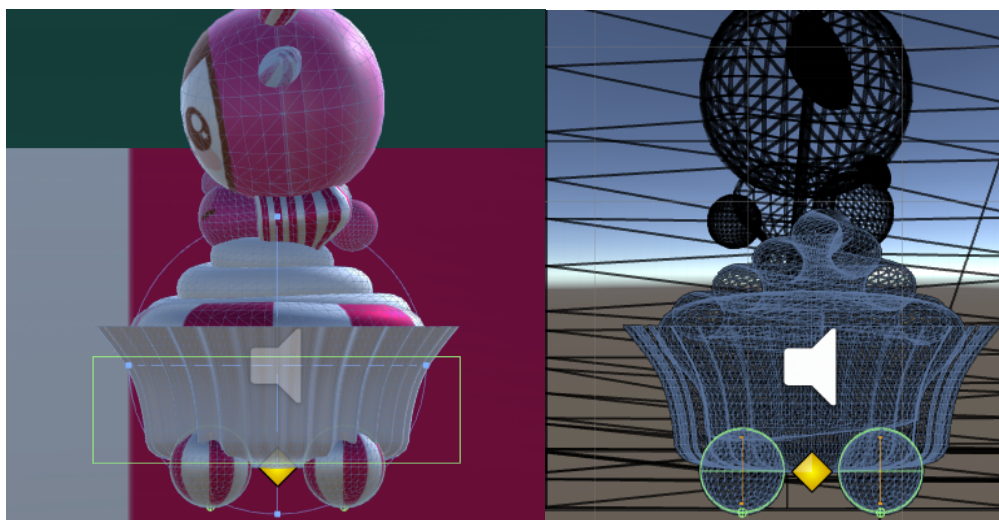


Figura 53: Componentes *Box Collider*, *WheelColliders* y centro de masas del *Cupcake Car*.

10. 17 Prefabs (prefabricados)

Los *prefabs* son instancias de un objeto que comparten las mismas propiedades, de forma que cuando se edita un objeto, no hay que repetir la edición en todas las copias. El *prefab* almacena el *GameObject* completamente con sus componentes y propiedades como si fuera una plantilla. Se utiliza esta técnica para la instanciación de todas las piruletas a lo largo del circuito. Dichas piruletas poseen las letras del alfabeto dactilológico que cambian aleatoriamente cada x tiempo y son el objeto del juego.

10. 18 Animación

Es preciso realizar pequeñas animaciones sobre el personaje para dotarle de más realismo y que no se vea con un aspecto tan rígido. Las animaciones son las siguientes:

- Movimiento de orejas.
- Movimiento de cabeza hacia la derecha e izquierda cuando gira en estos sentidos.

También se realiza una pequeña animación de rotación para el principio de juego, con la finalidad de ofrecer al espectador un plano general del mismo.

10. 19 Sonidos FX y Música del Juego.

Unity soporta las extensiones de sonido: .mp3, .ogg, .wav, aiff/.aif, .mod, .it, .s3m, .sm.

Para el proyecto se elige el formato .ogg ya que ofrece una mayor compresión de los archivos, y reduce su tamaño, sin conllevar pérdidas de calidad. Además, utiliza un sistema de compresión inteligente que elimina las partes no audibles para el oído humano y parte del ruido ambiente. Todos los sonidos han sido comprimidos con Audacity.

Para el tratamiento del sonido, previamente se realiza un listado de los Sonidos, Efectos y Grabaciones que se necesitan:

- Arranque de Motor.
- Sonido de Motor.
- Sonido Bip Cuenta Atrás.
- Sonido Aplausos Inicio.
- Sonido Colisión con Piruleta Buena.
- Sonido Colisión con Piruleta Mala.
- Sonido Quita Vidas.
- Sonido Game Over.
- Música Intro Juego.
- Música Juego.

Posteriormente se buscan dichos recursos en Internet, principalmente se han obtenido de 3 fuentes: *Flashkit*, *PlayOnLoop* y *Taira Komori*. Por otro lado, la música que se escucha durante el transcurso del juego es de tipo *Loop*.

Se necesitado, también, grabar sonido para realizar el sonido de las letras del alfabeto. A los sonidos se han aplicado los filtros: Nivelado, para igual el sonido, Reducción de ruido, para eliminar el ruido de fondo, y cambiar tiempo para aumentar poco la velocidad.

Primeramente se han grabado todas las letras en un único archivo, se han aplicado los filtros y finalmente se han exportado a .ogg cada letra por separado.

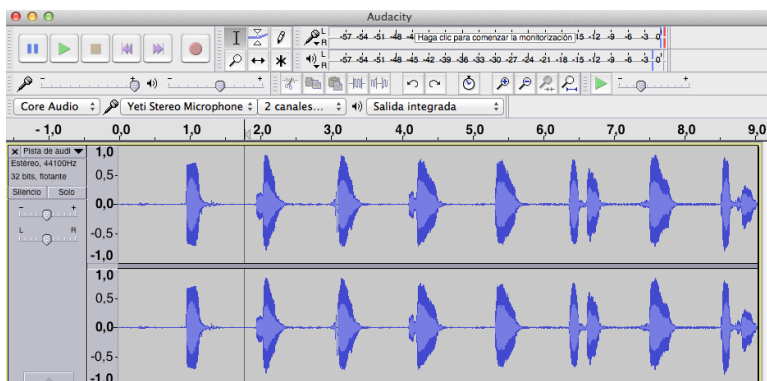


Figura 54: Proceso grabación letras alfabeto con Audacity.

10. 20 Optimización del juego.

- Se elimina del Update() algunas funciones que no era necesario estar comprobando en cada *frame*.
- Se optimiza la compresión de algunas texturas.
- Se utiliza el tamaño texturas potencias de 2.
- Se intenta compartir el mayor número de texturas y materiales posible.
- Se elimina el *gameobject* terreno con montañas y se substituye por un plano ya que se ha comprobado que el uso de terrenos incrementa bastante el peso del archivo final.
- Eliminación de degradados ya que para su visualización perfecta se había de utilizar color verdadero con incremento peso considerable

10. 21 Resolución y publicación de proyecto.

Unity es una herramienta Multiplataforma que permite publicar desde un simple menú a varias plataformas, por ello se publica para:

- Sistema Operativo Windows, resolución pantalla: 1280x720.
- Sistema Operativo Macintosh, resolución pantalla: 1280x720.
- Versión Web. <http://multimedia.com/box/pirueto/>

Es necesario importar las 2 las escenas que se han realizado para el juego para que Unity compile en 1 solo archivo y genere la aplicación.

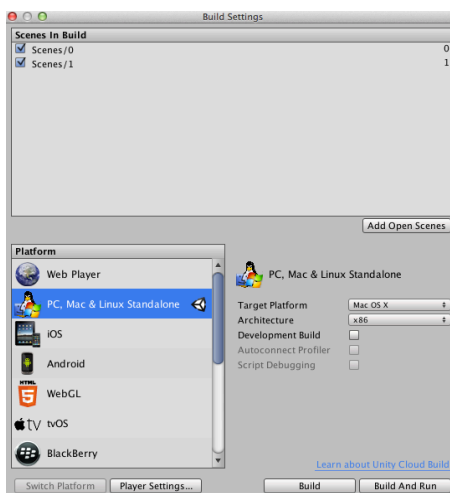


Figura 55: Pantalla *Build Settings* de Unity.

Así mismo, a través del estudio realizado a 80 personas (como se ha mencionado anteriormente) se detecta la necesidad de publicar para Mobile también, aunque queda fuera del ámbito del proyecto.

5. ¿Que Sistema Operativo utilizas para jugar? (77 respuestas)

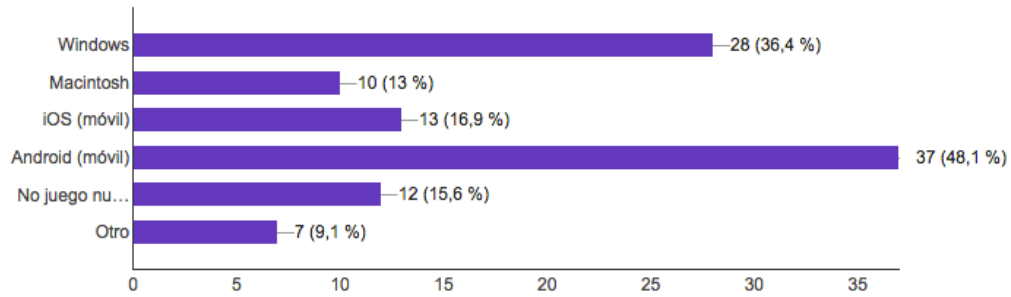


Figura 56: Datos encuesta Sistema Operativo empleado para jugar.

11. APIs utilizadas

OpenGL (Open Graphics Library) es una especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D. La interfaz consiste en más de 250 funciones diferentes que pueden usarse para dibujar escenas tridimensionales complejas a partir de primitivas geométricas simples, tales como puntos, líneas y triángulos. Fue desarrollada originalmente por *Silicon Graphics Inc. (SGI)* en 1992 y se usa ampliamente en CAD, realidad virtual, representación científica, visualización de información y simulación de vuelo. También se usa en desarrollo de videojuegos, donde compite con Direct3D en plataformas Microsoft Windows.

13. Prototipos

13.1 Lo-Fi



Figura 57: Wireframe Menú principal.



Figura 58: Wireframe Pantalla Juego.

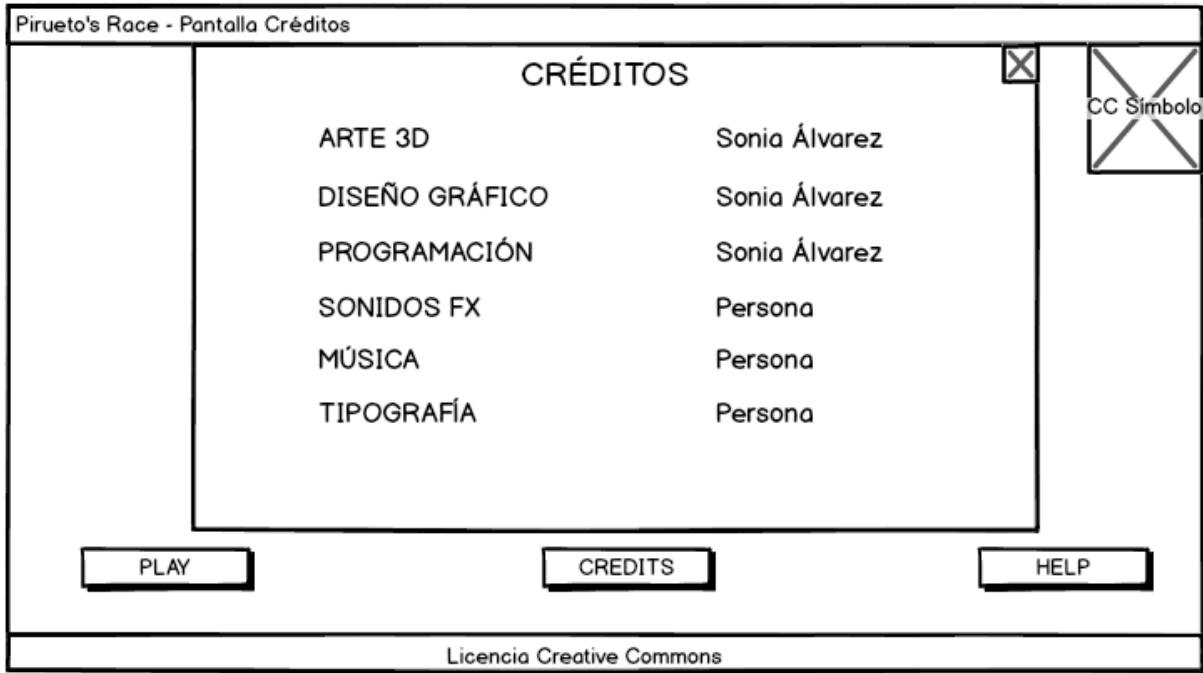


Figura 59: Wireframe pantalla créditos

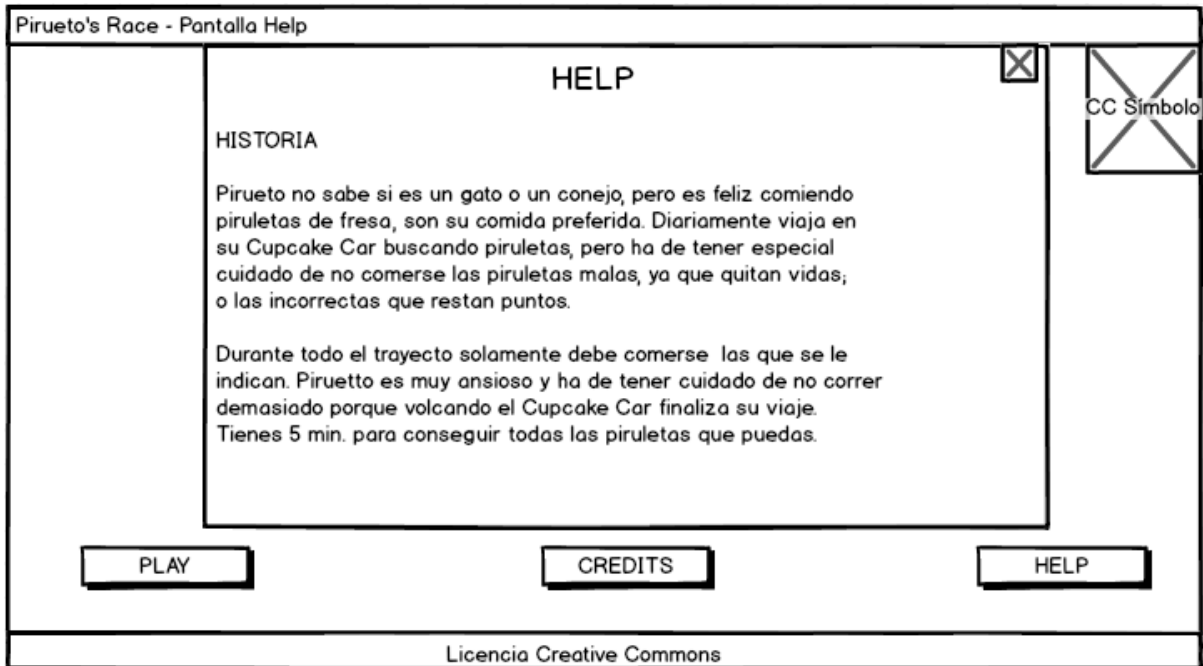


Figura 60: Wireframe Pantalla Help



Figura 61: Wireframe Game Over.

13.2 Hi-Fi

Pantalla Menú principal:



Figura 62: Prototipo Hi-Fi Menú Principal.

Pantalla juego:



Figura 63: Prototipo Hi-Fi pantalla de juego.

Pantalla Créditos:

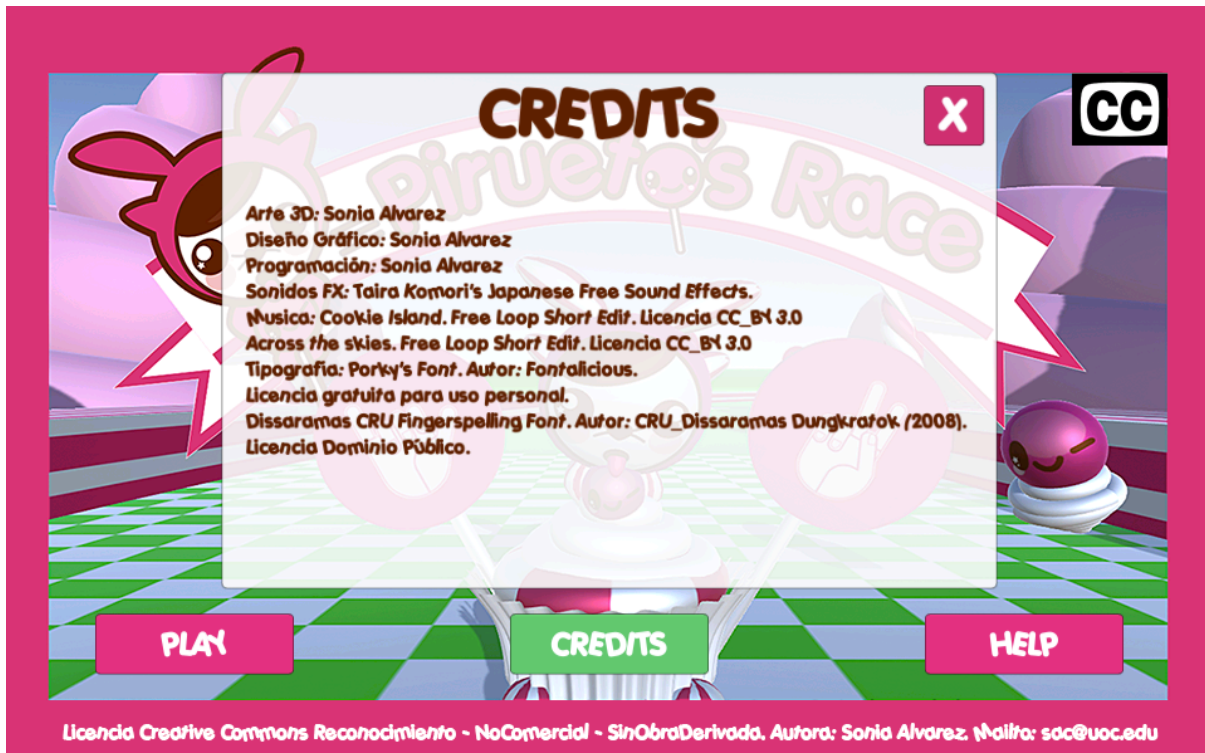


Figura 64: Prototipo HI-Fi pantalla de Créditos.

Pantalla Help:



Figura 65: Prototipo HI-Fi pantalla Help.

Pantalla Game Over:



Figura 66: Prototipo HI-Fi pantalla Game Over.

14. Guiones

PIRUETO'S RACE GAME SCRIPT

SEQUENCE 1: WARM UP.-

SCENE 1: EXT. PARRILLA DE SALIDA -INICIO DÍA (INTRO)

PIRUETO no sabe si es un gato o conejo pero es feliz porque come piruletas de fresa cada día. Piruetto se está preparando en su Cupcake para participar en su rally particular de cada día. Es un día soleado, la afición y la gente se oye en las gradas, hoy Piruetto tiene la oportunidad de superarse a sí mismo y conseguir más piruletas de fresa que el día anterior. Piruetto mira a su alrededor y observa el entorno. Al mismo tiempo, se ve un plano general orbital para situar al espectador.

En 4 segundos va a comenzar la carrera. Piruetto se prepara, 3,2, 1 y GO!

FX: 3, 2, 1

FX: START

FX: START ENGINE

SEQUENCE 2: PIRUETO'S RACE STARTS.-

SCENE 2: EXT. CIRCUITO-TRANSCURSO DÍA

Piruetto empieza su carrera, está nervioso y tiene muchas ganas de conseguir sus preciosas piruletas, pero no todas son buenas, ha de tener especial cuidado de no comerse las piruletas malas, ya que quitan vidas, o las incorrectas que restan puntos.

Durante todo el trayecto solamente debe comerse las que se le indican, por eso Piruetto observa atentamente todas las piruletas. Dos CONTROLADORES DE PIRULETAS siguen el recorrido junto a su Piruetto para cerciorarse que elige las correctas y que no ocurre ninguna incidencia.

Finalmente, Piruetto es muy ansioso y ha de tener especial cuidado de no correr demasiado porque volcando su Cupcake puede finalizar su viaje.

MUSIC: GAME

FX: ENGINE

HELP: SPACE KEY

MOVE CUPCAKE CAR : RIGHT, LEFT, UP, DOWN or WASD

SEQUENCE 3: GAME OVER.-

SCENE 3: EXT. CIRCUITO-FIN DÍA

Si Piruetto recopila todas las Piruletas buenas finalizará su aventura y Piruetto continuará sintiéndose feliz. Si Piruetto no consigue controlar su ansia de Piruletas y se precipita volcando su Cupcake, o pierde todas sus vidas o puntos finalizará su aventura y Piruetto tendrá que volver a intentarlo en otra ocasión.

MUSIC: GAME OVER

15. Perfiles de usuario

Según la encuesta previa realizada para obtener una pequeña base de información sobre como encarar ciertos aspectos del proyecto, se ajusta el perfil de los usuarios de la siguiente forma:

A nivel general, el target de dicha aplicación es cualquier persona que quiera aprender la lengua de signos de una forma amena o que simplemente quiera pasar un rato entretenido.

Así mismo, el juego está enfocado a un perfil de usuarios de cualquier edad ya que posee una interfaz simple y una metodología de juego sencilla.

Según los datos recopilados, la mitad de los encuestados (sobre una muestra de 80 personas) juegan a algún tipo de juego de forma habitual y casi la otra mitad declaran jugar alguna vez. Sólo hay un 18% que alega no jugar nunca. Despunta un poco que en cuanto a diferencias por sexo y edad, la mayoría de los que alegan no jugar nunca son de género femenino. Estamos por tanto ante un producto que tiene bastante potencial entre los usuarios a nivel general.

En cuanto a los dispositivos con los que los usuarios juegan habitualmente destaca el Smartphone, es el producto estrella, ya que la mitad de los usuarios juegan con este dispositivo, (Sistema Operativo *Android*). Seguidamente encontramos el ordenador de sobremesa con sistema operativo *Windows*, y *Tablet*. Se deja al margen la consola que también obtiene buenos resultados pero no es el ámbito de este proyecto. La mitad de los encuestados también manifiesta que les gusta jugar en formato on-line, así que no se descarta esta plataforma.

Por otro lado, la mayoría de encuestados piensan que el aprendizaje a través del juego es una forma divertida y buen método para aprender. Si lo llevamos al terreno de las personas que poseen algún tipo de discapacidad, el 51% de los usuarios reconoce que no existen suficientes juegos en el mercado enfocados a este sector y el 45% desconoce el tema.

A modo de resumen se detalla que el target principal del producto, es cualquier persona de indistinto género y edad que interactúa principalmente con: *Smartphone (Android)*, *Windows*, plataformas web y que está interesada en el aprendizaje de la lengua de signos o simplemente pasar un rato diversión.

16. Usabilidad/UX y Accesibilidad

Principalmente este juego se basa en el Diseño Centrado en el Usuario, ya que es un juego concebido para el aprendizaje del jugador. Se diseña de la forma más sencilla y entendible posible para facilitar el proceso de aprendizaje.

La interacción que realiza el usuario es a través del teclado, donde debe utilizar las teclas de desplazamiento para mover el coche-personaje con el que se mueve por todo el escenario, recopilando las piruletas que necesita.

La navegación también se realiza de una forma muy sencilla. Al ejecutar el juego, se abre un menú con 3 botones: *Jugar*, *Créditos* y *Help* (la versión de escritorio también posee la opción de salir del juego). En la misma pantalla también se observan las instrucciones de juego y cómo jugar. Si el usuario pulsa sobre el botón de *Jugar*, el juego se carga y el jugador puede comenzar la partida. Cuando el juego finaliza, aparece una pantalla de *Game Over* con 1 botón que nos permite volver a jugar, y otro botón para salir del juego.

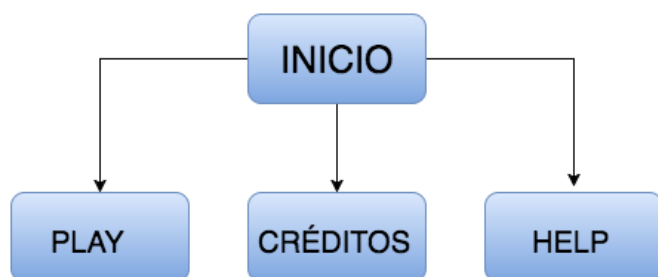


Figura 67: Mapa de Navegación del Juego.

Puesto que durante todo el desarrollo del juego prioriza la visualización de la parte superior de la pantalla, ya que en esta zona aparecen las piruletas con los pictogramas del lenguaje de signos, se dispone toda la información del juego en la parte inferior.

En la parte superior sólo aparece el *Timer* o contador de tiempo, y el sistema de vidas, colocados en las esquinas. Ocasionalmente en el centro superior de la pantalla aparecen mensajes con los puntos que suman o restan.

Para organizar el resto de información del juego se crea un panel de control o *Dashboard*, desde donde el usuario puede ver las letras que a cada momento se solicitan, también consultar la ayuda y finalmente obtener información de las piruletas buenas y malas que lleva acumuladas, así como la puntuación total.

También se tiene en cuenta el tamaño y la forma de los elementos gráficos, para que en todo momento sean visibles por el usuario y faciliten su interacción con el juego.

Por otro lado, se crea un sistema de ayuda que consiste en asociar la letra del alfabeto tradicional con el símbolo del lenguaje de signos. Si el usuario pulsa la tecla espacio, automáticamente la letra del alfabeto se transforma en su signo durante el transcurso de 2 segundos.

El proyecto también incluye una opción de accesibilidad consistente en el subtítulo para sordos: El *Closed Caption* que diferencia por un lado, los subtítulos de los diálogos y por otro, los subtítulos que aportan un plus y mejoraran la audición, como mostrar en formato de texto los sonidos, músicas y demás elemento que se representan en el juego para una mayor inmersión del jugador. Para una persona con una discapacidad auditiva es importante conocer el ruido ambiental ya que ayuda a sumergirse en el juego y aumenta la experiencia de usuario.



Figura 68: Opciones de Closed Captioning.

Finalmente, el juego está ideado para poder ser utilizado por un rango amplio de usuarios. Nada más ejecutar el juego, se obtienen las instrucciones de cómo jugar y la historia del juego.

17. Seguridad

El proyecto se ubica en Google Drive. Todos los productos de Google comparten características en cuanto a la seguridad, además de las mejoras y actualizaciones específicas de cada producto:

- Cifrado mediante el protocolo de transporte SSL²⁵, no importa si se accede desde el navegador web, aplicaciones de escritorio o móviles.
- Todos los datos son cifrados con los estándares que marca la industria.
- Nivel de seguridad de Drive es óptimo, aunque los responsables del producto siempre podrán acceder a los datos del usuario.
- En cuanto a Google Play, constantemente se realizan actualizaciones para mejorar la seguridad de las compras dentro de una app y surgen proyectos e iniciativas de análisis como el proyecto *PlayDrone*²⁶ que analizar la seguridad de las *apps* en Google Play y corregir los errores encontrados.

²⁵ El protocolo SSL (Secure Sockets Layer) fue diseñado con el objeto de proveer privacidad y confiabilidad a la comunicación entre dos aplicaciones. <http://penta2.ufrgs.br/gereseg/unlp/tut1998/ssl.htm>

²⁶ **PlayDrone**. <http://www.pandasecurity.com/spain/mediacenter/seguridad/playdrone-para-analizar-la-seguridad-en-google-play/>

18. Tests

Los tests de usuario se realizan sobre las 3 plataformas para las que se exporta el proyecto:

- Windows
- Mac
- Web

En la fase *Alpha* las pruebas de usuario son realizadas por mi persona. En la fase Beta cuando la totalidad del juego ya está operativa es cuando se utilizan usuarios externos al juego para testear. Los tests se realizan en persona, es muy importante realizarlos de forma presencial para analizar el comportamiento de los usuarios, mientras juegan. De esta forma se tiene la oportunidad de dialogar con los usuarios después de todas las pruebas. En definitiva, se ha priorizado calidad de los tests ante la cantidad. Los *testers* han realizado las pruebas en las 3 plataformas y han jugado durante varios días.

Los *tests* preparados analizan los siguientes aspectos:

- Preguntas sobre el perfil del usuario (4)
- Identidad (4)
- Diseño Gráfico (3)
- Navegación (4)
- Jugabilidad (7)
- Feedback (2)
- Utilidad del juego (2)
- Pregunta abierta de posibles mejoras (1)

En general, los resultados después de las pruebas y de haber mantenido *feedback* oral, son los siguientes:

- El aspecto gráfico del juego, la originalidad, creatividad, idea del juego y la forma en que está implementado para aprender el lenguaje de signos, son los factores mejor valorados. La gama de colores gusta, se considera muy apropiada para un juego de estas características y se reconoce la utilidad del juego.
- En cuanto a la información que ofrece el juego a medida que se va jugando, los menús del juego, la transcripción de las letras y sonidos son aspectos que también salen bien parados, aunque se echa de menos algunas opciones de configuración o personalización de juego, como el poder desactivar el sonido en un momento dado.

- La jugabilidad del juego gusta bastante a nivel general, pero hay un aspecto bastante importante a mejorar: el manejo del coche resulta un tanto difícil ya no por la teclas que se deben utilizar, ya que son las flechas del teclado, sino porque se detecta que el coche tarda un poco en acelerar, desde un estado de parado hasta que consigue la velocidad de juego. Con lo cual en algunos momentos el usuario puede aburrirse o ponerse nervioso.

A continuación se incluyen los tests de usuarios que se han diseñado y realizado:

TEST USUARIO PIRUETO'S RACE GAME

Antes de comenzar con las preguntas sobre el juego, comentarle que puede que haya ciertas acciones que no podrá completar de forma óptima. Si eso ocurre, significará que el juego debe ser perfeccionado y la prueba será exitosa. Si todo sale bien, significará que el sitio fue diseñado pensando en el usuario y también que esta prueba alcanzó el éxito.

1. Nombre:

Félix Alvarez

2. ¿Qué experiencia tiene con ordenadores?:

Poca

3. ¿Juega normalmente a Juegos en el ordenador?:

No. Juego en consola ocasionalmente

4. ¿Qué tipo de juegos le gustan?:

Cartas y Acción

INICIO TEST

IDENTIDAD

1. ¿Se identifica fácilmente el logotipo, el nombre del juego y autor?

Sí pero el nombre se ve muy pequeño.

2. ¿En términos generales, los elementos gráficos ayudan a identificar fácilmente una identidad corporativa que lo diferencia de otros juegos?

Sí, no se puede confundir con otros juegos.

3. ¿Se identifica claramente quien es el personaje principal del juego?

Sí, se identifica muy bien.

4. Si se tuviese que poner en contacto con la persona propietaria del juego se ofrece información de contacto?

Sí, pero apenas se ve.

ASPECTO GRÁFICO

1. ¿Es apropiado el aspecto gráfico para un juego de tipo infantil?

Sí, es infantil 100%.

2. ¿El aspecto visual es atractivo?

Sí, los tonos y colores están muy bien.

3. ¿Gráficamente: el juego está equilibrado, muy simple o muy recargado?

Está equilibrado.

NAVEGACIÓN

1. ¿La navegación del menú de inicio ofrece la información de forma sencilla y entendible?

Sí, se entiende muy bien.

2. Dentro del menú, ¿es fácil orientarse para regresar al punto de inicio sin perderse en el intento?

No, están los botones que se necesitan.

3. ¿Se echa en falta o sobra algún botón dentro de la navegación?

Sí, se entiende bien.

4. ¿Los textos son suficientemente descriptivos?

Sí, son suficientemente descriptivos.

JUGABILIDAD

1. Antes del inicio del juego se ofrece suficiente información sobre cómo jugar y cual es la historia del personaje?

Sí, se entiende cómo jugar perfectamente.

2. Una vez comienza el juego, ¿se ofrece suficiente información en pantalla sobre los elementos del juego (puntos, vidas, piruletas, etc.)?

Sí, hay suficiente información y se ve claramente.

3. La distribución de los elementos en pantalla, te parece adecuada?

Sí porque los elementos están bien agrupados.

4. ¿El nivel de dificultad del juego es demasiado fácil o excesivamente difícil?

Difícil en la parte de manejo del coche. El coche tarda mucho en acelerar.

5. ¿El número de *items* (piruletas) es demasiado elevado?

No, está bastante equilibrado.

6. ¿Durante la partida se ha encontrado algún error de juego?

Sí, cuando se choca con una piruleta, además de la imagen de la puntuación aparece otra imagen del juego que no tiene nada que ver.

7. El tiempo de juego son 5 min. ¿Se prefiere que la partida sea más corta o más larga?

No, es un tiempo razonable.

FEEDBACK

1. Se recibe suficiente información sobre lo que va ocurriendo durante el juego (puntos positivos, puntos negativos, número de piruletas, inicio de juego, fin de juego, etc.).

Sí, hay suficiente información.

2. ¿Los sonidos del juego ayudan a entender lo que está ocurriendo?.

Sí, ayudan a entender.

UTILIDAD

1. ¿Se cree que este juego puede ser un buen método y una forma divertida de aprender el Lenguaje de Signos?

Sí es bastante original para aprender el lenguaje de signos.

2. El juego a parte de su aspecto educativo, ¿consigue entretener?

Sí, entretiene bastante.

PREGUNTAS ABIERTAS

Posibles mejoras:

- Aumentar la velocidad del juego.
- Los giros a veces son muy lentos y otras veces son muy rápidos.

TEST USUARIO PIRUETO'S RACE GAME

Antes de comenzar con las preguntas sobre el juego, comentarle que puede que haya ciertas acciones que no podrá completar de forma óptima. Si eso ocurre, significará que el juego debe ser perfeccionado y la prueba será exitosa. Si todo sale bien, significará que el sitio fue diseñado pensando en el usuario y también que esta prueba alcanzó el éxito.

1. Nombre:

JAVIER DÍAZ SÁNCHEZ

2. ¿Qué experiencia tiene con ordenadores?:

Nivel usuario.

3. ¿Juega normalmente a Juegos en el ordenador?

En ordenador no, en consola y smartphone

4. ¿Qué tipo de juegos le gustan?

Arcade (plataformas) y Shooter 1ª persona y carreras.

INICIO TEST

IDENTIDAD

1. ¿Se identifica fácilmente el logotipo, el nombre del juego y autor?

Sí.

2. ¿En términos generales, los elementos gráficos ayudan a identificar fácilmente una identidad corporativa que lo diferencia de otros juegos?

Sí, se identifica una marca personalizada.

3. ¿Se identifica claramente quien es el personaje principal del juego?

Sí.

4. Si se tuviese que poner en contacto con la persona propietaria del juego se ofrece información de contacto?

Sí, pero cuesta un poco de ver.

ASPECTO GRÁFICO

1. ¿Es apropiado el aspecto gráfico para un juego de tipo infantil?

Sí. Muy acertado el mundo de golosinas en el que se desarrolla el juego.

2. ¿El aspecto visual es atractivo?

Si. Es divertido.

3. ¿Gráficamente: el juego está equilibrado, muy simple o muy recargado?

Normal, está equilibrado.

NAVEGACIÓN

1. ¿La navegación del menú de inicio ofrece la información de forma sencilla y entendible?

Sí, se entiende.

2. Dentro del menú, ¿es fácil orientarse para regresar al punto de inicio sin perderse en el intento?

Es muy fácil orientarse.

3. ¿Se echa en falta o sobra algún botón dentro de la navegación?

Sí se entienden fácilmente.

4. ¿Los textos son suficientemente descriptivos?

Botón de ajustes de juego.

JUGABILIDAD

1. Antes del inicio del juego se ofrece suficiente información sobre cómo jugar y cual es la historia del personaje?

Sí. Se entiende cómo se ha de jugar.

2. Una vez comienza el juego, ¿se ofrece suficiente información en pantalla sobre los elementos del juego (puntos, vidas, piruletas, etc.)?

Sí se ofrece suficiente información y clara.

3. La distribución de los elementos en pantalla, te parece adecuada?

Sí, se ve un diseño muy limpio.

4. ¿El nivel de dificultad del juego es demasiado fácil o excesivamente difícil?

Al principio del juego es algo difícil la maniobrabilidad del coche. Después de varias partidas va mejorando.

5. ¿El número de *items* (piruletas) es demasiado elevado?

En algunas zonas está muy saturado de piruletas.

6. ¿Durante la partida se ha encontrado algún error de juego?

Si vuelca el Cupcake Car desaparece la cabeza del personaje.

7. El tiempo de juego son 5 min. ¿Se prefiere que la partida sea más corta o más larga?

Se prefiere más corta.

FEEDBACK

3. Se recibe suficiente información sobre lo que va ocurriendo durante el juego (puntos positivos, puntos negativos, número de piruletas, inicio de juego, fin de juego, etc.).

Si, suficiente información. Pero me faltaría un indicador de número de vueltas realizadas.

4. ¿Los sonidos del juego ayudan a entender lo que está ocurriendo?.

Sí, los sonidos son acordes a la acción del juego.

UTILIDAD

5. ¿Se cree que este juego puede ser un buen método y una forma divertida de aprender el Lenguaje de Signos?

Sí podría ser, porque estás aprendiendo jugando y memorizando los signos inconscientemente.

6. El juego a parte de su aspecto educativo, ¿consigue entretener?

Sí, pero acaba siendo algo monótono. Habría que bajar el tiempo.

PREGUNTAS ABIERTAS

Posibles mejoras:

-En general el juego está muy bien, transmite una buena sensación y gráficamente es muy atractivo. La idea del mundo de golosinas me ha parecido muy acertada, pero considero que se podría mejorar los siguientes puntos:

- Mejorar el tiempo que tarda en ponerse en movimiento el coche.
- Añadir botón de opciones.
- Poder cambiar de música.
- Falta velocidad de arranque inicial.
- Falta un contador de vueltas.
- Excesivo tiempo de partida.

Finalmente, destacar que la idea del juego es buena pero hay que mejorar ciertos aspectos.

TEST USUARIO PIRUETO'S RACE GAME

Antes de comenzar con las preguntas sobre el juego, comentarle que puede que haya ciertas acciones que no podrá completar de forma óptima. Si eso ocurre, significará que el juego debe ser perfeccionado y la prueba será exitosa. Si todo sale bien, significará que el sitio fue diseñado pensando en el usuario y también que esta prueba alcanzó el éxito.

1. Nombre:

Jordi Zango Novell

2. ¿Qué experiencia tiene con ordenadores?:

Soy Graduado en Multimedia y usuario nivel experto.

3. ¿Juega normalmente a Juegos en el ordenador?

Sí

4. ¿Qué tipo de juegos le gustan?

Los de estrategia y los shoot'em up pero en línea y modo multijugador

INICIO TEST

IDENTIDAD

5. ¿Se identifica fácilmente el logotipo, el nombre del juego y autor?

Logotipo y nombre del juego sí. El de la autora no tanto si no es en los títulos de crédito.

6. ¿En términos generales, los elementos gráficos ayudan a identificar fácilmente una identidad corporativa que lo diferencia de otros juegos?

Sí, definen un estilo muy determinado.

7. ¿Se identifica claramente quien es el personaje principal del juego?

Sí, muy fácilmente.

8. Si se tuviese que poner en contacto con la persona propietaria del juego se ofrece información de contacto?

Sí, su correo electrónico.

ASPECTO GRÁFICO

1. ¿Es apropiado el aspecto gráfico para un juego de tipo infantil?

Sí, muy apropiado.

2. ¿El aspecto visual es atractivo?

Sí, es atractivo y agradable.

3. ¿Gráficamente: el juego está equilibrado, muy simple o muy recargado?

Está equilibrado.

NAVEGACIÓN

1. ¿La navegación del menú de inicio ofrece la información de forma sencilla y entendible?

Sí, ofrece toda la información necesaria.

2. Dentro del menú, ¿es fácil orientarse para regresar al punto de inicio sin perderse en el intento?

No

3. ¿Se echa en falta o sobra algún botón dentro de la navegación?

(Funcionalidad no disponible)

4. ¿Los textos son suficientemente descriptivos?

Sí, son descriptivos y fácilmente entendibles.

JUGABILIDAD

1. Antes del inicio del juego se ofrece suficiente información sobre cómo jugar y cual es la historia del personaje?

Sí, tanto en el menú principal como en la opción de ayuda.

2. Una vez comienza el juego, ¿se ofrece suficiente información en pantalla sobre los elementos del juego (puntos, vidas, piruletas, etc.)?

Sí

3. La distribución de los elementos en pantalla, te parece adecuada?

Sí, ofrece una interfaz cómoda.

4. ¿El nivel de dificultad del juego es demasiado fácil o excesivamente difícil?

Si se piensa en público infantil el nivel es el adecuado. Para usuarios mayores puede resultar un poco fácil.

5. ¿El número de *items* (piruletas) es demasiado elevado?

No, en mi opinión es el número adecuado para cumplir con el objetivo pedagógico del juego.

6. ¿Durante la partida se ha encontrado algún error de juego?

No

7. El tiempo de juego son 5 min. ¿Se prefiere que la partida sea más corta o más larga?

El tiempo me parece adecuado.

FEEDBACK

1. Se recibe suficiente información sobre lo que va ocurriendo durante el juego (puntos positivos, puntos negativos, número de piruletas, inicio de juego, fin de juego, etc.).

Sí, la interfaz ofrece suficiente información al respecto.

2. ¿Los sonidos del juego ayudan a entender lo que está ocurriendo?.

Sí, ayudan al usuario a saber lo que está haciendo.

UTILIDAD

1. ¿Se cree que este juego puede ser un buen método y una forma divertida de aprender el Lenguaje de Signos?

Sí, resulta bastante fácil ir aprendiendo sin darse cuenta.

2. El juego a parte de su aspecto educativo, ¿consigue entretener?

Sí

PREGUNTAS ABIERTAS

Posibles mejoras:

Una posible mejora sería la posibilidad de ajustar la velocidad del Piruetto, cosa que redundaría en conseguir un nivel de dificultad adecuado al nivel del jugador.

Cuando el Piruetto contacta con la valla se podría intentar reducir la fricción que hace que en algunos momentos se pierda un poco de tiempo intentando reanudar la marcha.

Estaría bien disponer de la posibilidad de activar/desactivar la música de fondo.

19. Versiones de la aplicación

19.1 Versión Alpha

La versión *Alpha* se ha utilizado para investigar el funcionamiento de Unity, para probar ideas, testear los elementos, y en definitiva familiarizarse con el entorno. La idea inicial era trabajar con *placeholders* para realizar las físicas del juego y después substituir por los objetos definitivos, aunque realmente no ha podido ser así ya que las físicas varían su comportamiento dependiendo del objeto que se trabaje. Así pues, el uso de *placeholders* se ha quedado reducido a algunos elementos secundarios del juego.

Después de tener la idea más clara, se ha trabajado el personaje principal y el Cupcake car sobre un plano, ya que son los elementos principales y más importantes del juego. Los objetos, después de haber sido modelado en *3DS Max*, se han importado a Unity, controlando que la importación fuera correcta, se han ajustado cosas que no terminaban de funcionar, se han aplicado las texturas definitivas y después se ha procedido a realizar la física del Cupcake car. Cuando todo ha funcionado, finaliza la fase Alpha y se pasa montar la fase Beta.



Figura 69: Aspecto Juego en fase Alpha. Personaje, Cupcake Car sobre plano con física realizada.

19.1 Versión Beta

Una vez en funcionamiento el Cupcake car, el siguiente elemento necesario para el desarrollo del juego son las piruletas, se realizan las piruletas buenas, malas y quitavidas. Para el montaje es necesario integrar un plano para colocar la textura de la lengua de signos, para su mejor visualización.

Posteriormente se incorpora un terreno con un *heightmap* de montañas, pero como se explica en apartados anteriores de la presente memoria, finalmente se opta por eliminar ya que aporta un

peso excesivo al proyecto, sin ser necesario. Se acaba substituyendo por un plano mucho más ligero que cumple el mismo cometido.

Posteriormente, se incorpora la iluminación, las cámaras, y el resto de elementos que comprenden la totalidad del escenario con sus componentes físicos (*colliders*, *triggers*, etc.). Paralelamente se trabaja la programación del juego. La parte con más dificultad del proyecto es enlazar todos los elementos para que la mecánica del juego quede bien ajustada y no hayan cabos sueltos.

Al mismo tiempo, también se trabaja la interfaz gráfica ya que la elaboración de ciertos elementos obligan a la elaboración de otros. La UI (Interfaz Gráfica) sufre diversos cambios por motivos de usabilidad y visualización. Concretamente se cambia el tamaño de las letras del alfabeto y la distribución de algunos elementos.

Posteriormente se realizan las animaciones, se incorporan los efectos especiales de sonido, y la música del juego. Y finalmente, se realiza el menú del juego en otra escena del proyecto unificándose después en la compilación, dando lugar a la primera versión Beta, totalmente funcional y con el aspecto gráfico acabado.



Figura 70: Aspecto Juego en fase Beta.

20. Requisitos de instalación/implantación/uso

En general el contenido desarrollado con Unity se ejecuta bastante bien en todos los dispositivos, aunque depende de la complejidad del proyecto.

Para ejecutar juegos de Unity es necesario:

Versión de escritorio:

- Sistema Operativo: Windows XP SP2+, Mac OS X 10.8+.
- Tarjeta de vídeo: capacidades DX9 (shader modelo 2.0); por lo general, todo lo que se haya lanzado desde 2004 debería funcionar.
- CPU: compatible con el conjunto de instrucciones SSE2.

Versión WebGL:

WebGL: Cualquier versión de escritorio reciente de Firefox, Chrome, Edge o Safari

**No se necesitan conocimientos previos ni formación.

21. Instrucciones de instalación/implantación

Unity es un software multiplataforma que compila para las diferentes plataformas. No es preciso ningún tipo de instalación previa, simplemente hay que ejecutar el archivo .EXE en el caso del Sistema Operativo *Windows*, o ejecutar el archivo .APP en el caso de OS X para MAC.

En relación a la aplicación para web inicialmente se había pensado exportar para Web Player, pero era necesario descargarse e instalar previamente un *plugin* (el Reproductor Web de *Unity*) en el navegador. Finalmente, se descarta esta tecnología por quedar obsoleta y además, será eliminada de versiones posteriores de Unity. En su lugar se exporta para *WebGL*, tecnología que utiliza el *canvas* de *HTML5* para mostrar gráficos 3D acelerados por *hardware*.

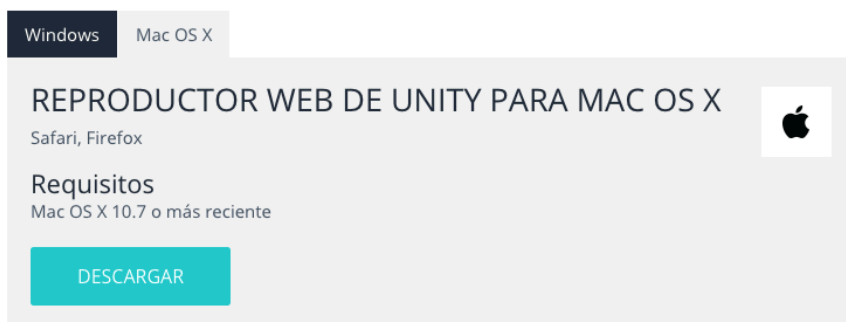


Figura 71: Reproductor Web de Unity (tecnología obsoleta)



Figura 72: Tecnología WebGL

22. Instrucciones de uso

Historia

Piruetto no sabe si es un gato o un conejo, pero es feliz comiendo piruletas de fresa, son su comida preferida. Diariamente viaja en su Cupcake Car buscando piruletas, pero ha de tener especial cuidado de no comerse las piruletas malas, ya que quitan vidas; o las incorrectas que restan puntos.

Durante todo el trayecto solamente debe comerse las que se le indican. Piruetto es muy ansioso y ha de tener cuidado de no correr demasiado porque volcando el Cupcake Car finaliza su viaje.



Cómo Jugar

Movimiento del Cupcake Car (o teclas WASD).



Colisiona con las piruletas que contengan la letra que se indique en cada momento. Las piruletas contienen dicha letra pero en lenguaje de signos, para saber cuales son pulsa la barra espaciadora.

23. Bugs

A continuación un informe de los errores encontrados en la aplicación, durante su fase de testeo en modo *Alpha* y *Beta*. Posteriormente a su publicación aparecerán nuevos errores y posibles mejoras que también habrá que anotar para mejorar la jugabilidad.

REFERENCIA	FECHA	DESCRIPCIÓN	GRAVEDAD	ESTADO
1	23/04/16	FÍSICAS: Aceleración del <i>Cupcake Car</i> algo lenta. El coche tarda mucho en coger velocidad cuando está parado.	MEDIA	PENDIENTE
2	23/04/16	CÁMARAS: Las cámaras 3ª persona presentan el problema de que dependiendo del ángulo de visión atraviesan las paredes de los objetos y se producen visiones indeseables, se ha de buscar la forma par evitar objetos que se encuentren entre el personaje y el punto de vista de la cámara. SOLUCIÓN: prácticamente se ha resuelto el problema. Únicamente aparece ocasionalmente cuando algún palo de piruleta se interpone entre la cámara y el personaje.	BAJA	MINIMIZADO
3	28/04/16	PUBLICACIÓN WEB: En la versión para web, se detecta que no se oyen algunos sonidos (botones y cuenta atrás regresiva) en el navegador <i>Chrome</i> . No ocurre en Safari ni Firefox. SOLUCIÓN: El error es producido porque el tamaño del archivo de sonido es tan pequeño que <i>Chrome</i> no lo llega a reproducir. Se ha aumentado la duración del sonido.	BAJA	SOLUCIONADO
4	28/05/16	FÍSICAS: Fricción en el muro del circuito. Cuando el coche choca contra el muro del circuito, en determinados momentos, se queda como "enganchado" en el muro. SOLUCIÓN: Para evitar esto, se ha creado un material físico al que se le ha añadido algo de fricción y rebote.	MEDIA	SOLUCIONADO
5	06/05/16	BOTÓN REPLAY: cuando se pulsa el botón, el juego vuelve a iniciar pero sin la animación de inicio. El error se encontraba en el valor de una variable booleana true/false.	BAJA	SOLUCIONADO

	SOLUCIÓN: Se cambia su valor al contrario.			
6	06/05/16	JUGABILIDAD: cuando se colisiona con una piruleta y surge la información del número de puntos que ha ganado o perdido, se ve una imagen indeseada.	BAJA	PENDIENTE
		SOLUCIÓN: Se cambia su valor al contrario.		
7	06/05/16	JUGABILIDAD: cuando el Cupcake car vuelca, ocasionalmente desaparece la cabeza del personaje por debajo del suelo.	BAJA	PENDIENTE

24. Proyección a futuro

- Utilización del proyecto como posible método de aprendizaje Lengua de signos, en todos aquellos centros, asociaciones, federaciones interesados en herramientas tecnológicas para el aprendizaje de la Lengua de signos.
- Estudiar como adaptar el juego a dispositivos móviles ya que según las encuestas realizadas es el dispositivo que más se emplea para jugar. Formará fuera del ámbito del proyecto ya que inicialmente no se había concebido para esta plataforma.
- Una posible extensión del juego sería: Transformar el juego a Realidad Virtual VR utilizando la API de Google *Cardboard* y el SDK de *Android Studio* que aportará un valor añadido.
- Otra posible extensión: Desarrollar un prototipo 3D de gafas VR estilo *Kawaii*, acorde con el juego para su posterior impresión en 3D.
- Estudiar crear varios niveles de juego con diferentes dificultades. En un primer momento se podría ofrecer el nivel actual como gratuito y ofrecer la posibilidad de jugar a otros niveles más personalizados a través de un pequeño pago.
- Utilizar el mismo sistema para el aprendizaje de palabras con contenido temático en la Lengua de signos.
- Investigar la posibilidad de adaptar el programa para otros procesos de aprendizaje y/o fines sociales (por ejemplo: autismo).

25. Presupuesto

Puesto que el videojuego se desarrolla íntegramente por 1 sola persona y el programa con el que se ha realizado posee una Licencia Gratuita, los costes de realización serían cero. Así que para la realización del presupuesto se ha simulado ser un estudio independiente o (*Indie*), que trabaja sin ceñirse a las órdenes de ninguna editora.

PRESUPUESTO DESARROLLO VIDEOJUEGO

	HORAS	PRECIO/ HORA	TOTAL
Designer (Gameplay)	40 horas	40 Euros	1600€
Artist (Game Art)	32 horas	30 Euros	960€
Programmer (Scripting)	100 horas	60 Euros	6000€
Level Designer	10 horas	40 Euros	400€
Sound Engineer	16 horas	30 Euros	480€
Tester	30 horas	22 Euros	660€
		Subtotal	10.100 €
		IVA 21%	2.121 €
		TOTAL	12.221 €

** Cualquier modificación sobre lo pactado supondrá un plus de 65 Euros/hora.

Tabla 3: Presupuesto.

26. Análisis de mercado

"El mercado de los videojuegos accesibles o dirigidos a personas con discapacidades no es un mercado saturado. El 15% de la población de la población mundial sufre algún tipo de discapacidad y en España un 8,5% por lo que hay suficiente target en el mercado para desarrollar videojuegos de este tipo. El motivo principal de que existan pocas iniciativas es la falta de concienciación, información y/o limitación de recursos." (Gamer 2013)²⁷:

Como se señala anteriormente, en términos genéricos el mercado de los videojuegos accesibles y/o dirigidos a personas con discapacidades está todavía pendiente de explotar y existe suficiente público objetivo como para enfocar los videojuegos hacia este sector.

En España, como se comenta más detalladamente en el punto... de la presente memoria, existen 2 videojuegos a destacar: el proyecto Iredia: El Secreto de Atram donde el juego presenta un mundo donde el sonido varía de diferentes formas para que el jugador pueda experimentar y aprender sobre los diferentes tipos de sordera. Y por otro lado el programa E-mintza desarrollado por la Fundación Orange apoyado por el Ministerio de Industria, Turismo y Comercio dirigido a personas con Autismo pero la técnica se podría utilizar también para aprender el Lenguaje de Signos (inspiración para el proyecto).

En cuanto a juegos y/o aplicaciones dedicadas más concretamente al aprendizaje de la Lengua de Signos, donde se enmarca este proyecto, se encuentran diversas iniciativas como canales *Youtube* donde se exponen videos para el aprendizaje de la Lengua de Signos, páginas web con juegos muy simples y a nivel de gráficos poco acabados, y en cuanto a aplicaciones podemos encontrar: aprendizaje de las letras a través de *flash cards*²⁸, traductores²⁹, diccionarios³⁰ y glosarios de términos³¹. En conclusión, existe mercado suficiente para desarrollar videojuegos en este sector, además de que es un sector pendiente de desarrollo por falta de información y concienciación. Juegos y/o aplicaciones para el aprendizaje LSE hay algunos como los mencionados anteriormente, pero ninguno de características similares en las que se enmarca este proyecto.

²⁷ **Gamer. Artículo Videojuegos y accesibilidad:** ¿hay juegos para todos?. Redacción FS Gamer. 21 enero,2013. <http://www.fsgamer.com/videojuegos-y-accesibilidad-hay-juegos-para-todos-20130121.html> [en línea] [Fecha de consulta: 01/05/16]

²⁸ **Flash cards:** <https://play.google.com/store/apps/details?id=com.teachersparadise.aslamericansignlanguage&hl=es>

²⁹ **Traductores** <https://play.google.com/store/apps/details?id=air.com.kikedao.signslator&hl=es>

³⁰ **Diccionarios Fundación Vodafone España** <https://play.google.com/store/apps/details?id=es.fundacionvf.dilse.cnse&hl=es>

³¹ **Glosario Términos** <https://play.google.com/store/apps/details?id=com.fundacionvf.avatares&hl=es>

27. Marketing y Ventas

El videojuego es un proyecto de tipo social y no tiene un objetivo económico detrás, no se espera que directamente genere beneficios por sí mismo, ya que la idea es difundirlo de forma gratuita entre las diferentes Asociaciones, Federaciones y Organizaciones relacionadas con la Lengua de Signos. Ver si el producto funciona y gusta, y a partir de ahí, estar en disposición de realizar otros proyectos similares para estas organizaciones.

Paralelamente se investigará si es posible adquirir algún tipo de subvención por parte del Estado como han logrado algunos productos mencionados en la presente memoria, como el programa E-mintza para investigar y/o desarrollar productos multimedia en este ámbito respaldado por la Fundación Orange³². Se ha visto que tanto Orange como Vodafone tienen bastantes iniciativas de este tipo³³.

El juego está disponible desde los servidores de Google, en su versión para ordenador (*Mac* y *Windows*), concretamente se pondrán a descargar desde *Google Drive*.

Posteriormente, si se realiza su adaptación a dispositivos móviles, se alojará en el *Play Store*. Además, el proyecto también se quiere adaptar a VR. Todo ello quedará para una fase posterior del TFG, ya que inicialmente no estaba concebido como un proyecto para *Mobile*, pero durante el proceso de trabajo se ha visto que sería un plus muy valioso.

La promoción del juego se realizará a través una página web personalizada del videojuego donde se podrá jugar on-line o descargar el producto. Se realizará un estudio de las principales Asociaciones y Organizaciones y se trazará un plan de Marketing consistente en ponerse en contacto por vía e-mail para dar a conocer la existencia del producto y ver si dichas organizaciones estarían interesadas en incluirlo en su sitio web y de paso saber un poco que necesidades tienen para realizar proyectos futuros.

³³ **Fundación Vodafone.** Discapacidad. <http://www.fundacionvodafone.es/>

³² **Orange Fundación.** <http://www.fundacionorange.es/>

28. Conclusión/-es

En el proceso de elaboración de un videojuego normalmente interviene un equipo multidisciplinar de profesionales especializados en diferentes áreas: Programadores, Modeladores, Animadores 3D, Artistas, Ilustradores, Músicos, Directores de Arte, etc.

Asumir el rol de todos ellos en espacio tan reducido de tiempo, además de tener que aprender paralelamente el uso del software Unity y el lenguaje de programación *C Sharp* ha sido una tarea ardua y con un volumen de trabajo bastante importante. Algunos momentos han sido especialmente duros sobre todo la parte de producción donde el diseño y la mecánica del juego han de cobrar forma y sentido.

El motivo de la elección de este proyecto es precisamente poder trabajar en una actividad multidisciplinaria, donde desarrollar y poner en práctica un proyecto muy personal y dar uso a la mayoría de competencias digitales adquiridas en el Grado Multimedia: programación, diseño gráfico, modelado 3D, animación, sonido, música etc. Además del plus que representa realizar un proyecto con un fin social.

El hecho de enmarcar el proyecto en la mención de Comunicación Visual y Creatividad ha servido para dar un enfoque más visual y creativo. La idea era crear un proyecto en el ámbito de la comunicación con un objetivo diferente al mero entretenimiento, en un intento de dar uso de las nuevas tecnologías para dar forma a un proyecto social, sensible, educativo y de concienciación ciudadana.

Finalmente, destacar que habiendo visto apenas un mero destello del amplio camino que queda por recorrer, la experiencia ha sido muy motivadora y gratificante. Posiblemente sea un punto de partida de algún proyecto social a nivel profesional ya que este proyecto ha despertado mi interés por los videojuegos enfocados a personas con discapacidades.

Anexo 1. Entregables del proyecto

- **Memoria_Sonia_Alvarez_Campa.pdf** → Memoria del Proyecto.
- **WEB_Sonia_Alvarez.zip** → Versión WEB del juego y también puede verse en:
<http://multimedia.com/box/piruetos/>
- **PiruetosRace_MAC.zip** → Versión Escritorio para MAC OS.
- **PiruetosRace_WIN.zip** → Versión Escritorio para WINDOWS.
- **Libro_Estilo_Sonia_Alvarez.pdf** → Manual de Identidad. Documento externo para su mejor visualización y presentación.
- **Libro_Como_Jugar_Sonia_Alvarez.pdf** → Manual de Cómo Jugar. Documento externo para su mejor visualización y presentación.
- **Game_Design_Document_Sonia_Alvarez.pdf** → Mécanicas del Juego. Documento externo para su mejor visualización.
- **ASSETS_Sonia_Alvarez.zip** → Todos los archivos que se han creado y que han sido necesario para la elaboración del juego.
- **Autoinforme.pdf** → Valoración competencias transversales.
- **MASTER.zip** → Otros archivos originales que se han hecho servir en el proyecto.

Anexo 2. Código fuente (extractos)

- El Script *LoadGame* se encarga de gestionar el Menú de Inicio del juego. Primeramente se inicializa la pantalla de Inicio quedando la pantalla de créditos y la de *GameOver* ocultas, se acciona la música de introducción, y si se presiona en el botón de *Play* se activa la función *Play* que carga la escena del juego y detiene la música de introducción. La función *Exit* permite salir del juego, la función *Credits* activa la pantalla de créditos, la función *Back* muestra la pantalla de inicio de nuevo.

```
//SCRIPT PARA CONTROLAR EL MENÚ DE INICIO
public class LoadGame : MonoBehaviour {

    public GameObject credits;
    public GameObject help;
    private AudioSource player;
    public AudioClip sound;

    void Awake(){
        credits.SetActive (false);
        help.SetActive (false);
        player = gameObject.GetComponent<AudioSource> ();
    }

    //BOTÓN JUGAR
    public void Play(string x){
        SceneManager.LoadScene (x, LoadSceneMode.Single);
        player.clip = sound;
        player.Play ();
    }

    //BOTÓN SALIR
    public void Exit(){
        player.Stop ();
        Application.Quit ();
        player.clip = sound;
        player.Play ();
    }

    //BOTÓN CREDITS
    public void Credits(){
        help.SetActive (false);
        credits.SetActive (true);
        player.clip = sound;
        player.Play ();
    }

    //BOTÓN BACK (DESDE CRÉDITS)
    public void Back(){
        credits.SetActive (false);
        player.clip = sound;
        player.Play ();
    }
}
```

```

//BOTÓN BACK (DESDE HELP)
public void Help(){
    credits.SetActive (false);
    help.SetActive (true);
    player.clip = sound;
    player.Play ();
}

//BOTÓN BACK (DESDE HELP)
public void HBack(){
    help.SetActive (false);
    player.clip = sound;
    player.Play ();
}
}

```

- Función OnTriggerEnter(): Controla los eventos que ocurrirán al colisionar con un objeto. En este caso controla con que Piruletas (buenas, malas, quitavidas) colisionará el Cupcake Car y su destrucción. También activa la suma puntos del marcador y un aviso visual conforme se suman o restan puntos. Si es el caso de las piruletas Quitavidas, además de restar puntos se desactiva una vida en la pantalla. Esta función forma parte de la mecánica principal del juego.

```

void OnTriggerEnter(Collider other){
    if (CountDown.EndCount && CountDown.endGame == false) {

        //PIRULETAS BUENAS

        if (other.tag == "piruleta_b" && gameObject != null) {
            cuentaPuntos += 100; // suma 100 puntos
            cuentaPiruletas_b -= 1;
            points.text = cuentaPuntos.ToString () + " Points";
            shadowPoints.text = cuentaPuntos.ToString () + " Points";
            mostrarSumaPuntos (); // muestra por pantalla +100
            pb += 1; // añade 1 piruleta al contador de piruletas buenas
            piruletas_b.text = pb.ToString () + " Good";
            shadowPiruletas_b.text = pb.ToString () + " Good"; //sombra
            player.clip = soundLollipop [0]; //activa el sonido piruleta buena
            player.Play ();
            Destroy (other.gameObject);

            //PIRULETAS MALAS

        } else if (other.tag == "piruleta_m" && gameObject != null) {
            cuentaPuntos -= 100;
            points.text = cuentaPuntos.ToString () + " Points";
            shadowPoints.text = cuentaPuntos.ToString () + " Points";
            mostrarRestaPuntos ();
            pm += 1;
            piruletas_m.text = pm.ToString () + " Bad";
            shadowPiruletas_m.text = pm.ToString () + " Bad";
            player.clip = soundLollipop [2];
            player.Play ();
        }
    }
}

```

```
        Destroy (other.gameObject);

        //PIRULETAS QUITAVIDAS

    } else if (other.tag == "quitavidas" && gameObject != null) {
        controlVidas [contadorVidas].SetActive (false);
        contadorVidas += 1;
        player.clip = soundLollipop [1];
        player.Play ();
        InitQuitaVidas ();
        Destroy (other.gameObject);
    }

    if (cuentaPiruletas_b <= 0 || cuentaPuntos <= 0 || contadorVidas >= 3) {
        InitGameOver ();
    }
}
}
```


- Estructura condicional para controlar que cada x tiempo cambie la letra que se debe buscar. Todas las letras del alfabeto están guardadas en un array y cada 20 segundos cambia la letra que se debe buscar. Al mismo tiempo se carga dicha letra en las piruletas que serán las piruletas correctas, aunque en realidad lo que cambia es la textura del objeto. Dichas texturas están guardadas en otro array y su índice será el número aleatorio que surge a partir de la función `Random.Range()`. Seguidamente se activa el sonido de esa letra que previamente ha sido grabado.

```

if (t % 20 == 0) {

    //RANDOM PARA PIRULETAS BUENAS
    n = (int)Random.Range (0, 27);

    transcripcionLetra.texture = texturas [n];

    GetComponent<AudioSource> ().clip = abcAudio [n];
    GetComponent<AudioSource> ().Play();
    GetComponent<AudioSource> ().loop = false;

    //CAMBIAR LETRA
    letra_buena = GetComponent<Text> ();
    letra_buena.text = abecedario [n]

    //PIRULETAS BUENAS
    for(int i = 0; i < right.Length; i++){
        if (right [i] != null) {
            right [i].GetComponent<Renderer> ().material.mainTexture = texturas[n];
        }
    }
}

```

- Para controlar que piruletas serán incorrectas se utiliza el mismo procedimiento que para las piruletas correctas pero se generan otros números aleatorios controlando que la letra que ha salido para las piruletas correctas, no pueda volver a salir en estas.

```

for(int i = 0; i < wrong.Length; i++){
    if (wrong [i] != null) {
        //RANDOM PIRULETAS MALAS
        m = (int)Random.Range (0, 27);
        while (m == n) {
            m = (int)Random.Range (0, 27);
        }
        wrong [i].GetComponent<Renderer> ().material.mainTexture = texturas [m];
    }
}

```

- Se han creado 2 corrutinas: para controlar las imágenes que deben aparecer cada vez que se suman o se restan puntos y que después deben desaparecer. Estas funciones mantienen ocultas dichas imágenes y se vuelven visibles durante 1 segundo cuando hay suma o resta de puntos. Una corrutina es una función que tiene la habilidad de pausar su ejecución y devolver el control a Unity para luego continuar donde lo dejó en el siguiente frame.

```
//CORRUTINAS MOSTRAR +100 Y -100 DURANTE 1 SEGUNDO

public void mostrarSumaPuntos(){
    StartCoroutine (EmpezarCuentaAtras1());
}

public IEnumerator EmpezarCuentaAtras1(){
    masPuntos.CrossFadeAlpha (1, 0.1f, false);
    yield return new WaitForSeconds (1);
    masPuntos.CrossFadeAlpha (0, 0, false);
}

public void mostrarRestaPuntos(){
    StartCoroutine (EmpezarCuentaAtras2());
}

public IEnumerator EmpezarCuentaAtras2(){
    menosPuntos.CrossFadeAlpha (1, 0.1f, false);
    yield return new WaitForSeconds (1);
    menosPuntos.CrossFadeAlpha (0, 0, false);
}
```

Anexo 3. Librerías/Código externo utilizado

- Se utiliza un script de seguimiento de cámara que lleva incorporado Unity. Su cometido es que la cámara siga al coche, cuando esté en movimiento. **SmoothFollow.cs**:

```

using UnityEngine;

namespace UnityStandardAssets.Utility
{
    public class SmoothFollow : MonoBehaviour
    {
        // The target we are following
        [SerializeField]
        private Transform target;
        // The distance in the x-z plane to the target
        [SerializeField]
        private float distance = 10.0f;
        // the height we want the camera to be above the target
        [SerializeField]
        private float height = 5.0f;

        [SerializeField]
        private float rotationDamping;
        [SerializeField]
        private float heightDamping;

        // Use this for initialization
        void Start() { }

        // Update is called once per frame
        void LateUpdate()
        {
            // Early out if we don't have a target
            if (!target)
                return;

            // Calculate the current rotation angles
            var wantedRotationAngle = target.eulerAngles.y;
            var wantedHeight = target.position.y + height;

            var currentRotationAngle = transform.eulerAngles.y;
            var currentHeight = transform.position.y;

            // Damp the rotation around the y-axis
            currentRotationAngle = Mathf.LerpAngle(currentRotationAngle, wantedRotationAngle,
            rotationDamping * Time.deltaTime);

            // Damp the height
            currentHeight = Mathf.Lerp(currentHeight, wantedHeight, heightDamping * Time.deltaTime);

            // Convert the angle into a rotation
            var currentRotation = Quaternion.Euler(0, currentRotationAngle, 0);

            // Set the position of the camera on the x-z plane to:
            // distance meters behind the target
            transform.position = target.position;
            transform.position -= currentRotation * Vector3.forward * distance;

            // Set the height of the camera
            transform.position = new Vector3(transform.position.x, currentHeight, transform.position.z);

            // Always look at the target
            transform.LookAt(target);
        }
    }
}

```

- Para la realización del Script: **SimpleCarController.cs** se sigue el *Unity Manual* apartado *WheelCollider Tutorial* y se utiliza el código recomendado para el control del movimiento del coche a partir de los colliders de las ruedas.

```

[System.Serializable]
public class AxleInfo {
    public WheelCollider leftWheel;
    public WheelCollider rightWheel;
    public bool motor;
    public bool steering;
}

public class SimpleCarController : MonoBehaviour {
    public List<AxleInfo> axleInfos;
    public float maxMotorTorque;
    public float maxSteeringAngle;

    // finds the corresponding visual wheel
    // correctly applies the transform
    public void ApplyLocalPositionToVisuals(WheelCollider collider)
    {
        if (collider.transform.childCount == 0) {
            return;
        }

        Transform visualWheel = collider.transform.GetChild(0);

        Vector3 position;
        Quaternion rotation;
        collider.GetWorldPose(out position, out rotation);

        visualWheel.transform.position = position;
        visualWheel.transform.rotation = rotation;
    }

    public void FixedUpdate(){

        if(CountDown.EndCount && CountDown.endGame==false){

            float motor = maxMotorTorque * Input.GetAxis ("Vertical");
            float steering = maxSteeringAngle * Input.GetAxis ("Horizontal");

            foreach (AxleInfo axleInfo in axleInfos) {
                if (axleInfo.steering) {
                    axleInfo.leftWheel.steerAngle = steering;
                    axleInfo.rightWheel.steerAngle = steering;
                }
                if (axleInfo.motor) {
                    axleInfo.leftWheel.motorTorque = motor;
                    axleInfo.rightWheel.motorTorque = motor;
                }
                ApplyLocalPositionToVisuals (axleInfo.leftWheel);
                ApplyLocalPositionToVisuals (axleInfo.rightWheel);
            }
        }
    }
}

```

Anexo 4. Capturas de pantalla

Ampliación Imagen Boceto

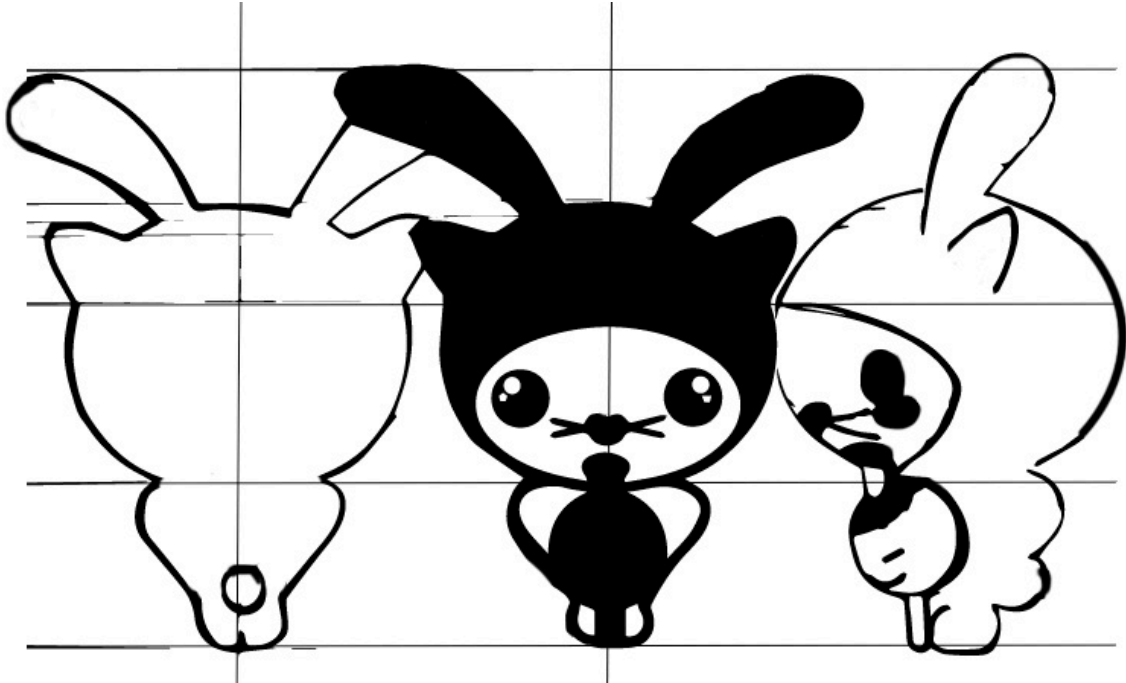


Figura 73: Ampliación Imagen Boceto Piruetto.

Ampliación Imagen Vistas Modelado Piruetto

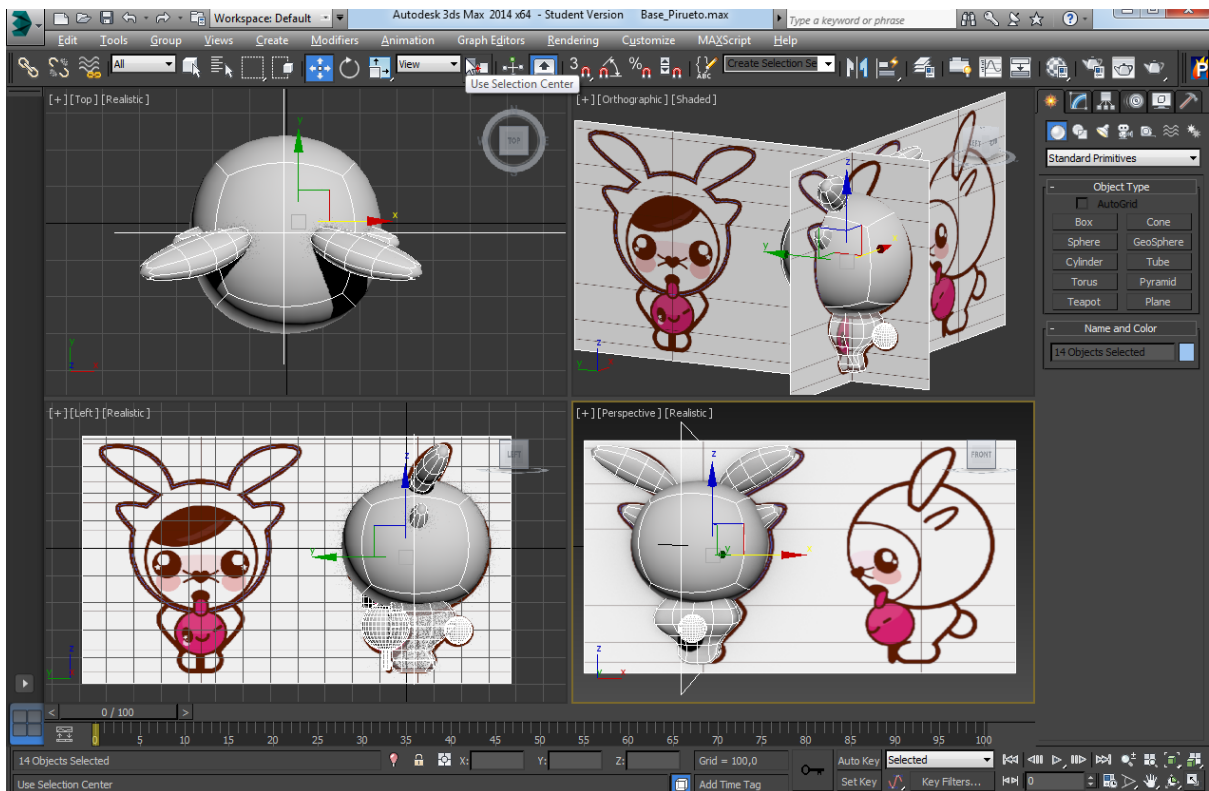


Figura 74: Ampliación Imagen Vistas 3D Piruetto.

Ampliación modelado Cupcake

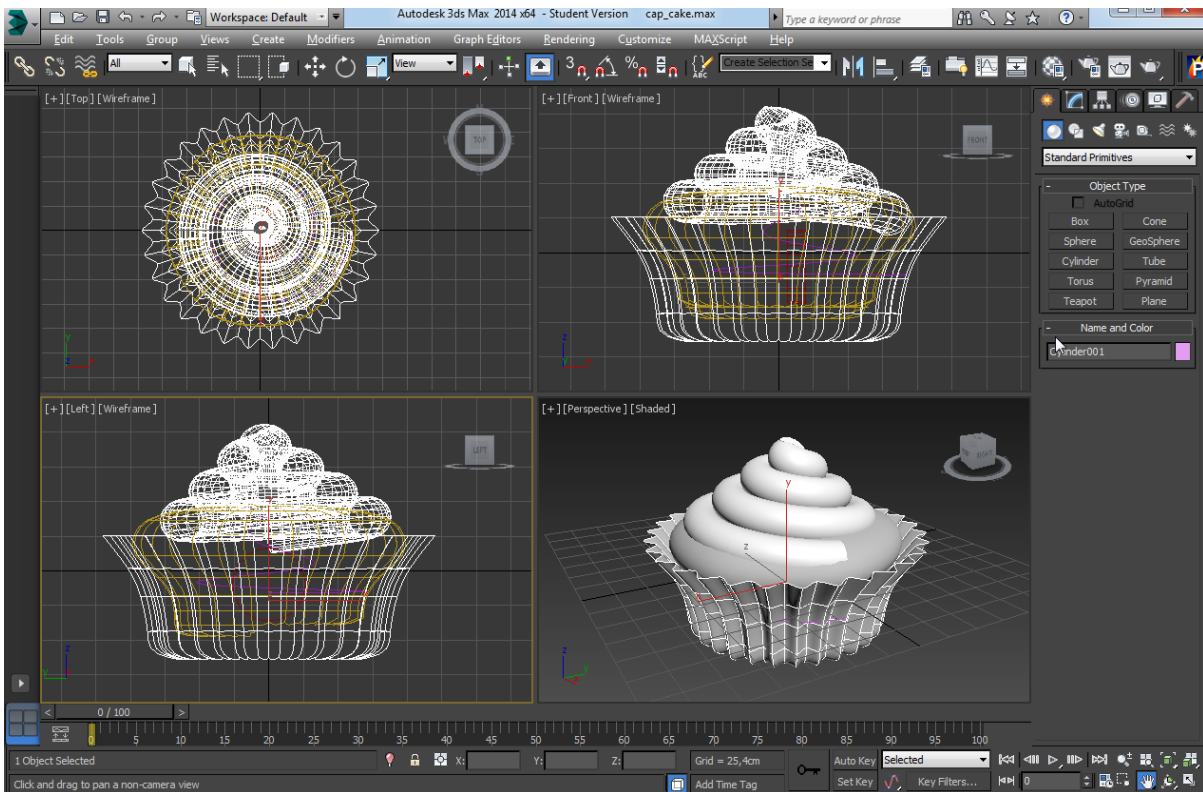


Figura 75: Ampliación Imagen modelado Cupcake en 3DS Max.

Aspecto final de Pirueto incorporado en Unity



Figura 76: Ampliación imagen aspecto Final Pirueto incorporado en Unity.

Aspecto final de Piruetto & Cupcake Car incorporado en Unity



Figura 77: Ampliación imagen aspecto final Piruetto & Cupcake Car incorporados en Unity.

Vista ampliada modelado circuito

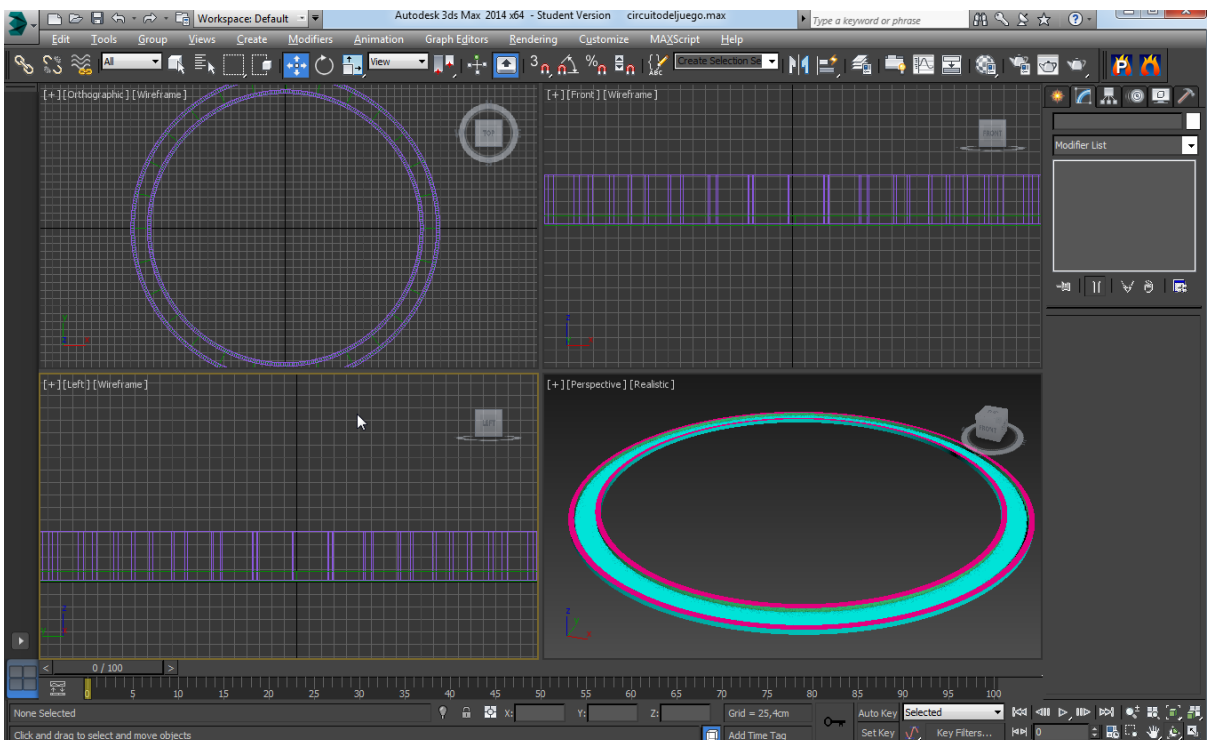


Figura 78: Ampliación imagen circuito en 3DS Max.

Vista ampliada modelado piruletas

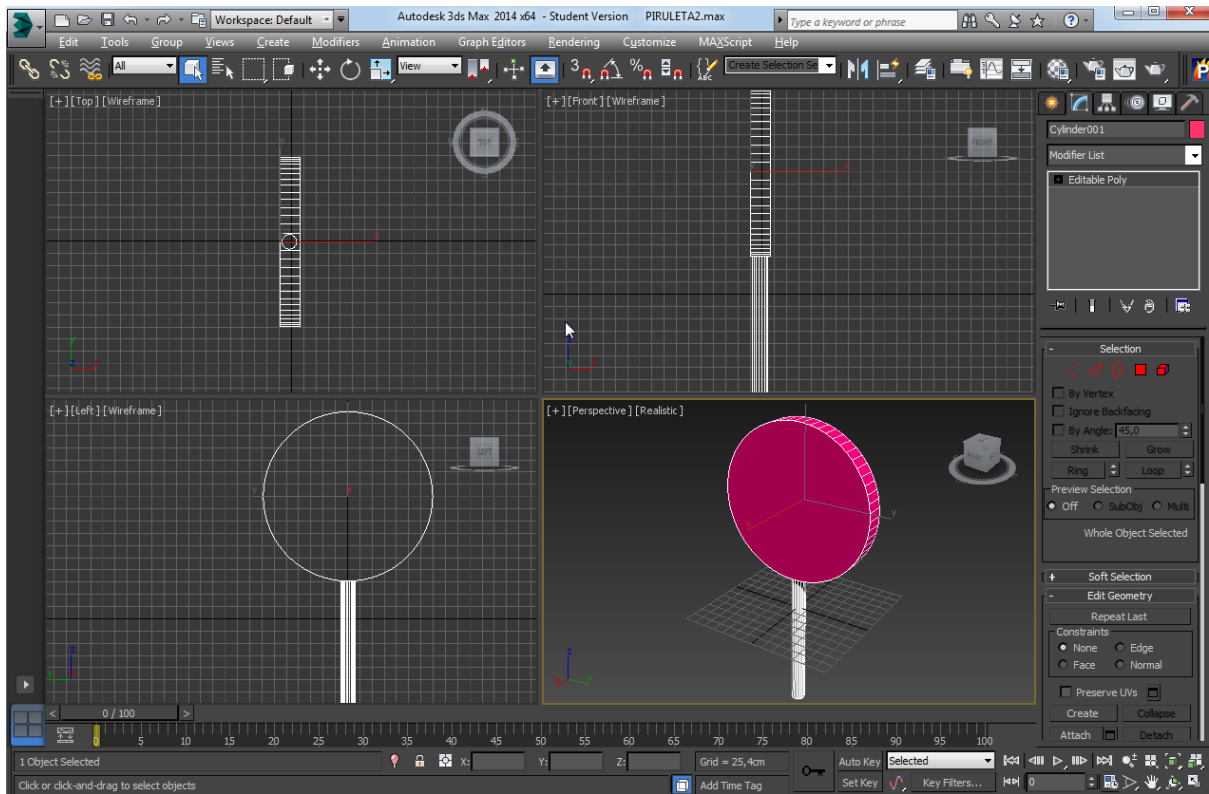


Figura 79: Ampliación imagen modelado piruletas en 3DS Max.

Vista ampliada modelado de Chupa-Chups.

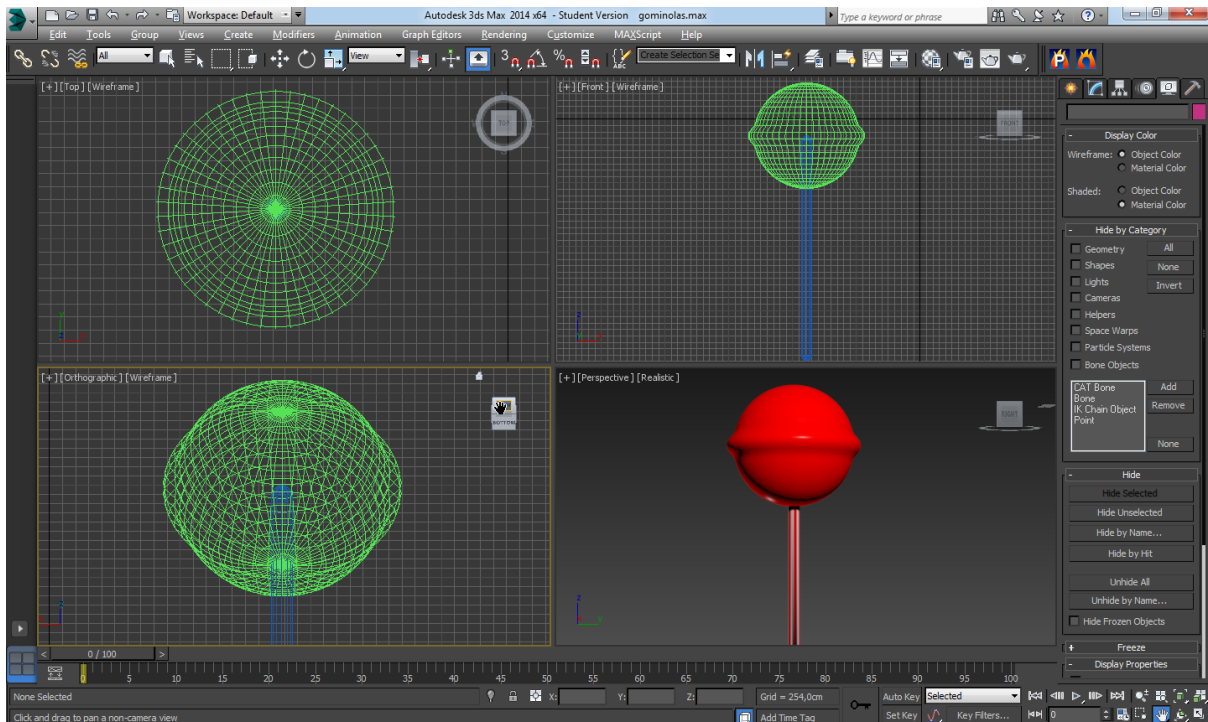


Figura 80: Ampliación modelado Chupa-Chups.

Vista ampliada modelado nubes de nata y fresa

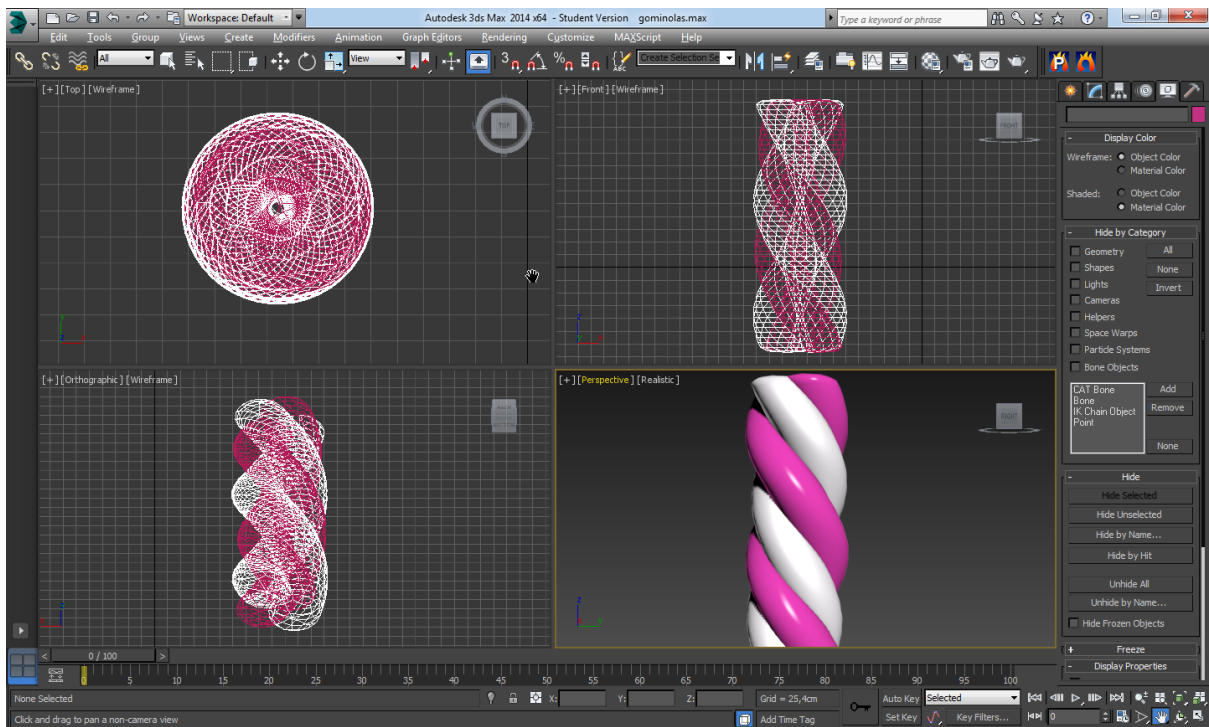


Figura 81: Ampliación modelado nubes de fresa y nata.

Vista ampliada modelado de galletas (elemento que finalmente no se ha utilizado)

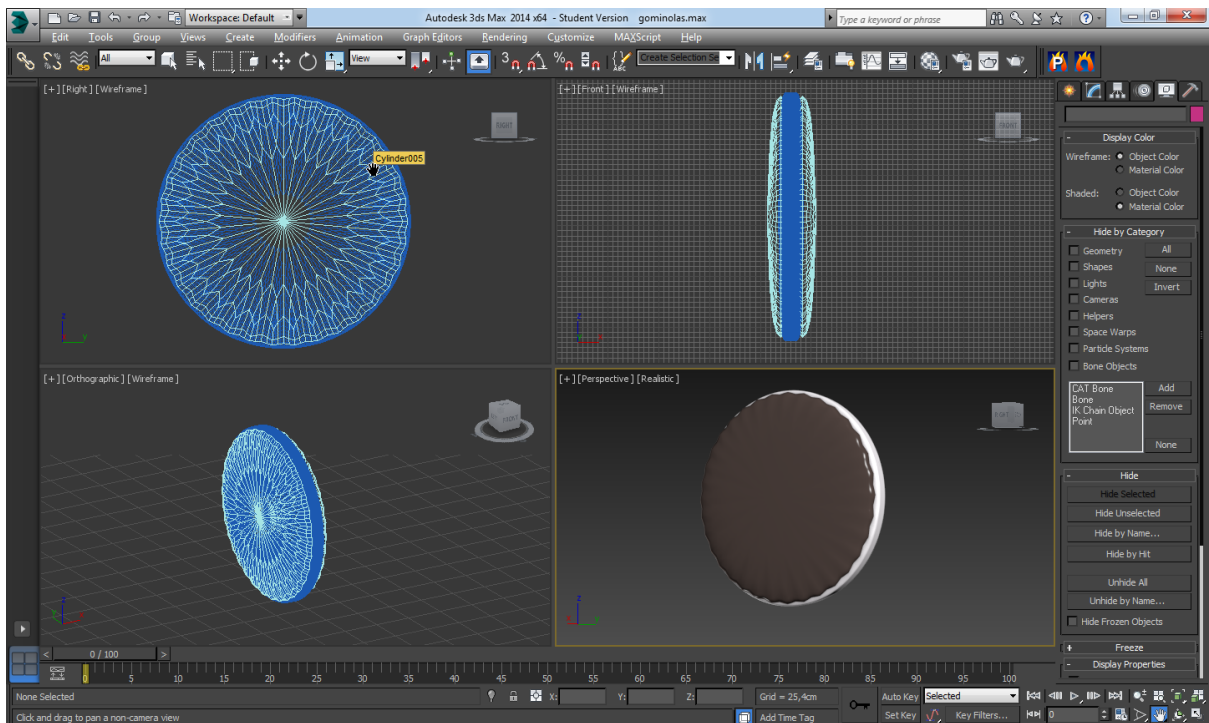


Figura 82: Ampliación modelado de galletas en 3DS Max.

Vista ampliada modelado chocolate (elemento que finalmente no se ha utilizado)

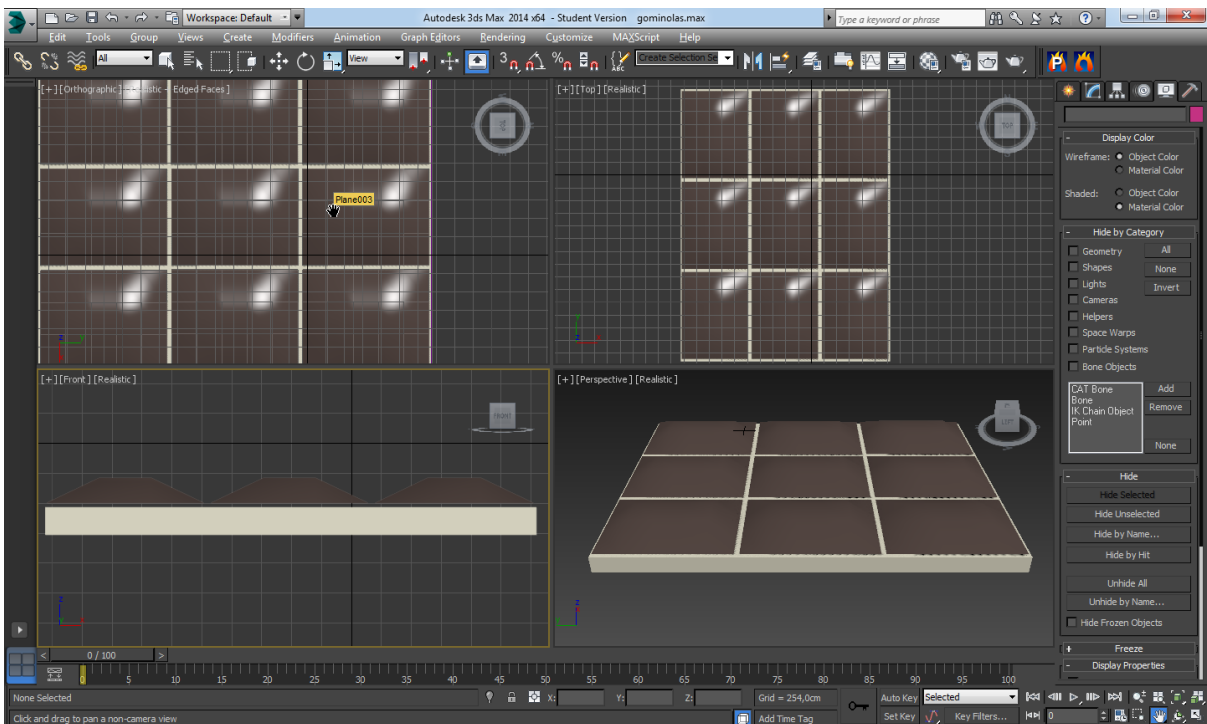


Figura 83: Ampliación modelado chocolate en 3DS Max.

Entorno de trabajo en Unity



Figura 84: Aspecto entorno de trabajo Unity.

Anexo 5. Guía de usuario

Infografía sobre cómo jugar a Piruetto's Race 3D

1

timer
0:00

vidas

Dashboard con los controles de juego

puntos
nº errores
nº aciertos

2

La partida acaba a los 5 min.

Si se pierden las 3 vidas finaliza la partida

Durante la partida se pedirán letras en el panel de control y el usuario tendrá que colisionar con aquellas piruletas cuya letra coincida con la que se pide. En las piruletas las letras se presentan en la Lengua de Signos.

Si se pierden todos los puntos finaliza la partida

3

Si se pulsa ayuda (barra espaciadora), se verá su signo correspondiente en el panel de control.

Otras formas de finalizar el juego:

- Se consiguen todas las piruletas.
- Se vuelca el Cupcake Car

Anexo 6. Libro de estilo



Isotipo (parte icónica)



Isologo (texto e iconos fundidos)



Imagotipo (parte icónica + textual)



**Manual Identidad
Piruetto's Race**



Manual Identidad Piruetto's Race

Tipografía

Tipografía Porky's Logotipo y elementos importantes del juego

0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz

Tipografía Anja Eliane accent Normal Tipografía para textos

0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz

* Dichas tipografías no poseen cursivas ni negritas



Manual Identidad Piruetto's Race

Colores

**MARRÓN
CHOCOLATE**



RGB= 83, 35, 19
CMYK= 53% 96% 100% 42%
HEX #532313

**ROSA
PIRULETA SUAVE**



RGB= 202, 105, 160
CMYK= 12% 79% 9% 0%
HEX #CA69A0

**ROSA
PIRULETA FUERTE**



RGB= 191, 49, 113
CMYK= 12% 95% 32% 0%
HEX #BF3171

**VERDE
MENTA**



RGB= 10, 207, 65
CMYK= 93% 0% 100% 0%
HEX #0ACF41

**BLANCO
NATA**



RGB= 255, 255, 255
CMYK= 0% 0% 0% 0%
HEX #FFFFFF

**Manual Identidad
Piruetto's Race**



Uso Correcto



Usos Incorrectos

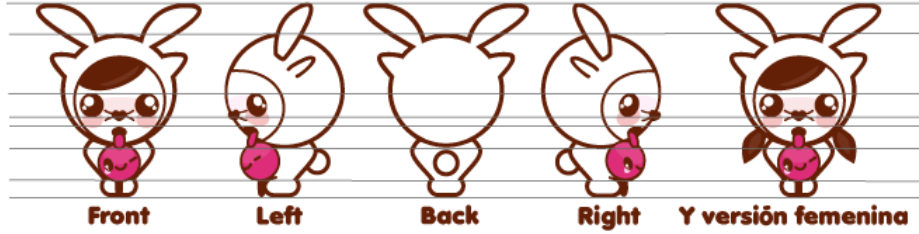




Manual Identidad Piruetto's Race

Personaje

Turnaround Piruetto



Variaciones de color

MARRÓN
CHOCOLATE



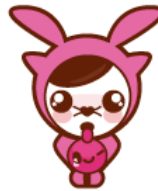
RGB= 63, 35, 19
CMYK= 53% 96% 100% 42%
HEX #532313

ROSA
PIRULETA FUERTE



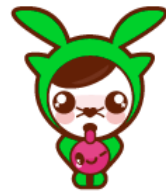
RGB= 191, 49, 113
CMYK= 12% 95% 32% 0%
HEX #BF3171

ROSA
PIRULETA SUAVE



RGB= 202, 105, 160
CMYK= 12% 79% 9% 0%
HEX #CA69A0

VERDE
MENTA



RGB= 10, 207, 65
CMYK= 93% 0% 100% 0%
HEX #0ACF41



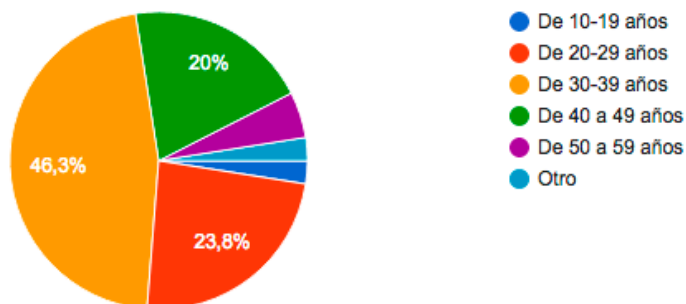
Manual Identidad Piruetto's Race

Elementos de la interfaz gráfica

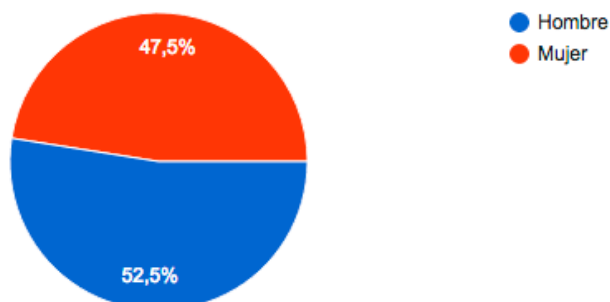


Anexo 8. Encuesta Previa

1. ¿Cuál es tu edad? (80 respuestas)

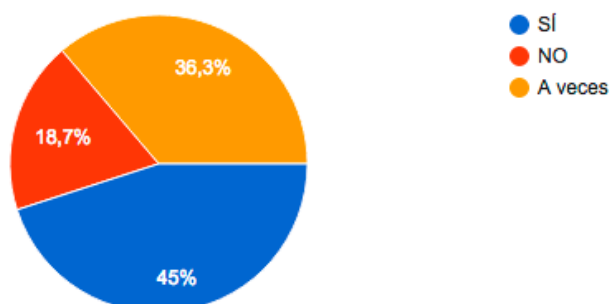


2. ¿Cuál es tu género? (80 respuestas)

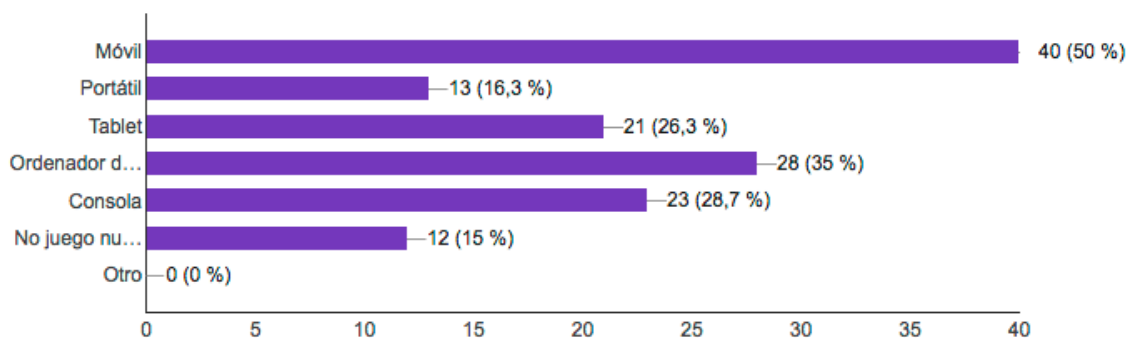


3. ¿Te gusta jugar de forma habitual en el ordenador, móvil, consola o cualquier otro dispositivo?

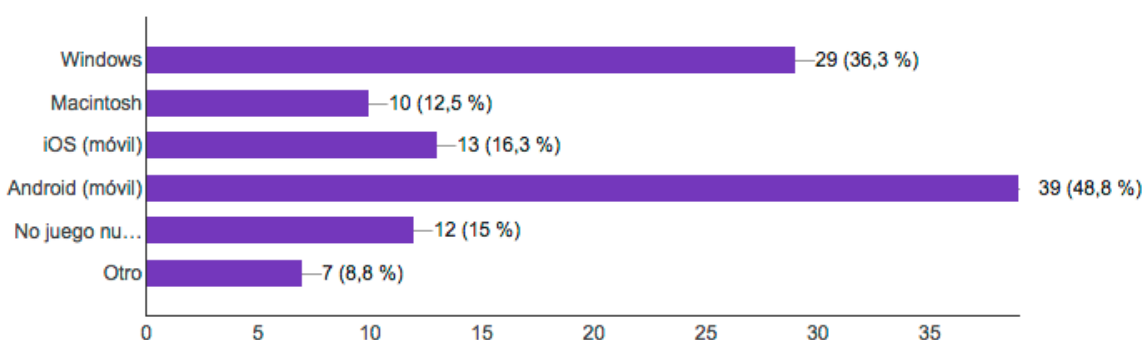
(80 respuestas)



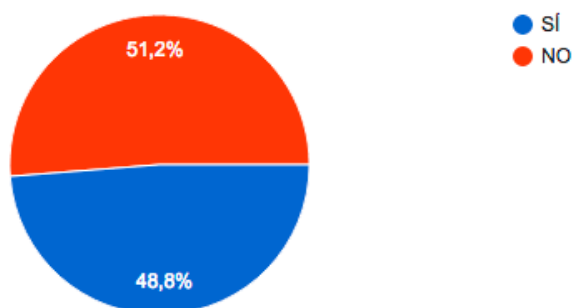
4. ¿En qué dispositivos prefieres jugar? (80 respuestas)



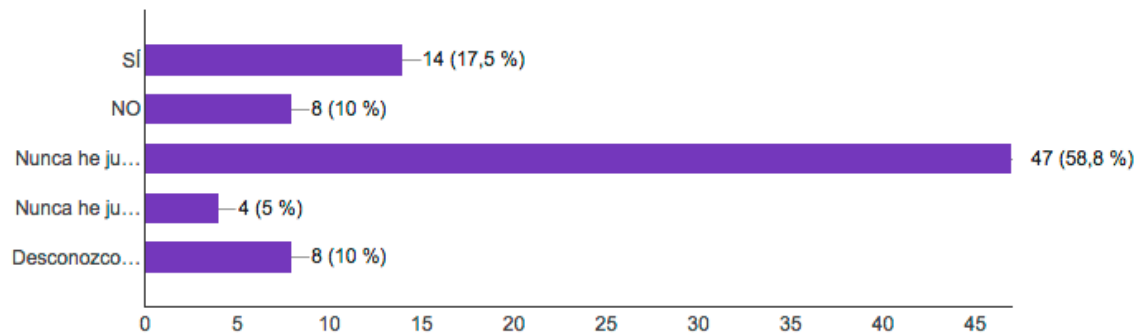
5. ¿Que Sistema Operativo utilizas para jugar? (80 respuestas)



6. ¿Te gustan los juegos en formato web, para jugar desde el navegador? (80 respuestas)

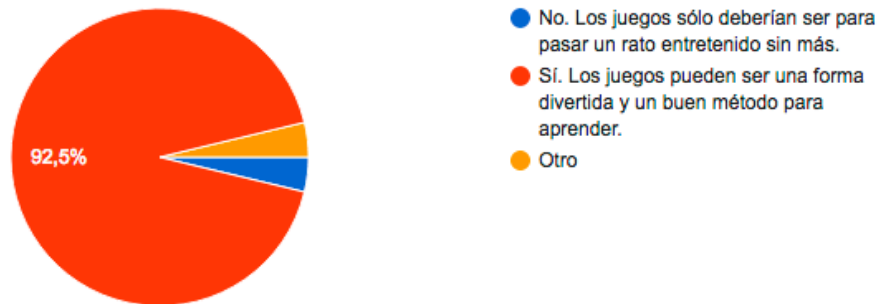


7. ¿Te gusta jugar a juegos en Realidad Virtual a través del móvil? (80 respuestas)



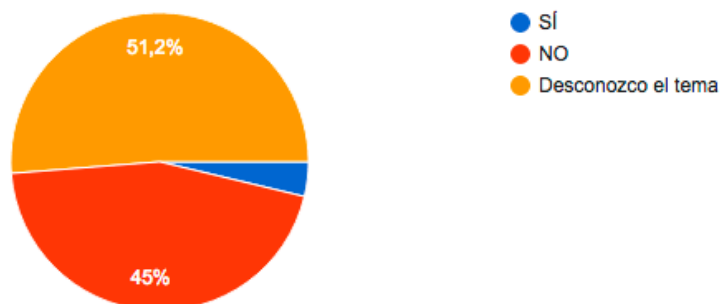
8. ¿Crees que es una buena idea el aprendizaje a través del juego? (80 respuestas)

(80 respuestas)



9. ¿Crees que en el mercado existen suficientes juegos educativos enfocados a personas con alguna discapacidad? (80 respuestas)

(80 respuestas)



Encuesta TFG (respuestas)

NÚMERO	FECHA Y HORA	PREGUNTA 1	PREGUNTA 2	PREGUNTA 3	PREGUNTA 4	PREGUNTA 5	PREGUNTA 6	PREGUNTA 7	PREGUNTA 8	PREGUNTA 9
1	4/04/2016 10:36:51	De 20-29 años	Mujer	A veces	Móvil, Portátil, Ordenador de sobremesa	Macintosh	Sí	Sí	Si. Los juegos pueden s	NO
2	4/04/2016 10:56:19	De 20-29 años	Hombre	Sí	Ordenador de sobremesa	Windows	NO	NO	Si. Los juegos pueden s	Desconozco el tema
3	4/04/2016 10:57:25	De 20-29 años	Hombre	Sí	Móvil, Ordenador de sobremesa	Windows, Android (móvil)	Sí	Nunca he jugado pero m	Si. Los juegos pueden s	NO
4	4/04/2016 11:03:00	De 30-39 años	Mujer	Sí	Móvil	Android (móvil)	Sí	Nunca he jugado pero m	Si. Los juegos pueden s	Desconozco el tema
5	4/04/2016 11:03:15	De 40 a 49 años	Hombre	A veces	Consola	X	NO	NO	Si. Los juegos pueden s	Desconozco el tema
6	4/04/2016 11:04:52	De 30-39 años	Mujer	A veces	Móvil, Consola	Macintosh, iOS (móvil)	NO	Desconozco el tema.	Si. Los juegos pueden s	Desconozco el tema
7	4/04/2016 11:08:50	De 40 a 49 años	Hombre	A veces	Tablet, Consola	Windows, iOS (móvil)	Sí	Nunca he jugado pero m	Si. Los juegos pueden s	NO
8	4/04/2016 11:13:11	De 30-39 años	Mujer	Sí	Portátil, Tablet, Consola	iOS (móvil), Ps3	NO	Nunca he jugado pero m	Si. Los juegos pueden s	NO
9	4/04/2016 11:16:25	De 40 a 49 años	Hombre	A veces	Consola	Sony	NO	Sí	Si. Los juegos pueden s	NO
10	4/04/2016 11:17:36	De 40 a 49 años	Hombre	Sí	Móvil, Tablet, Ordenador de sobremesa	Macintosh, iOS (móvil),	Sí	Nunca he jugado pero m	Si. Los juegos pueden s	Desconozco el tema
11	4/04/2016 11:18:20	De 20-29 años	Mujer	A veces	Ordenador de sobremesa	Windows	Sí	Nunca he jugado pero m	Si. Los juegos pueden s	NO
12	4/04/2016 11:19:46	De 40 a 49 años	Hombre	A veces	Móvil, Portátil, Consola	Macintosh, iOS (móvil)	Sí	Nunca he jugado pero m	Si. Los juegos pueden s	NO
13	4/04/2016 11:23:11	De 20-29 años	Mujer	Sí	Móvil, Ordenador de sobremesa, Consola	Windows, Android (móvil)	Sí	Nunca he jugado pero m	Si. Los juegos pueden s	Desconozco el tema
14	4/04/2016 11:24:48	De 30-39 años	Mujer	A veces	Tablet	Android (móvil)	Sí	Sí	Si. Los juegos pueden s	NO
15	4/04/2016 11:36:18	De 30-39 años	Hombre	Sí	Móvil, Ordenador de sobremesa, Consola	Windows, iOS (móvil), A	Sí	Sí	Si. Los juegos pueden s	NO
16	4/04/2016 11:39:59	De 40 a 49 años	Hombre	A veces	Móvil	Android (móvil)	Sí	Sí	Si. Los juegos pueden s	Desconozco el tema
17	4/04/2016 11:41:02	De 30-39 años	Mujer	Sí	Tablet, Ordenador de sobremesa	Windows, Android (móvil)	Sí	Sí	Si. Los juegos pueden s	NO
18	4/04/2016 11:43:30	De 20-29 años	Mujer	Sí	Móvil, Portátil, Tablet, Ordenador de sobremesa	Windows, Android (móvil)	Sí	Nunca he jugado pero m	Si. Los juegos pueden s	NO
19	4/04/2016 11:45:23	De 20-29 años	Hombre	Sí	Ordenador de sobremesa, Consola	Windows	NO	Nunca he jugado pero m	Si, pero siempre que el	Desconozco el tema
20	4/04/2016 11:53:42	De 30-39 años	Hombre	A veces	No juego nunca	No juego nunca	NO	Nunca he jugado pero m	No. Los juegos sólo deb	Desconozco el tema
21	4/04/2016 11:57:18	De 30-39 años	Mujer	A veces	Ordenador de sobremesa	Windows	NO	Nunca he jugado pero m	Si. Los juegos pueden s	Desconozco el tema
22	4/04/2016 11:58:27	De 50 a 59 años	Mujer	NO	No juego nunca	No juego nunca	Sí	Nunca he jugado pero m	Si. Los juegos pueden s	Desconozco el tema
23	4/04/2016 11:58:47	De 20-29 años	Hombre	Sí	Consola	Windows	NO	Nunca he jugado pero m	Si. Los juegos pueden s	Desconozco el tema
24	4/04/2016 12:00:19	De 40 a 49 años	Mujer	Sí	Móvil	Android (móvil)	Sí	NO	Si. Los juegos pueden s	NO
25	4/04/2016 12:06:08	De 40 a 49 años	Mujer	NO	No juego nunca	No juego nunca	NO	Desconozco el tema.	Si. Los juegos pueden s	NO
26	4/04/2016 12:10:06	De 40 a 49 años	Mujer	NO	Móvil	Android (móvil)	Sí	Nunca he jugado pero m	Si. Los juegos pueden s	Desconozco el tema
27	4/04/2016 12:10:09	De 40 a 49 años	Hombre	Sí	Móvil, Ordenador de sobremesa	Macintosh, Android (mó)	Sí	Nunca he jugado pero m	Si. Los juegos pueden s	Desconozco el tema
28	4/04/2016 12:11:26	De 20-29 años	Mujer	Sí	Móvil, Consola	Android (móvil)	Sí	Sí	Si. Los juegos pueden s	NO
29	4/04/2016 12:19:34	De 50 a 59 años	Hombre	A veces	Portátil, Ordenador de sobremesa	Windows	Sí	Nunca he jugado y tamp	Si. Los juegos pueden s	NO
30	4/04/2016 12:19:55	De 30-39 años	Hombre	Sí	Móvil, Consola	Android (móvil), Playstat	NO	Nunca he jugado pero m	Si. Los juegos pueden s	Desconozco el tema
31	4/04/2016 12:22:54	De 30-39 años	Mujer	A veces	Portátil, Tablet	Macintosh, Android (mó)	Sí	Nunca he jugado pero m	Si. Los juegos pueden s	Desconozco el tema
32	4/04/2016 12:24:29	De 30-39 años	Mujer	NO	No juego nunca	No juego nunca	NO	NO, Nunca he jugado pe	Si. Los juegos pueden s	Desconozco el tema
33	4/04/2016 12:28:14	De 40 a 49 años	Hombre	NO	No juego nunca	No juego nunca	NO	Nunca he jugado pero m	Si. Los juegos pueden s	NO
34	4/04/2016 12:32:55	De 30-39 años	Hombre	NO	No juego nunca	No juego nunca	NO	Nunca he jugado pero m	Si. Los juegos pueden s	Desconozco el tema
35	4/04/2016 12:39:17	De 30-39 años	Mujer	NO	No juego nunca	No juego nunca	NO	NO	Si. Los juegos pueden s	Sí
36	4/04/2016 12:43:38	De 20-29 años	Mujer	Sí	Ordenador de sobremesa	Windows	NO	Nunca he jugado y tamp	Si. Los juegos pueden s	NO
37	4/04/2016 12:50:49	De 50 a 59 años	Hombre	A veces	Ordenador de sobremesa	Macintosh	Sí	Sí	Si. Los juegos pueden s	NO
38	4/04/2016 13:06:07	De 30-39 años	Hombre	A veces	Ordenador de sobremesa, Consola	Windows	NO	Nunca he jugado pero m	Si. Los juegos pueden s	Desconozco el tema
39	4/04/2016 13:24:09	De 30-39 años	Mujer	A veces	Móvil, Tablet	Android (móvil)	Sí	Nunca he jugado pero m	Si. Los juegos pueden s	Desconozco el tema
40	4/04/2016 13:24:47	De 30-39 años	Hombre	Sí	Móvil, Consola	Windows, Android (móvil)	NO	NO	Si. Los juegos pueden s	NO
41	4/04/2016 13:43:21	De 30-39 años	Mujer	Sí	Móvil, Portátil, Ordenador de sobremesa	Macintosh, iOS (móvil)	NO	NO	Si, siempre y cuando no	Desconozco el tema

Pirueto's Race, Grado Multimedia, Sonia Alvarez

Encuesta TFG (respuestas)

NÚMERO	FECHA Y HORA	PREGUNTA 1	PREGUNTA 2	PREGUNTA 3	PREGUNTA 4	PREGUNTA 5	PREGUNTA 6	PREGUNTA 7	PREGUNTA 8	PREGUNTA 9
42	4/04/2016 13:46:44	De 30-39 años	Mujer	NO	No juego nunca	No juego nunca	Sí	Nunca he jugado pero m	Si. Los juegos pueden s	Desconozco el tema
43	4/04/2016 13:47:29	De 30-39 años	Hombre	Sí	Ordenador de sobremesa, Consola	Windows, Android (móvil	Sí	Nunca he jugado pero m	Si. Los juegos pueden s	Desconozco el tema
44	4/04/2016 13:49:06	De 30-39 años	Hombre	Sí	Móvil, Portátil, Tablet, Ordenador de sobremesa, Consol	Windows, iOS (móvil), A	NO	Nunca he jugado y tamp	Si. Los juegos pueden s	NO
45	4/04/2016 13:52:17	De 20-29 años	Hombre	Sí	Móvil, Tablet	Macintosh, Android (mó	NO	Nunca he jugado pero m	Si. Los juegos pueden s	NO
46	4/04/2016 13:52:45	De 30-39 años	Mujer	NO	No juego nunca	No juego nunca	NO	Nunca he jugado y tamp	Si. Los juegos pueden s	Desconozco el tema
47	4/04/2016 13:56:01	De 30-39 años	Hombre	Sí	Móvil	iOS (móvil)	NO	Nunca he jugado pero m	No. Los juegos sólo deb	Desconozco el tema
48	4/04/2016 13:57:37	De 10-19 años	Hombre	Sí	Tablet, Consola	Windows, Android (móvil	Sí	Nunca he jugado pero m	Si. Los juegos pueden s	Desconozco el tema
49	4/04/2016 14:05:00	De 20-29 años	Hombre	Sí	Móvil, Tablet, Ordenador de sobremesa	Windows, Android (móvil	Sí	Sí	Si. Los juegos pueden s	Sí
50	4/04/2016 14:28:09	De 30-39 años	Hombre	Sí	Consola	ps4	Sí	Sí	Si. Los juegos pueden s	NO
51	4/04/2016 14:30:13	De 40 a 49 años	Hombre	NO	Móvil, Portátil, Tablet, Ordenador de sobremesa	Windows, Android (móvil	Sí	Nunca he jugado pero m	Si. Los juegos pueden s	Desconozco el tema
52	4/04/2016 14:56:26	De 10-19 años	Mujer	A veces	Tablet	Android (móvil)	NO	Nunca he jugado pero m	Si pienso que son una b	NO
53	4/04/2016 15:01:07	De 30-39 años	Hombre	A veces	Ordenador de sobremesa	Windows	Sí	Nunca he jugado pero m	Si. Los juegos pueden s	NO
54	4/04/2016 15:14:21	De 30-39 años	Hombre	A veces	Móvil, Tablet	Android (móvil)	NO	Nunca he jugado pero m	Si. Los juegos pueden s	Desconozco el tema
55	4/04/2016 15:33:17	De 40 a 49 años	Mujer	Sí	Móvil, Tablet	Android (móvil)	Sí	Sí	Si. Los juegos pueden s	NO
56	4/04/2016 15:40:33	De 50 a 59 años	Hombre	A veces	Móvil, Tablet	iOS (móvil)	NO	Nunca he jugado pero m	Si. Los juegos pueden s	Desconozco el tema
57	4/04/2016 15:47:12	De 30-39 años	Mujer	A veces	Móvil, Tablet	Android (móvil)	NO	Nunca he jugado pero m	Si. Los juegos pueden s	NO
58	4/04/2016 15:52:52	De 20-29 años	Hombre	Sí	Consola	Playstation 4	NO	Sí	Si. Los juegos pueden s	NO
59	4/04/2016 16:15:36	De 30-39 años	Mujer	A veces	Portátil	Windows	NO	Nunca he jugado pero m	Si. Los juegos pueden s	Desconozco el tema
60	4/04/2016 16:39:10	De 30-39 años	Mujer	NO	Móvil	Android (móvil)	NO	Nunca he jugado pero m	Si. Los juegos pueden s	NO
61	4/04/2016 16:40:26	De 20-29 años	Mujer	Sí	Móvil, Portátil	Macintosh, iOS (móvil)	NO	Nunca he jugado pero m	Si. Los juegos pueden s	NO
62	4/04/2016 16:41:15	63	Mujer	NO	No juego nunca	No juego nunca	NO	Desconozco el tema.	Si. Los juegos pueden s	Desconozco el tema
63	4/04/2016 16:46:44	De 20-29 años	Hombre	Sí	Tablet	iOS (móvil)	NO	Nunca he jugado pero m	Si. Los juegos pueden s	NO
64	4/04/2016 17:40:13	De 20-29 años	Hombre	A veces	Móvil, Portátil	Windows, iOS (móvil)	Sí	Desconozco el tema.	Si. Los juegos pueden s	Desconozco el tema
65	4/04/2016 17:47:16	De 40 a 49 años	Hombre	A veces	Consola	ps4	NO	Nunca he jugado pero m	Si. Los juegos pueden s	NO
66	4/04/2016 17:48:42	+ 60	Hombre	NO	Móvil, Ordenador de sobremesa	Windows, Android (móvil	Sí	Desconozco el tema.	Si. Los juegos pueden s	Sí
67	4/04/2016 17:51:30	De 30-39 años	Mujer	NO	No juego nunca	No juego nunca	NO	Nunca he jugado pero m	Si. Los juegos pueden s	Desconozco el tema
68	4/04/2016 18:05:34	De 30-39 años	Mujer	A veces	Móvil	Android (móvil)	NO	Desconozco el tema.	Si. Los juegos pueden s	NO
69	4/04/2016 18:40:20	De 40 a 49 años	Mujer	NO	No juego nunca	No juego nunca	NO	Desconozco el tema.	Si. Los juegos pueden s	Desconozco el tema
70	4/04/2016 18:49:42	De 40 a 49 años	Hombre	Sí	Móvil, Tablet, Ordenador de sobremesa	Windows, Android (móvil	Sí	Sí	Si. Los juegos pueden s	Desconozco el tema
71	4/04/2016 18:52:01	De 30-39 años	Mujer	A veces	Móvil	Android (móvil)	Sí	NO	Si. Los juegos pueden s	NO
72	4/04/2016 18:56:06	De 30-39 años	Hombre	Sí	Móvil	Android (móvil)	NO	Desconozco el tema.	No. Los juegos sólo deb	Desconozco el tema
73	4/04/2016 19:49:47	De 30-39 años	Mujer	Sí	Móvil, Consola	Android (móvil)	Sí	Nunca he jugado pero m	Si. Los juegos pueden s	Desconozco el tema
74	4/04/2016 20:35:11	De 30-39 años	Mujer	A veces	Móvil	Android (móvil)	NO	Nunca he jugado pero m	Si. Los juegos pueden s	Desconozco el tema
75	4/04/2016 20:58:12	De 20-29 años	Hombre	Sí	Ordenador de sobremesa	Windows	NO	Nunca he jugado pero m	Si. Los juegos pueden s	NO
76	4/04/2016 21:17:37	De 30-39 años	Mujer	A veces	Móvil, Ordenador de sobremesa	Windows, Android (móvil	Sí	Nunca he jugado pero m	Si. Los juegos pueden s	NO
77	4/04/2016 21:22:16	De 20-29 años	Hombre	Sí	Ordenador de sobremesa, Consola	Windows, Android (móvil	Sí	Sí	Si. Los juegos pueden s	Desconozco el tema
78	5/04/2016 0:07:12	De 20-29 años	Mujer	Sí	Portátil, Ordenador de sobremesa, Consola	Windows	Sí	Nunca he jugado pero m	Si. Los juegos pueden s	NO
79	5/04/2016 1:00:13	De 30-39 años	Hombre	A veces	Móvil, Tablet	Android (móvil)	NO	Nunca he jugado pero m	Si. Los juegos pueden s	Desconozco el tema
80	5/04/2016 9:39:24	De 30-39 años	Hombre	Sí	Móvil	Android (móvil)	Sí	Nunca he jugado pero m	Si. Los juegos pueden s	Desconozco el tema

Piruetto's Race

Game Design Document

15 de marzo de 2016

Revisión 2

Sonia Álvarez Campá

Índice

1.	Introducción	
1.1.	Concepto del juego	121
1.2.	Características principales	121
1.3.	Género	121
1.4.	Propósito y público objetivo	122
1.5.	Jugabilidad	122
1.6.	Estilo visual	123
1.7.	Alcance	123
2.	Mecánicas del juego	
2.1	Jugabilidad	124
2.2	Flujo de juego:	124
2.3	Personaje	125
2.4	Movimiento y físicas:	126
2.5	Controles:	126
3.	Interfaz	
3.1	Diagrama Flujo:	127
3.2	Menú principal	129
3.3	Créditos:	130
3.4	Final Juego:	131
4.	Arte	
1.8.	Arte 2D	132
1.9.	Arte 3D	132
1.10.	Animación	133
1.11.	Sonido	133

1. Introducción

Este es el documento de diseño de Piruetto's Race. Un videojuego para PC, MAC y Web realizado con el motor de videojuegos Unity versión 5.3.2f1 Personal. Este escrito tiene como objetivo plasmar los elementos que debe incluir Piruetto's Race y servir de carta de presentación en caso de buscar colaboradores o financiación en un futuro.

1.1 Concepto del juego

Piruetto's Race es un juego en el que se controla al personaje de la historia: Piruetto, que debe su nombre en honor a que le encantan las Piruletas de sabor de fresa. Piruetto se moverá mediante un coche, a través de un circuito intentando recopilar el mayor número posible de piruletas. Durante el transcurso del juego se le irá indicando cuales debe recoger.

1. 2. Características principales

El juego se basa en:

- **Planteamiento sencillo:** Debido a que el producto está enfocado a todo tipo de usuarios pero principalmente se potencia el juego infantil, se ha elegido una historia sencilla para poder llevar a cabo el objetivo del juego: el aprendizaje.
- **Educación:** Mediante la elección correcta o incorrecta de piruletas con las letras del lenguaje de signos se realiza un proceso de aprendizaje y se inculcan valores como la obediencia y recompensa.
- **Acción:** Es un juego dinámico, con cierta sensación de tensión ya que pone al usuario en una carrera en la que tendrá que tomar decisiones que le otorgaran o restaran puntos.

1. 3 Género

Atendiendo a las características del juego, Piruetto's Race se enmarca dentro de varios géneros:

Principalmente:

Educativo: Ya que está dirigido a aportar conocimientos y/o habilidades al usuario durante el transcurso del mismo. Tiene una función didáctica que combina la formación con entretenimiento.

Secundariamente:

Acción: El usuario tiene que hacer uso de sus reflejos, y habilidad para conducir el coche y para seleccionar las piruletas correctas. Además ha de superar una serie de obstáculos (piruletas que quitan vidas).

Simulación de vehículos: De forma indirecta también es un videojuego que permite al jugador operar de forma sencilla con un vehículo.

1. 4 Propósito y público objetivo

El objetivo principal es ofrecer un producto de entretenimiento, diversión y al mismo tiempo aprendizaje del lenguaje dactilológico mientras se va desempeñando la partida.

El juego está orientado a un amplio rango de edades pero principalmente se dirige a un público infantil. Se trata de un juego sencillo que debe ser jugable por cualquier tipo de público.

1. 5 Jugabilidad

Como se ha comentado anteriormente, se trata de un juego sencillo enfocado al aprendizaje, así que únicamente dispondrá de un nivel de juego, aunque fácilmente se podrían implementar otros niveles si fuera necesario.

Movilidad: El personaje principal se moverá a través del escenario en 3ª persona, con las flechas de teclado y/o teclas específicas que se le indiquen.

Recompensas: Piruletas que suman puntos. Son las piruletas correctas que se piden durante el juego. De esta forma el jugador sabe que lo está realizando bien.

Obstáculos: Piruletas que restan puntos. Todas aquellas que no sean correctas.

Quita Vidas: Piruletas que directamente quitaran vidas. Son aquellas que bajo ningún concepto se han de recoger.

1.6 Estilo visual

Infantil tipo *cartoon*, concretamente se trabajará el estilo *Kawaii*, ya que es un estilo bastante sencillo de trabajar que se caracteriza por la reducción de sus elementos a lo esencial.

La idea es trabajar la estética infantil a través del uso de formas muy simples, redondas y amables, colores son cálidos y vivos, contrastes y personajes simples, con el fin de evocar características infantiles y sentimientos de ternura.

El escenario estará formado por todo tipo de caramelos y dulces. Los colores serán vivos y las texturas serán simples.

1.7 Alcance

El objetivo principal es desarrollar un juego sólido con un primer nivel de juego que pueda ser ampliado en un futuro para cubrir las necesidades educativas de los usuarios.

2. Mecánicas del juego

En esta sección se explicará en detalle las mecánicas del juego y los pilares fundamentales del juego.

2.1 Jugabilidad

Niveles: Como se ha comentado anteriormente, inicialmente el juego partirá con 1 nivel de dificultad jugable por cualquier tipo de usuario.

Movimiento: El usuario tiene la posibilidad de moverse con total autonomía sobre las zonas libres del escenario. Lo cual le permitirá llegar a todas las piruletas que se le pidan. El movimiento del personaje será sobre el plano del suelo.

Habilidades: El usuario habrá de demostrar su habilidad a la hora de seleccionar las piruletas con las que debe colisionar (para recogerlas).

Recursos limitados: El jugador puede hacer uso de la ayuda de forma limitada. Para saber que letra ha de buscar dentro de la piruleta.

Progresión del jugador: El jugador progresará a medida que avance la partida y vaya sumando puntos.

2.2 Flujo de juego:

Empieza el juego con el Menú principal, donde aparecerán las siguientes opciones (para que el jugador seleccione la más apropiada):

1. Play.
2. Credits.
3. Help.
4. Exit (sólo versión de escritorio).

En el momento de inicio de la partida se verá una cuenta regresiva (3, 2,1) mientras se muestra una pequeña animación de la cámara mostrando un plano general alrededor del personaje.

Una vez comience el juego, Piruetto se moverá con las teclas indicadas, por el circuito y deberá ir colisionando con aquellas piruletas que se le pidan durante el transcurso del juego. Dichas piruletas serán las que lleven una imagen del lenguaje de signos que se esté pidiendo en ese momento. En la pantalla principal de juego se mostrará la letra en forma de alfabeto tradicional y en las piruletas se visualizará en forma de signo. El usuario puede hacer uso de la ayuda para saber cual es el signo que debe buscar entre las piruletas. Cuando se colisione con las piruletas correctas, se sumaran puntos y cuando no sean correctas, se restaran. Si se colisiona con piruletas prohibidas se pierde una vida.

La partida termina cuando el personaje se quede sin vida, en puntos negativos, vuelque su vehículo o transcurran 5 minutos desde el inicio del juego.

2.3 Personaje

El personaje protagonista de la historia es Piruetto, que debe su nombre en honor a que su comida preferida son las piruletas de fresa. Piruetto, no sabe muy bien si es un gato o un conejo pero es feliz comiendo piruletas. Su apariencia es sencilla, joven y simpática, fresca, basada en el estilo Kawaii.

En general estas son las características principales del personaje:

Cabeza: También se ha hecho uso de la exageración de ciertos elementos como es la cabeza. La cabeza del personaje es grande y redonda. Su tamaño puede constituir es incluso mayor que del cuerpo.

Cara: Los ojos se colocan proporcionalmente bajos en la cara y en el cráneo, así la frente es muy ancha.

Los ojos: Están muy lejos el uno del otro.

Boca: Boca pequeña.

Dientes: Ausencia de dientes para evocar la figura de los bebés.

Brazos y Piernas: Hay una reducción de elementos: no presenta brazos, ni manos ni pies.

Vida Inicial:	3 vidas
Puntos Iniciales:	500 puntos
Piruletas buenas:	Suman 100 puntos
Piruletas malas:	Restan 100 puntos
Piruletas Quita Vidas:	Restan 1 vida

2.4 Movimiento y físicas:

Piruetto's Race se desarrolla en un plano 3D, sujeto a gravedad, donde el personaje puede desplazarse libremente por aquellas zonas habilitadas para ello. La cámara seguirá al personaje en todo momento.

El escenario presenta ciertos obstáculos con los que debe colisionar, que son el objetivo del juego, pero también presenta otros objetos del mundo real que no podrán ser atravesados.

Tipos colisiones:

Personaje- Piruletas (suman, restan o quitan vidas)

Personaje- objetos del escenario (no se produce ninguna acción, son objetos que no pueden ser atravesados).

2.5 Controles:

Movimiento: teclas W, A, S, D, y flechas del teclado.

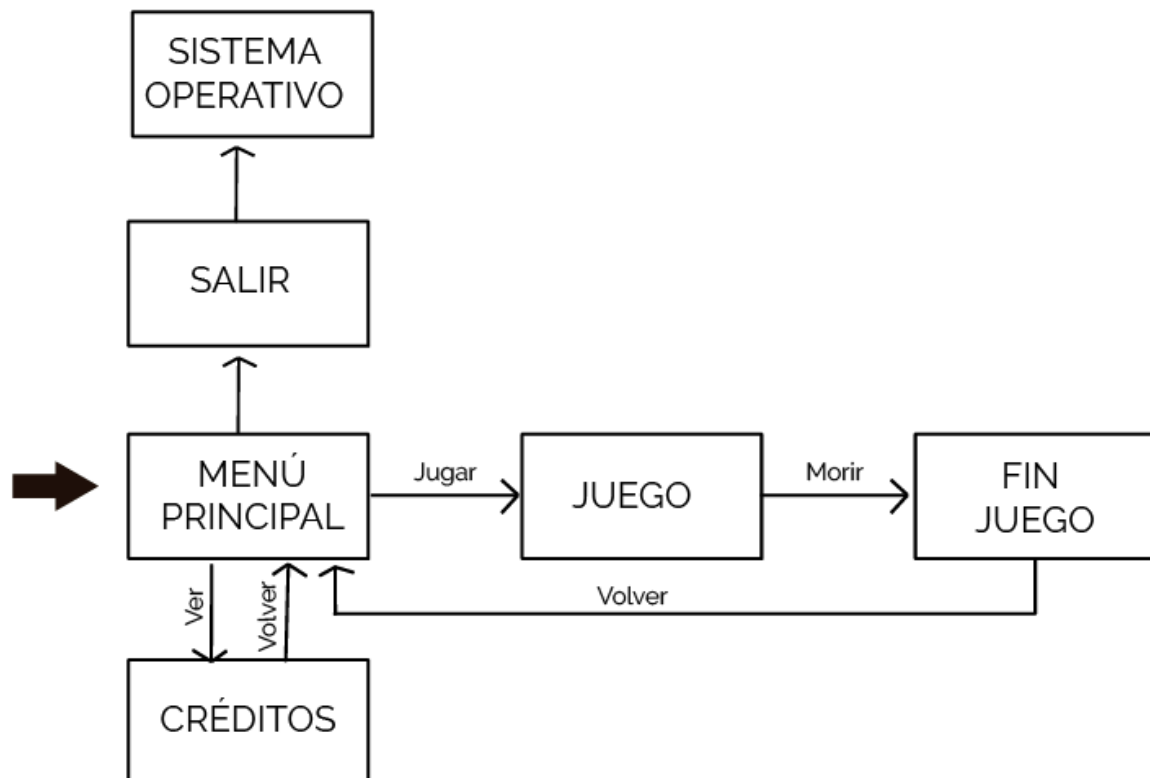
Ayuda: Barra Espaciadora.

3. Interfaz:

En esta sección se especificará con detalle cada una de las pantallas. No obstante dicha información está sujeta a cambios cuando así se considere necesario.

4.1 Diagrama Flujo:

En el siguiente diagrama se muestran las pantallas presentes de Piruetto's Race.

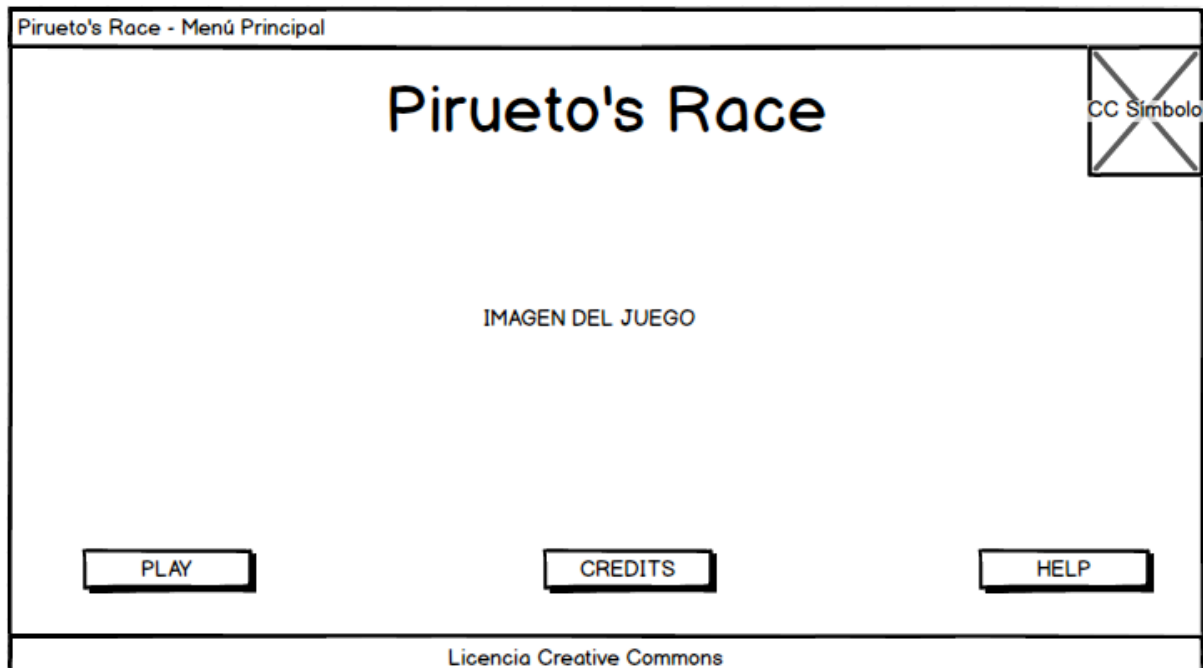


4.2 Diagrama Mecánicas de Juego:



4.3 Menú principal

A continuación el *wireframe* de la pantalla de Menú Principal:



Botón *Play*: al pulsarlo se inicia el juego

Botón *Credits*: al pulsarlo lleva a la pantalla *Créditos*.

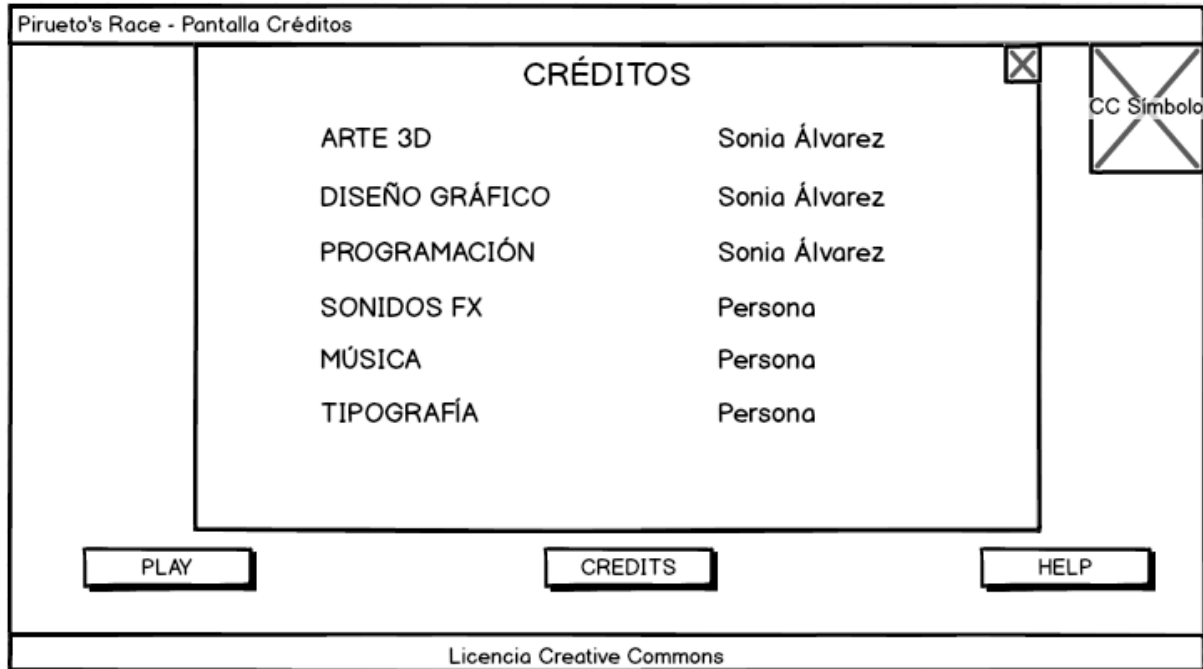
Botón *Help*: al pulsarlo lleva a la pantalla de Ayuda.

Botón *Exit*: al pulsarlos lleva de vuelta al Sistema Operativo (sólo en versión escritorio).

Fondo de pantalla: Imagen 3D Personaje y Juego: Vista 3D del Protagonista y del juego a modo de ambientación.

4.4 Créditos:

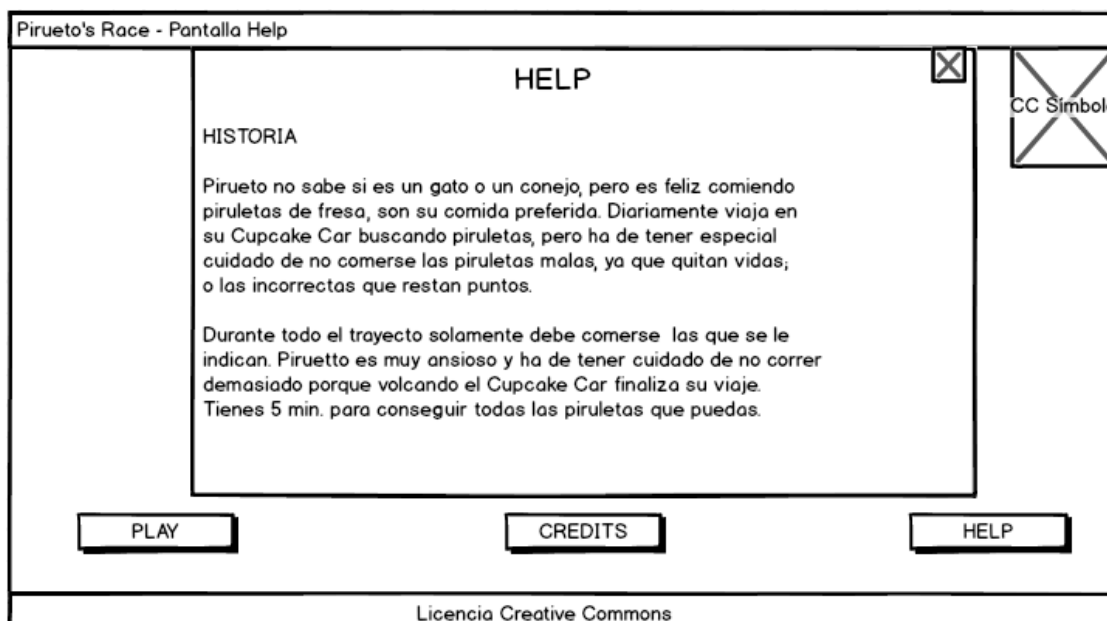
A continuación el boceto de la pantalla de Créditos:



Pantalla: texto con los componentes del equipo de desarrollo.

4.5 Help:

A continuación el boceto de la pantalla de Help:



4.6 Final Juego:

A continuación el boceto de la pantalla de Final de Juego:



Panel: texto con la notificación de fin de juego.

Botón Replay: al pulsarlo, se regresa al Menú Principal.

4. Arte

Piruetto's Race debe tener un carácter simpático, dulce y colorido, con modelos con formas muy básicas y música acorde al estilo del juego. A continuación se enumeran los recursos necesarios:

4.7 Arte 2D

Todas las imágenes se realizarán en formato .png (además del formato propio de creación: .psd para Adobe Photoshop, .ai para Adobe Illustrator) para poder realizar modificaciones futuras.

Interfaz

- Logo: Logo con el texto: *Piruetto's Race* estilo Cartoon, colorido.
- Diseño creativo de personaje 2D.
- Plantilla para la GUI: en principio no se utilizará ninguna plantilla sino que se desarrollará una personalizada para el juego.
- Cartel de Inicio: Imagen con el logo e imagen del videojuego que aparecerá al principio del juego.
- Cartel de Final de juego: Imagen con el texto *Game Over*.

Texturas

Todo los modelos 3D llevan su textura. Algunas texturas se elaborarán en Illustrator y se colocarán directamente dentro de Unity.

4.8 Arte 3D

Comprenderá la elaboración del personaje Piruetto, el coche, los objetos del escenario y el propio escenario en sí.

Todos los modelos 3D se guardaran en el formato de trabajo .max de Autodesk 3DS Max y se exportaran a Unity en formato .FBX.

4.9 Animación

Piruetto presentará unas sencillas animaciones para dar más realismo al personaje:

- Movimiento de cabeza.
- Movimiento de orejas.

4.10 Sonido

La música se convertirá a formato .ogg y los archivos de sonido estarán en formato .wav.

- Música general del Juego.
- Música de Introducción.

Sonidos FX:

- Suma puntos
- Resta puntos
- Choque
- Cuenta regresiva
- Quitar Vida
- Sonido motor.
- Encendido motor.
- Aplausos público.
- Game Over

Anexo 10. Glosario/Índice analítico

API: Interfaz de Programación de Aplicaciones

Art Toy: Juguetes coleccionables producidos en edición limitada.

Asset: Recursos que por defecto proporciona Unity y/o que se pueden crear para realizar un proyecto

C Sharp: Lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft.

Closed Caption: Sistema de subtítulos originalmente destinado a personas sordas, o con dificultades para captar la señal de audio, para que puedan ver de forma escrita lo que se emite oralmente. Este sistema incluye la música de fondo y efectos de sonido.

Colliders: Componente colisionador que define la forma de un objeto a efectos de colisiones físicas.

EULA: Acuerdo de Licencia de Software Unity

FBX: Formato de archivo 3D independiente y gratuito, que proporciona acceso al contenido de cualquier software.

Flat Design: Técnica de diseño plano y minimalista. Se puede ver en proyectos como Windows Vista.

Game Design Document: Documento básico donde se describen las directrices necesarias para el desarrollo de un videojuego.

Heightmap: Mapa de altura para la generación automática de terrenos realistas.

Indie: Videojuego independiente.

Kawaii: Adjetivo del idioma japonés que puede ser traducido al español como "lindo" o "tierno".

Mecanim: tecnología de animación de Unity para realizar el movimiento fluido y natural del personaje y otros elementos.

Monodevelop: Software de depuración y edición de Scripts.

OpenGL: especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D

Parenting: asignando un objeto hijo a un objeto padre

Placeholder: Objetos simples de trabajo que posteriormente se substituyen por los objetos finales.

Plugin: aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

Prefabs: instancias de un objeto que comparten las mismas propiedades,

Rigidbody: Componente de Unity que permite recibir fuerza y torque para hacer que sus objetos se muevan en una manera realista.

Anexo 11. Bibliografía

José Luis Rodríguez Illera. (2013). Aprendizaje y educación en la sociedad digital. Universidad de Barcelona. http://www.ub.edu/seasd/wp-content/uploads/2013/11/ApyEd-en-la-sociedad-digital_completo.pdf

DEV. Libro Blanco de Desarrollo de videojuegos presentado por la Asociación Española de Productoras y Desarrolladoras de Videojuegos y Software de Entretenimiento. [en línea] <http://www.dev.org.es/> [Fecha de consulta: 06/03/2016]

Gros, B. (2007). Digital games in education. The design of games-based learning environments. Journal of Research on Technology in Education.

Whitton, N. (2010). Learning with digital games: A practical guide to engaging students in higher education. New York, NY. Gee, J. P. (2005). Good video games and good learning. Phi Kappa Phi Forum.

Unity Tutorials. [en línea] <https://unity3d.com/es/learn/tutorials> [Fecha de consulta: 06/03/2016]

Unity Documentation. [en línea] <http://docs.unity3d.com/Manual/index.html>. [Fecha de consulta: 06/03/2016]

Unity Scripting API. [en línea] <http://docs.unity3d.com/ScriptReference/index.html>. [Fecha de consulta: 06/03/2016]

Unity Tutorial WheelCollider. [en línea] <http://docs.unity3d.com/Manual/WheelColliderTutorial.html>. [Fecha de consulta: 06/03/2016]

Unity Documentation. Time.deltaTime. [en línea] <http://docs.unity3d.com/ScriptReference/Time-deltaTime.html>[Fecha de consulta: 06/03/2016]

Unity Documentation. Update and FixedUpdate. [en línea] <https://unity3d.com/es/learn/tutorials/modules/beginner/scripting/update-and-fixedupdate>. [Fecha de consulta: 06/03/2016]

Unity Documentation. Prefabs [en línea] <http://docs.unity3d.com/es/current/Manual/Prefabs.html>. [Fecha de consulta: 06/03/2016]

Nicolás Arrija Landa Cosio. Guía Total del Programador. C#. Manuales Users. Fox Andina. Argentina.

Fersocam.(2011) Confederación Estatal de Personas Sordas. Dossier CNSE. [en línea] [Fecha de consulta:01/05/16] <http://www.fesocam.org/wp-content/descargas/2011/06/Dossier-CNSE-2011.pdf>

Informe de la situación de la Lengua de Signos Española Hoy. Actas del Congreso CNLSE sobre la investigación de la Lengua de Signos Española 2013. Centro de Normalización Lingüística de la Lengua de Signos Española. Ministerio Sanidad. Servicios Sociales e Igualdad. [en línea] [Fecha de consulta:01/05/16] [http://www.sis.net/documentos/documentacion/INFLenguaSignos\(online\).pdf](http://www.sis.net/documentos/documentacion/INFLenguaSignos(online).pdf)