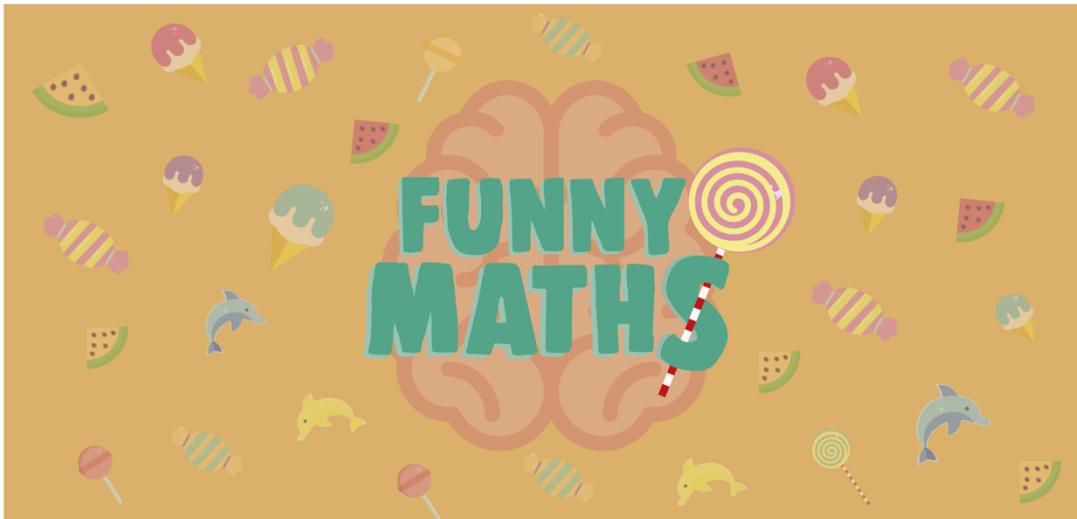




FACULTAT
**DE CIÈNCIES
I TECNOLOGIA**

UVIC | UVIC·UCC

Trabajo de Fin de Grado



Rocío Barcos Martínez

Grado en Multimedia

Tutor/a: Sergi Grau Carrión

Vic, junio de 2019

Resumen

Título: "Funny Maths"

Palabras clave: Minijuego 2D, Unity, Discalculia, Gamificación, Serious Game

Autor/a: Rocío Barcos Martínez

Tutor: Sergi Grau Carrión

Fecha: junio de 2019

Castellano

En este trabajo de final de grado, expondré el procedimiento, el funcionamiento y el resultado de la creación de una App de minijuegos infantiles (6 a 8 años), para mejorar la discalculia, una dificultad para aprender los principios del cálculo originada por un problema cerebral que dificulta el uso del sistema simbólico.

Así pues, el objetivo principal de este proyecto ha sido el diseño (con Adobe Illustrator) y la implementación (con Unity y C#) de una App de minijuegos en 2D. Se han implementado y diseñado dos de los minijuegos: "Consigue el Helado" y "Completa la chocolatina". Estos dos minijuegos son dos ejercicios *gamificados* que ayudan a mejorar esta dificultad, adaptándolos estéticamente al público objetivo, los niños.

El resultado del proyecto ha sido un primer demostrador funcional de una App de minijuegos, con posibilidad de añadir en un futuro más minijuegos adaptados siguiendo la historia y temática de éste, para la mejora de esta dificultad que hoy en día cada vez es más diagnosticada.

Inglés

In this final project, I will explain the procedure, the performance and the result obtained after the development of a children's mini-games App (6 to 8 years), to improve the dyscalculia, a difficulty to learn the principles of calculus caused by a brain disorder that interferes with the use of the symbolic system.

Therefore, the main objective of this project has been the design (with Illustrator) and the implementation (with Unity and C#) of a 2D mini-game App. Two mini-games have been implemented and designed: "Get the Ice Cream" and "Complete the Chocolate". These two mini-games are gamified exercises that help to improve that disorder, adapting them aesthetically to the target audience: children.

The result of the project was a first functional demo of a mini-game App, with the possibility of adding more adapted games in the future, following the history of the game, for the improvement of this difficulty that nowadays is increasingly diagnosed.

Índice

RESUMEN	1
INTRODUCCIÓN	3
CONTEXTUALIZACIÓN	4
¿QUÉ SON LOS <i>SERIOUS GAMES</i> ?	4
¿QUÉ ES LA GAMIFICACIÓN?	4
¿QUÉ ES LA DISCALCULIA?	4
EJERCICIOS SELECCIONADOS	6
APLICACIONES Y SOFTWARES ACTUALES	6
OBJETIVOS	10
GENERALES.....	10
ESPECÍFICOS.....	10
REQUERIMIENTOS	11
EL JUEGO	12
<i>MOCKUPS</i> Y MECÁNICAS.....	12
PROCESO DE CREACIÓN	18
PLANIFICACIÓN.....	18
TECNOLOGÍAS UTILIZADAS	19
DISEÑO GRÁFICO	21
IMPLEMENTACIÓN.....	27
ESTUDIO ECONÓMICO	36
GASTOS DE DESARROLLO:	36
GASTOS FIJOS:.....	36
CONCLUSIONES	37
FUTURAS MEJORAS.....	37
WEB GRAFÍA	39

Introducción

La App de minijuegos “Funny Maths” se enmarca en un proyecto de investigación del grupo de investigación Digital Care de la UVic-UCC que trabaja con la reeducación de las diferentes dificultades de aprendizaje. En este caso en concreto el objetivo es desarrollar una aplicación infantil con una serie de minijuegos para la reeducación de la discalculia.

Como estudiante de Multimedia, mi reto era hacer un trabajo de final de grado implementando las dos mayores disciplinas que he ido trabajando y aprendiendo durante estos últimos cuatro años. Asignaturas como programación avanzada, *game design*, entornos virtuales, arquitectura de la información (usabilidad) o diseño de interfaces, han ayudado a que haya decidido hacer como proyecto final diseñar y desarrollar una Aplicación de minijuegos en 2D, para poner en práctica todos los conocimientos adquiridos durante la carrera.

En esta memoria explicaré todo el proceso que he seguido para crear “Funny Maths”, una App de minijuegos infantiles de 6 a 8 años, así como el funcionamiento y el resultado final.

Contextualización

¿Qué son los *Serious Games*?

Los *Serious Games* [1] es un método de enseñanza que se le conoce como *game-based learning*, una tendencia que cada vez esta más presente en las aulas de educación primaria, en universidades o empresas. Consiste en convertir la experiencia del aprendizaje en una actividad que combina enseñanza y entretenimiento, mediante videojuegos y aplicaciones con un trasfondo didáctico.

En estos últimos años, los *Serious Games* han demostrado que es posible aprender jugando, ya que el objetivo principal no es la diversión o el entretenimiento, sino el aprendizaje o la práctica de habilidades, empleando nuevas formas didácticas basadas en la *gamificación*.

¿Qué es la gamificación?

La gamificación [2] es una técnica de aprendizaje capaz de motivar y enseñar a alumnos de una forma lúdica. Esta técnica traslada la mecánica de los juegos al ámbito educativo y profesional con el fin de conseguir mejores resultados. El objetivo principal es potenciar al máximo las habilidades de los niños a través de la experimentación y el juego, porque mediante la experiencia y la diversión es más fácil aprender.

Las actividades realizadas en este contexto buscan alcanzar tres objetivos principales: por una parte, la fidelización del alumno o usuario, por otro lado, una herramienta contra el aburrimiento y a favor de la motivación, y por último recompensar mediante premios, aunque realmente no hay incentivo más allá del aprendizaje.

¿Qué es la discalculia?

La discalculia [3][4] del desarrollo es un trastorno caracterizado por las dificultades en la correcta adquisición de las habilidades en las matemáticas, que afectan de una manera significativa al rendimiento académico o en las actividades de la vida cotidiana que requieren capacidad para el cálculo. Cuando los niños que padecen esta dificultad están en la escuela, queda reflejado en repetidos suspensos en el área de las matemáticas.

El porcentaje de la población con discalculia del desarrollo se estima entre el 3 y el 6 por ciento, y la probabilidad de que afecte tanto a niñas como a niños es la misma.

El diagnóstico debe realizarse cuando los niños finalizan primero de educación primaria (6-7 años). El periodo crítico para el aprendizaje numérico se sitúa entre los 4 y 7 años. Pasado este tiempo, si hay dificultades, hay que detectarlas y reeducarlas.

Está demostrado que es una dificultad que se puede reeducar mediante ejercicios personalizados según los déficits. Esta reeducación tiene que ser intensa, con un mínimo de 4 sesiones semanales de 10 minutos cada una [3][4].

A continuación, se detallan una serie de ejercicios en los que niños con este trastorno muestran dificultad:

- *Subitizing*, consiste en captar de inmediato cuantos elementos se encuentran dentro de una escena visual.
- Identificación numérica.
- Transcodificación, tanto de formato escrito a formato verbal, o viceversa, por ejemplo, capacidad de asociar el número “4” con la palabra “cuatro.
- Completar series, capacidad para completar secuencias numéricas, por ejemplo, en la serie “1 _ 3” falta el número 2.
- Memoria de trabajo numérica
- Línea numérica mental, es decir, la capacidad que tienen los niños para ubicar números en una línea imaginaria entre dos límites, por ejemplo, de 0 a 10, de 0 a 20, de 0 a 100.
- Sistema base 10 (composición y descomposición de números en sistema base 10)
- Cálculo mental

Ejercicios seleccionados

Para crear la aplicación de minijuegos “Funny Maths” he seleccionado dos de los ejercicios mencionados anteriormente para después gamificarlos y convertirlos en un juego atractivo y entretenido para los niños.

Estos ejercicios han sido la línea numérica mental, y completar series. La historia de esta aplicación consiste en que una vez se prueben estos dos minijuegos y veamos que es atractivo para los niños, pero además les ayude, poder crear todos los demás.

Para poder pilotar una primera prueba con niños, este proyecto de centra sólo en dos de los ejercicios.

Aplicaciones y softwares actuales

Durante estos últimos años se han desarrollado distintas aplicaciones o programas online que permiten trabajar los conceptos matemáticos en un formato totalmente digital, y la ventaja que tienen es que pueden adaptarse a las necesidades de cada niño, además de ser un medio de aprendizaje atractivo y entretenido.

1. *Dinamo Números*



Fig. 1: Logo de *Dynamo Números*

- **Objetivos:** Dar apoyo o identificar a estudiantes en riesgo de desarrollar discalculia y estudiantes que obtienen resultados en matemáticas por debajo de sus compañeros. Además de identificar la discalculia, también detecta el desarrollo tardío de las matemáticas.

- **Características:** Utiliza un enfoque graduado utilizando cuatro etapas: evaluar, planear, intervenir y revisar. Por lo tanto, guarda información de los resultados de cada niño para después intervenir de una manera personalizada.
- **Edad:** Niños de 6 a 9 años.
- **Tipo de software:** Programa de Ordenador
- **¿Gratuito o de pago?:** De pago, y el precio varía según si la compra es para el hogar o para la escuela.
- **¿Específico para la discalculia?:** Si
- **Web:** <https://dinamonumeros.com>

2. *The Number Race*

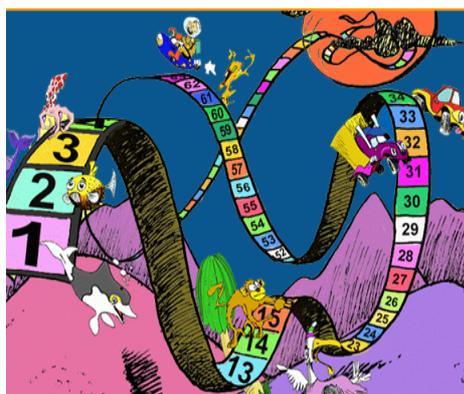


Fig. 2: Logo de The Number Raze

- **Objetivos:** fortalecer los mecanismos cerebrales de procesamiento de números, establecer la recta numérica mental, practicar contando, sumas y restas, fomentar la fluidez, y ayudar a los niños con discalculia.
- **Características:** Es un juego que se ha probado científicamente, aunque se necesitan más herramientas a parte de este software para conseguir mejorar.
- **Edad:** Niños de 4 a 8 años.
- **Tipo de software:** Programa de Ordenador para Windows Mac.

- **¿Gratuito o de pago?:** Gratuito.
- **¿Específico para la discalculia?** Si.
- **Web:** <http://www.thenumberrace.com/nr/home.php>

3. *The Number Catcher*

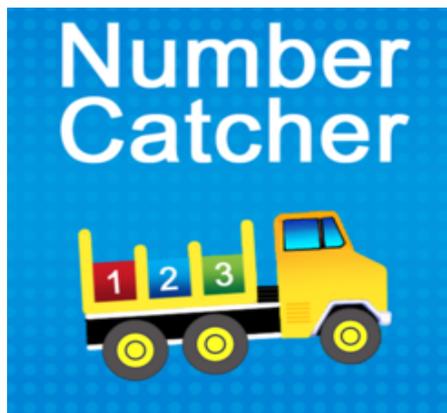


Fig. 3: Logo de The Number Catcher

- **Objetivos:** Sus principales objetivos son la enseñanza paso a paso de sumas y restas, fortalece los mecanismos cerebrales de procesamiento de números, fomentar la fluidez, centrarse en números de varios dígitos, y ayudar a niños con discalculia.
- **Características:** Muy similar a “The Number Race”, diseñado para que los niños que dan sus primeros pasos con números aprendan los conceptos básicos de número y aritmética, y para que los mas mayores desarrollen más fluidez con las matemáticas.
- **Edad:** Niños de 5 a 10 años.
- **Tipo de software:** Juego online
- **¿Gratuito o de pago?** Gratuito
- **¿Específico para la discalculia?** Si.
- **Web:** <http://www.thenumbercatcher.com/nc/home.php>

4. Math Playground



Fig. 4: Logo de Math Playground

- **Objetivos:** Practicar de una forma divertida una amplia variedad y temas de matemáticas.
- **Características:** es una página que recoge una amplia variedad de temas matemáticos, como por ejemplo aritmética, geometría, porcentajes, problemas de palabras, álgebra, gráficas e incluso lógica. Incluso videos explicando materia. Es una forma de que los niños exploren conceptos matemáticos en inglés.
- **Edad:** Niños de 8 a 12 años.
- **Tipo de software:** Juegos Online.
- **¿Gratuito o de pago?** Gratuito.
- **¿Específico para la discalculia?** No.
- **Web:** <https://www.mathplayground.com>

Objetivos

A continuación, se detallarán los objetivos generales de la aplicación, y los más específicos, a los que he tenido que ceñirme para que el proyecto salga adelante.

Generales

El objetivo principal del proyecto ha sido poder diseñar e implementar un demostrador que permita probar si el uso de los minijuegos puede ayudar a la reeducación y a la vez mejorar la adherencia del tratamiento. Por ese motivo se han creado dos minijuegos funcionales basados en los ejercicios mencionados anteriormente.

Tradicionalmente, la reeducación de la discalculia se ha hecho utilizando actividades clásicas a papel y lápiz. Así que uno de los objetivos también es poder demostrar que un videojuego es una manera de atraer más a los niños y que puedan mejorar de forma entretenida y amena, sin ser un simple ejercicio de matemáticas convencional.

Por último, otro de los objetivos ha sido poder convertir esta aplicación en un *Serious Game*, como he explicado antes, combinar la enseñanza con el entretenimiento.

Específicos

Desarrollar un entorno donde los niños puedan crear su propio avatar, con su nombre y gustos personales (elegir color de pelo, de piel, de ropa...), y en el que puedan acceder a diferentes actividades (los dos ejercicios *gamificados*), y poder ir subiendo de dificultad a medida que van superando los niveles.

Otro de los objetivos principales, es crear una aplicación que fuera atractiva visualmente para los niños. La temática son las golosinas, ya que generalmente es un premio para los más pequeños, y les suele gustar tanto a niñas como a niños.

Requerimientos

Los requerimientos de la aplicación están definidos, porque se enmarca en un proyecto de investigación del grupo de investigación Digital Care de la UVic-UCC. Éstos han sido:

- Que “Funny Maths” esté implementado con el motor de juegos Unity.
- Que todas las imágenes utilizadas sean de creación propia, creadas con Adobe Illustrator.
- Por otra parte, que fuera una aplicación para dispositivos móviles y para web.
- Que los minijuegos fueran 2D.

El juego

Mockups y mecánicas

El primer paso para crear esta aplicación fue pensar cómo convertir en un juego los dos ejercicios en los que me centré: colocar números en una línea imaginaria, y encontrar el número que falta en una serie.

El segundo paso fue buscar una temática atractiva que pudiera gustar tanto a niños como a niñas. Normalmente, las golosinas siempre han sido un premio para los niños, por eso toda la aplicación esta ambientada en esta dirección.

Para el primer juego pensé en colocar un helado sobre una barra de fuego, si el helado está bien colocado lo ganará puntos, si no el helado se derretirá y no ganará nada.

Para el segundo, pensé en una tableta de chocolate, donde cada onza es un número y entre las tres onzas componen una serie, y consiste en colocar la onza con el número que falta.

Una vez definidos estos dos minijuegos, el segundo paso fue crear una serie de *mockups* definiendo las mecánicas que desempeñarían cada una de las pantallas (ver *Fig. 5*).

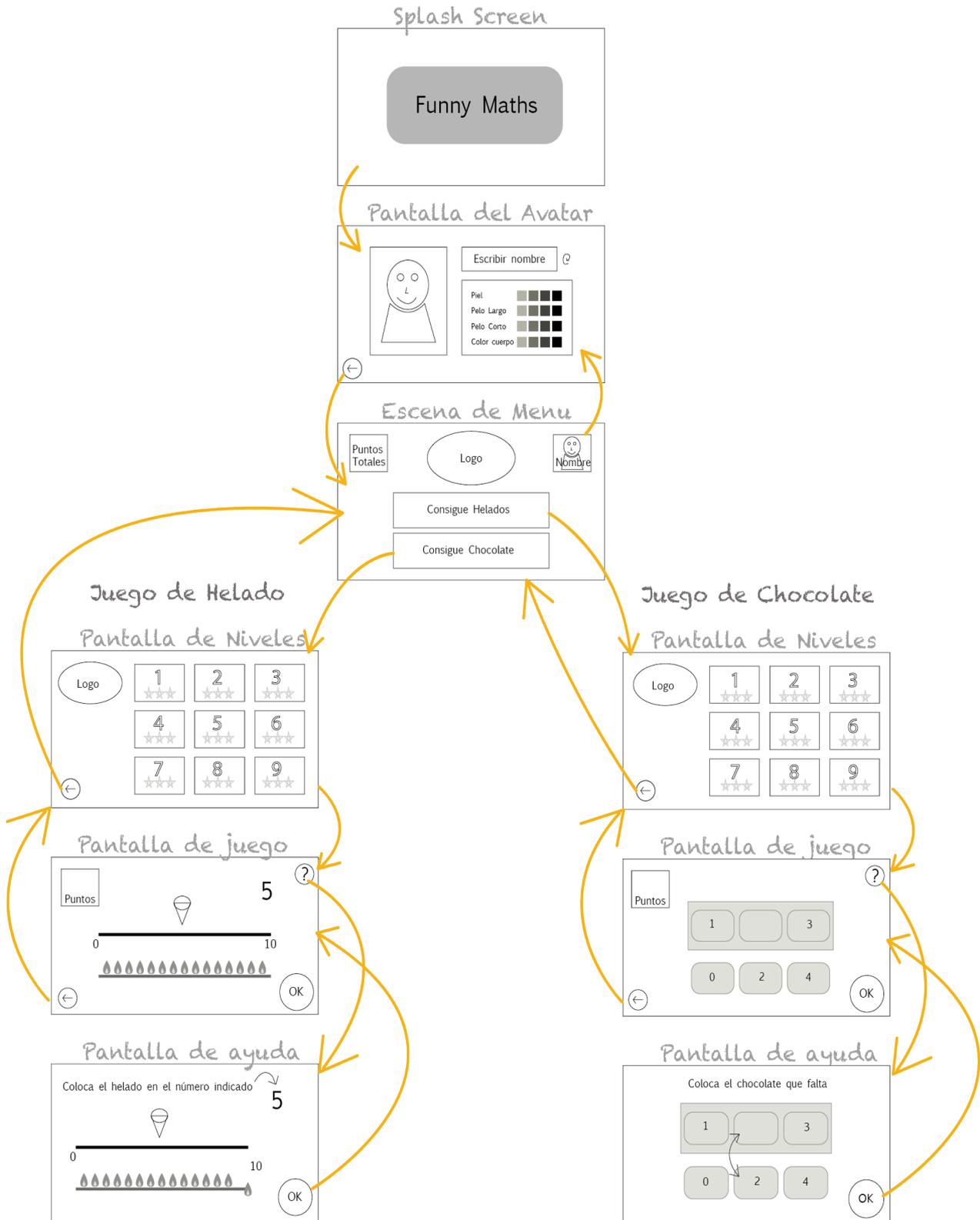


Fig. 5: Mockups de la aplicación Funny Maths.

1. Splash Screen

La "Splash Screen" es la pantalla inicial donde aparecerá el logo de la aplicación. Esta pantalla será estática y durará 5 segundos para seguidamente mostrar la pantalla de Avatar, o la pantalla de menú, según si es la primera vez que se entra al juego y no hay datos guardados, o ya se ha jugado anteriormente y sí que los hay.

2. Pantalla de Avatar

En esta pantalla se creará el Avatar del usuario. En primer lugar, el usuario escribirá su nombre y luego podrá elegir entre color de piel, color de pelo, pelo largo o corto y color de ropa. Una vez pulsa el botón de siguiente que se encuentra en la esquina inferior izquierda de la pantalla se redirigirá a la pantalla de menú.

El botón situado al lado del formulario para escribir el nombre tiene la función de deshacer, es decir, volver a dejar los datos del cuerpo y nombre que había anteriormente.

3. Pantalla de Menú

En la pantalla de menú aparecen los dos minijuegos disponibles. En la esquina superior derecha aparece el nombre y el avatar seleccionado en la pantalla anterior por el usuario, además también tiene la función de botón, que te redirige otra vez a la pantalla de Avatar por si el jugador quiere editar sus datos. En la esquina superior izquierda aparecen los puntos totales, es decir, la suma de los helados y chocolates que se han ido consiguiendo durante las partidas. En el centro de la pantalla irá ubicado el logo.

4. Pantalla de Niveles (juego del helado y juego del chocolate)

En esta pantalla se mostrarán los diferentes niveles del minijuego del escogido. En la esquina inferior izquierda habrá un botón de salida, que llevará de nuevo a la pantalla de menú. Además, el logo también estará presente en esta pantalla.

Sólo se podrá pasar de nivel si se consiguen 3 puntos como mínimo del nivel anterior. Para obtener estrellas:

- Dos o tres puntos equivalen a una estrella.
- Más de tres puntos y menos de nueve equivalen a dos estrellas.

- Los nueve puntos (todos) equivalen a tres estrellas.

5. Pantalla de juego

a. Juego del helado (ejercicio línea numérica)

La pantalla del juego del Helado consiste en colocar el helado en la posición correcta (número superior derecha) en una barra que tiene un valor inicial y otro final (por ejemplo, en el nivel 1 entre el valor 0 y el valor 10). El botón de OK servirá para que el usuario verifique que ha puesto el helado en la posición correcta, por si se equivoca poder cambiarlo de posición. Si no se coloca el helado en una posición cercana al número indicado, se encenderán los fuegos y el helado se derretirá, en caso de que lo coloque en la posición correcta se encenderán los fuegos, excepto el fuego situado en la posición correcta del helado.

En esta pantalla aparecen en la esquina superior izquierda un panel con los helados ganados, en la esquina superior derecha un botón de ayuda, que redirige a la pantalla de ayuda del juego, y finalmente en la esquina inferior izquierda un botón de salida, que lleva a la pantalla de los niveles.

Diferentes niveles del juego del helado:

En los dos primeros niveles, en caso de fallar, se encenderán todos los fuegos menos el que iría en la posición correcta, para poder dar alguna pista y saber donde se ha equivocado.

- Nivel 1: La barra irá de 0 a 10 y el numero aleatorio se podrá repetir.
- Nivel 2: La barra irá de 0 a 10 y el numero aleatorio no se repite.

A partir del nivel 3, en caso de fallar, sólo se encenderá el fuego de la posición en la que ha fallado, para aumentar el nivel de dificultad.

- Nivel 3: La barra irá de 0 a 20 y el numero aleatorio se podrá repetir.
- Nivel 4: La barra irá de 0 a 20 y el numero aleatorio no se podrá repetir.

- Nivel 5: La barra irá de 0 a 30 y el numero aleatorio se podrá repetir.
- Nivel 6: La barra irá de 0 a 30 y el numero aleatorio no se podrá repetir.
- Nivel 7: La barra irá de 0 a 50 y el numero aleatorio se podrá repetir.
- Nivel 8: La barra irá de 0 a 50 y el numero aleatorio se podrá repetir.
- Nivel 9: La barra irá de 0 a 200 y es difícil que se repita el número aleatorio, pero podría pasar.

b. Juego del chocolate (ejercicio completar series)

La pantalla del juego del Chocolate consiste en colocar la onza con el número que falta en la tableta de chocolate que hay justo encima. Esta tableta se compone de una serie de números en la que va incrementando la dificultad a medida que se superan los niveles. El botón de OK servirá para que el usuario verifique que ha puesto la onza en una posición correcta.

En esta pantalla, siguiendo el mismo estilo que la del helado, en la esquina superior izquierda estará el contador de los chocolates ganados. En la esquina superior derecha hay un botón de ayuda que redirige a la pantalla de ayuda. Finalmente, en la esquina inferior izquierda hay un botón para salir e ir a la pantalla de niveles.

Diferentes niveles del juego del chocolate:

- Nivel 1: Series de distancia 1 del 0 al 10.
- Nivel 2: Series de distancia 1 del 0 al 20.
- Nivel 3: Series de distancia 1 del 20 al 50.
- Nivel 4: Series de distancia 2 del 0 al 20 (pares).
- Nivel 5: Series de distancia 2 del 0 al 20 (impares).
- Nivel 6: Series de distancia 2 del 20 al 50 (pares).

- Nivel 7: Series de distancia 3 del 20 al 50.
- Nivel 8: Series de distancia 3 del 20 al 50.
- Nivel 9: Series de distancia 5 del 20 al 50 (Sin ser 5,10,15,20..., sino empezando por cualquier numero, por ejemplo 22,27,32...)

6. Pantalla de ayuda (juego helado y juego chocolate)

En esta pantalla se muestra una pequeña demostración con animaciones de cómo se juega, con un botón de OK que te vuelve a dirigir a la pantalla de juego.

Proceso de creación

A continuación, se explicarán las diferentes fases por las que ha pasado el proyecto, desde la implementación hasta el resultado final.

Planificación

Durante el proyecto he ido asumiendo diferentes roles:

Rol *Game Designer*

Durante el mes de enero he asumido el rol de *Game Designer*, es decir, en este proceso he pensado la historia, he hecho los *mockups* de la aplicación y ingeniado las mecánicas y normas del juego.

Rol Diseñadora

En el mes de febrero diseñé toda la aplicación con Illustrator, pero he considerado marcar también los meses de marzo y abril porque durante el proceso de desarrollo fueron surgiendo más ideas o detalles que he ido añadiendo a la aplicación.

Rol Desarrolladora

El proceso de desarrollo del código es el que me ha llevado más tiempo, ya que para mí tenía un nivel más alto de dificultad. Empecé creando escenas en marzo y ultimando los detalles a mediados de mayo.

Rol Documentación

Durante el mes de abril empecé a estructurar la memoria, pero ha sido durante las últimas 3 semanas antes de la entrega cuando he desarrollado la mayoría de ella. Durante este rol también he editado una video demostración de “Funny Maths”.

	Enero	Febrero	Marzo	Abril	Mayo	Horas
Rol <i>Game Designer</i>	x					15
Rol Diseñadora		x	x	x		50
Rol Desarrolladora			x	x	x	160
Rol Documentación				x	x	50

Tecnologías Utilizadas

Para poder desarrollar esta aplicación se hizo uso de un conjunto de tecnologías:

1. Adobe Illustrator



Figura 6: Logo de Adobe Illustrator

Adobe Illustrator es una herramienta de dibujo que he utilizado para dibujar toda aplicación (diseño de todas las pantallas, objetos, *sprites* animados, logo...) además de los mockups presentes en el anterior apartado. Elegí esta herramienta de dibujo ya que la he utilizado durante años y me pareció una manera fácil y eficaz de dibujar a mi gusto toda la interfaz de la aplicación.

2. Color Adobe



Figura 7: Logo de Color Adobe

Para elegir un abanico de colores que se complementasen entre si utilicé un servicio online llamado Adobe Color. Esta extensión de Photoshop ayuda a combinar los colores de forma armoniosa y atractiva en proyectos de diseño.

3. Unity



Figura 8: Logo de Unity

Unity es un motor y un entorno de videojuego multiplataforma que he utilizado para desarrollar el proyecto. Este entorno me ha permitido crear todas las escenas de la aplicación, animar sprites, asignar scripts en algunos objetos para controlar su comportamiento y añadir diferentes sonidos en el juego.

Además, Unity dispone de una extensa oferta de elementos creados por otros usuarios dentro de la “Asset Store” [5], que me ha permitido descargar los sonidos de la aplicación. Hay que decir que todos los elementos del proyecto (menos el sonido) son de creación propia.

4. Visual Studio



Figura 9: Logo de Visual Studio

Visual Studio es un entorno de desarrollo integrado (IDE) compatible con múltiples lenguajes de programación, con el que he desarrollado el proyecto. Actualmente es el programa externo que se utiliza para desarrollar los scripts que componen proyectos de Unity.

5. Final Cut Pro



Figura 10: Logo Final Cut Pro

Finalmente, para la edición del video de demostración he utilizado Final Cut Pro, un software para la edición de video profesional no lineal.

Diseño gráfico

En este apartado se expone el resultado final de la aplicación, comparando los *mockups* con el diseño final, para ver la evolución de la aplicación.

Splash Screen

El logo de la aplicación representa que es un juego mental mezclando entretenimiento ambientado en golosinas. Esta pantalla no tiene botones y dura 5 segundos.

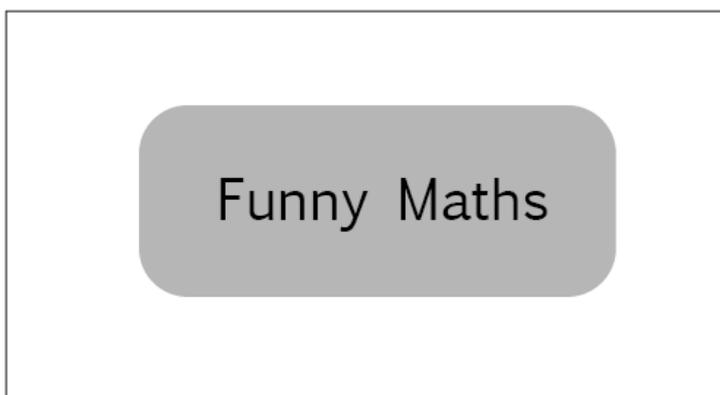


Fig. 11: Mockup Splash Screen.

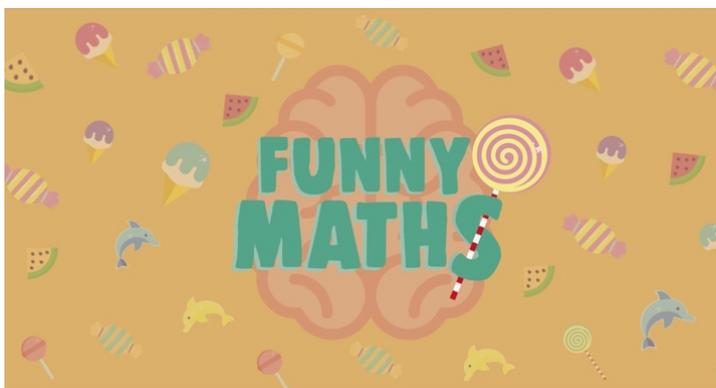


Fig. 12: Pantalla final Splash Screen

Pantalla Avatar

En esta pantalla el usuario escoge su avatar. Se pueden hacer diferentes combinaciones con los tres tipos de pelo corto o largo (moreno, rubio y pelirrojo), los tres tipos de color de piel (clara, morena y oscura) y los tres tipos de color de ropa (rosa, lila y azul). A continuación (Fig. 14) se verán las diferentes combinaciones.

Esta pantalla también contiene una celda que se puede rellenar con el nombre del usuario.

Hay un pequeño cambio en el botón del *mockup* (Fig. 13), al final no es una flecha, sino un OK.

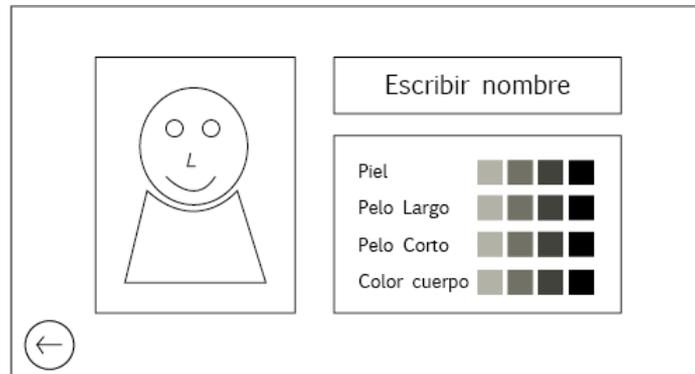


Figura 13: Mockup Avatar

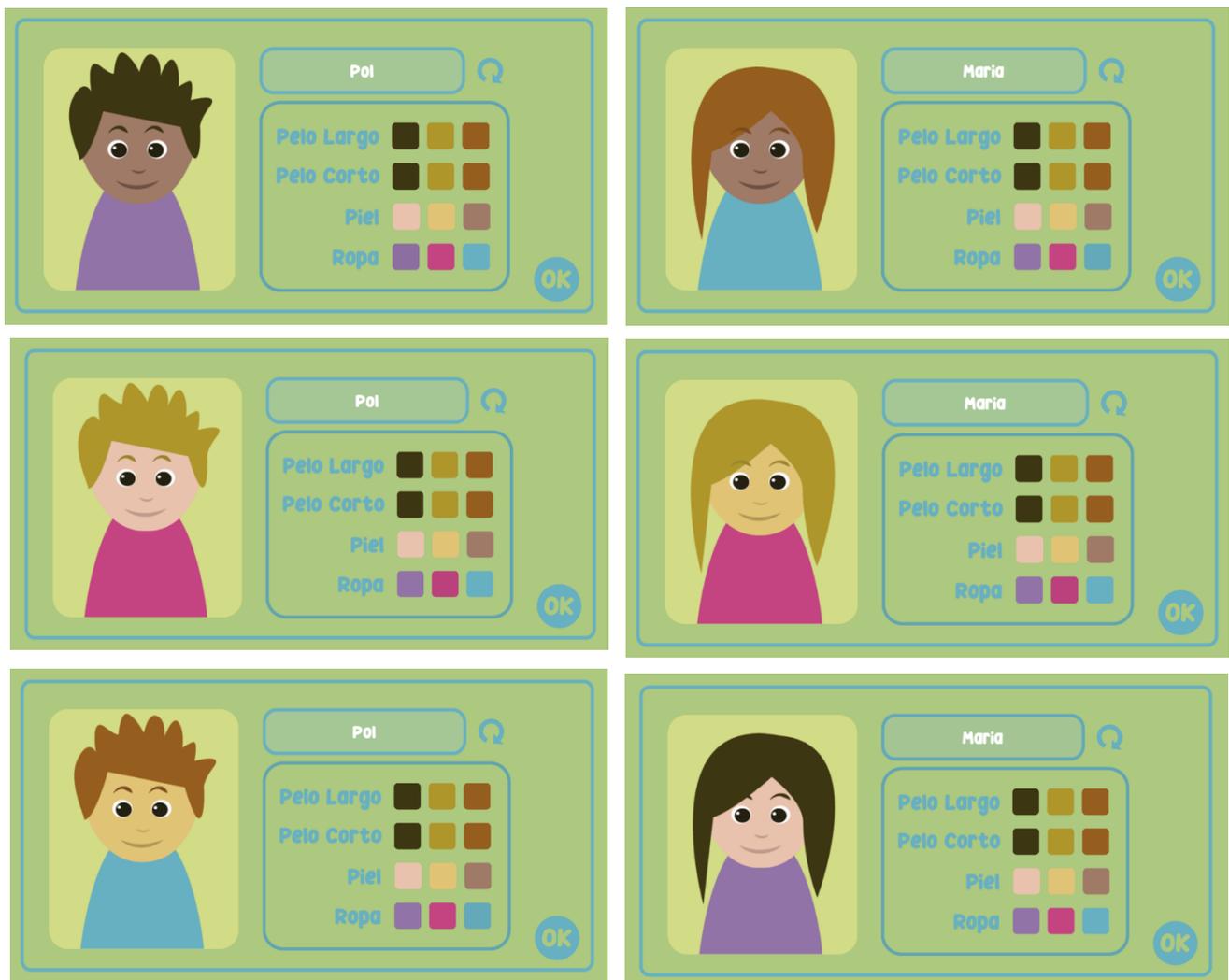


Figura 14: Diferentes combinaciones de la pantalla final de Avatar.

Pantalla de Menú

La pantalla de menú contiene tres botones. En la parte superior derecha de la pantalla está el botón que te lleva a la pantalla de Avatar para editar los datos del usuario. Los otros dos botones son los del juego del helado y chocolate, que te dirigen a la pantalla de los niveles de cada uno.

En la parte superior izquierda hay marcador con los puntos acumulados en el juego, es el único elemento de la pantalla que no es un botón.

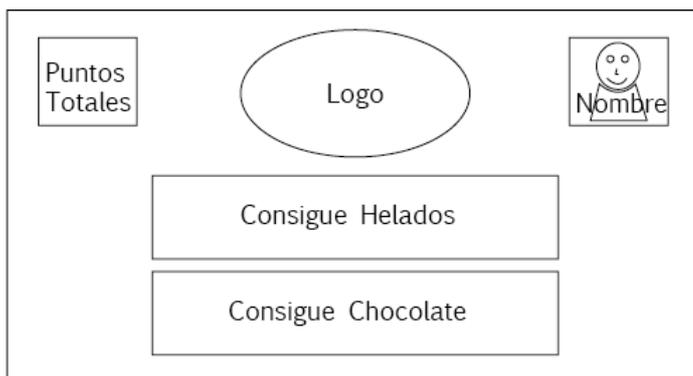


Figura 15: Mockup Menú



Figura 16: Pantalla final Menú

Pantalla de niveles del juego del helado y del chocolate

La única diferencia entre estas dos pantallas es la imagen del fondo de cada nivel, varios helados o varios chocolates respectivamente, con la opacidad muy baja para resaltar las estrellas y el número del nivel.

Cada uno de los niveles es un botón. Este botón será interactuable si el nivel anterior está superado. Los niveles desbloqueados mostrarán el número del nivel, las estrellas conseguidas en color naranja, y las restantes de color gris.

oscuro. En los niveles bloqueados se mostrarán las estrellas de un color gris más claro y en vez de el número del nivel se mostrará un candado.

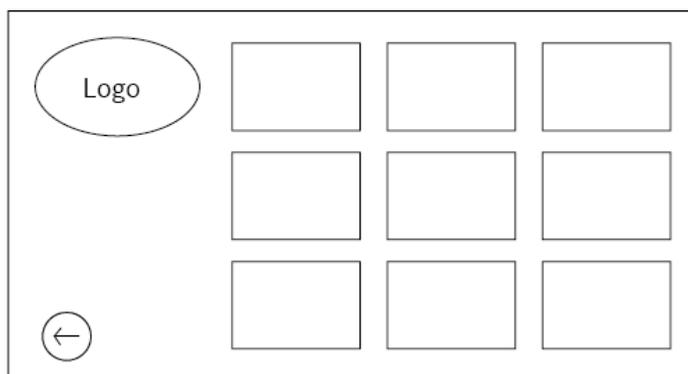


Figura 17: Mockup Niveles (juego helado y juego chocolate)



Figura 18: Pantalla de Niveles del juego del helado



Figura 19: Pantalla de Niveles del juego del chocolate

Pantalla del juego del helado

En esta pantalla de juego el helado cambia de posición clicando la barra. Hay tres botones: salir (que te vuelve a llevar a la pantalla de los niveles), ayuda (que te dirige a una pantalla de ayuda que se muestra mas adelante (Fig. 23)) y el OK, que sirve para verificar/comprobar si el helado esta en la posición correcta.

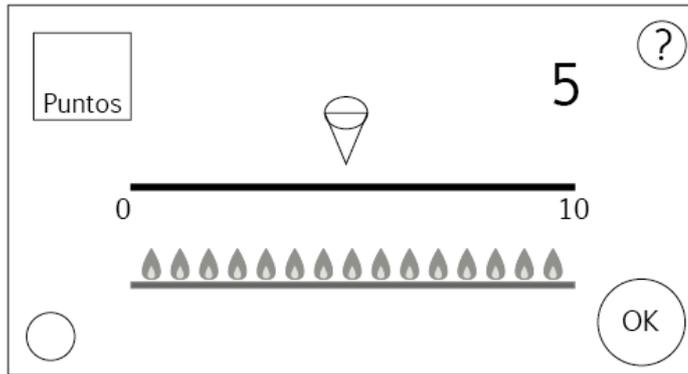


Fig. 20: Mockup del juego del helado

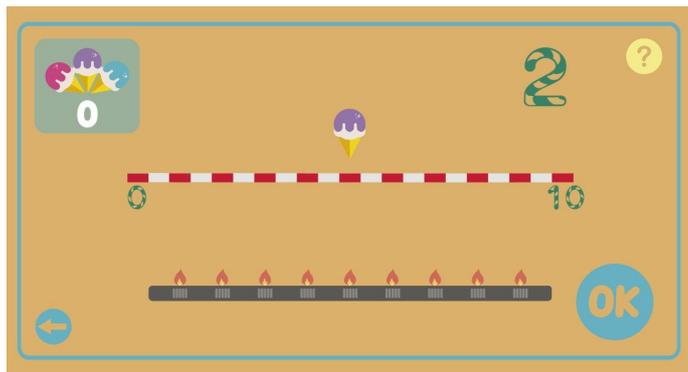


Fig. 21: Pantalla del juego del helado

Pantalla de ayuda (juego del helado)

En esta pantalla hay una animación que se va repitiendo. Primero el helado se coloca en la posición correcta y hace una animación que simula que ha ganado y seguidamente se coloca en la posición incorrecta y hace una animación simulando que ha perdido (el helado se deshace). Si se pulsa el botón de OK se vuelve a la pantalla de juego.

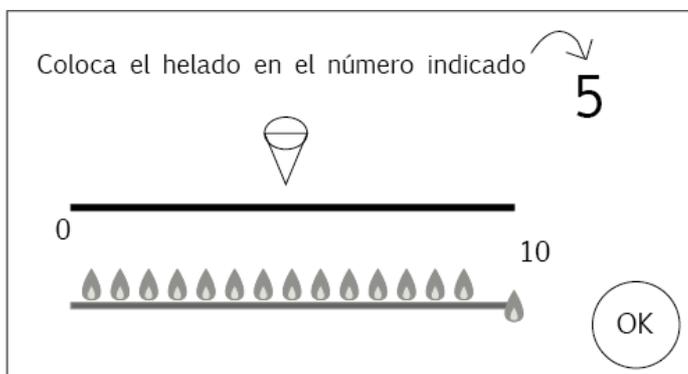


Fig. 22: Mockup ayuda helado



Fig. 23: Pantalla ayuda helado

Pantalla del juego del chocolate

En el juego del chocolate hay un *drag and drop*, un sistema para poder arrastrar las onzas de chocolate que hay sueltas en la parte inferior hacia la tableta. Si una de las onzas colocada en la tableta se quiere quitar y poner otra, hay que volver a arrastrarla hacia su posición inicial. En esta pantalla hay tres botones: salir (que te vuelve a llevar a la pantalla de los niveles), ayuda (que te dirige a una pantalla de ayuda que se muestra mas adelante (Fig. 27)) y el OK, que sirve para verificar/comprobar si la onza de chocolate esta en la posición correcta.

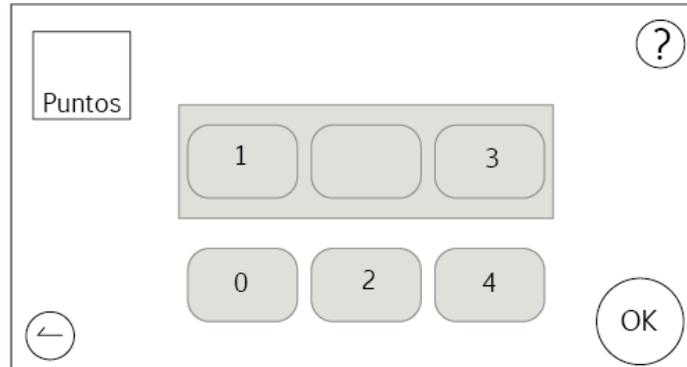


Fig. 24: Mockup juego chocolate

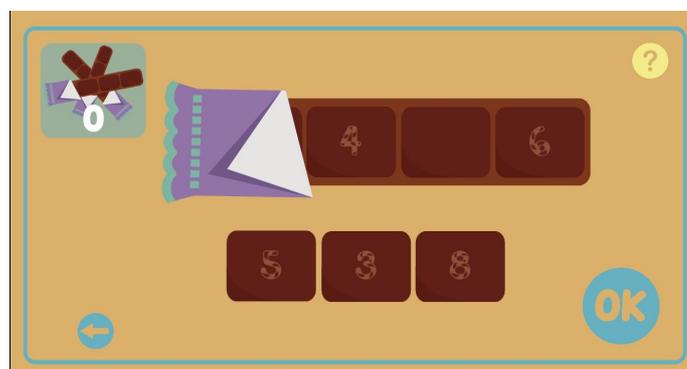


Fig. 25: Pantalla juego chocolate

Pantalla de ayuda (juego del chocolate)

Esta pantalla contiene una animación que muestra cómo tendría que posicionarse la onza de chocolate en la tableta para hacerlo correctamente. Si pulsas el botón de OK vuelves a la pantalla de juego.

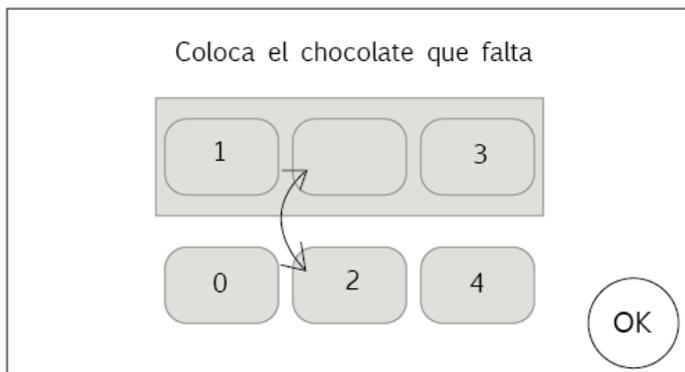


Fig. 26: Mockup ayuda chocolate

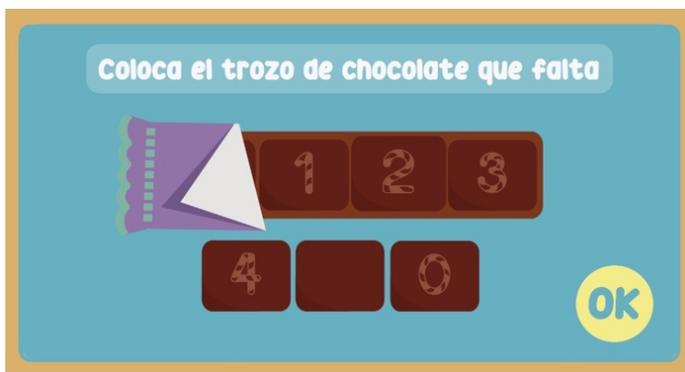


Fig. 27: Pantalla ayuda chocolate

Implementación

Para la creación del código de la aplicación he utilizado la documentación de Unity [6], la cual tiene un manual muy completo de cómo utilizar todos los componentes por editor y por código.

A continuación, desglosaré todos los pasos seguidos para el desarrollo en el entorno de Unity y las partes de código más importantes:

- 1) Crear las diferentes escenas (pantallas) de la aplicación: *splash screen*, avatar, menú, juego del helado, niveles del helado, juego del chocolate, niveles del chocolate.

2) Añadir un controlador en todas las pantallas asociándolo a un script llamado *NombreEscenaController* (por ejemplo, *SplashSceneController*). Este script servirá para enviar datos entre pantallas. Además, asociar este script a los botones utilizando *SceneManagement* para permitir el desplazamiento entre las diferentes escenas.

3) Se han creado 15 scripts que controlan el comportamiento de objetos, lanzan animaciones o gestionan datos entre pantallas. Cada uno de estos scripts desempeña esta función:

- Controladores de las diferentes pantallas:

a. Script *SplashSceneController*:

Únicamente muestra la pantalla inicial de la aplicación durante 5 segundos y pasa a la siguiente escena. Si es la primera vez que se juega, pasará a la escena de Avatar para rellenar los datos. En cambio, si ya se ha jugado antes y hay datos del usuario guardados, irá directamente a la escena de menú con los datos anteriormente guardados.

b. Script *AvatarSceneController*:

Si es la primera vez que se juega, el panel para poner el nombre estará vacío. Habrá un avatar creado por defecto que se puede modificar pulsando los botones del tipo de piel, pelo o cuerpo. La función que desempeñan los botones es activar o desactivar las diferentes capas del propio avatar para así crear uno personalizado. Una vez se pulsa el botón de siguiente, todos estos datos se envían hacia la escena del menú.

En cambio, si ya se ha jugado anteriormente, desde el menú es posible volver a acceder a esta pantalla, y se vuelven a enviar los datos guardados anteriormente, y en el panel de nombre aparecerá el nombre guardado anteriormente. El botón situado al lado del nombre simboliza “deshacer”, por si el jugador quiere volver a tener los datos anteriormente guardados y no cambiarlos, por lo tanto, se cargan de nuevo los datos de la pantalla anterior.

c. Script *MenuSceneController*:

Por un lado, recupera el avatar anteriormente creado y escribe el nombre del usuario, además de actualizarlo en el script que gestiona los datos. Por otro lado, recupera la puntuación total guardada en el script que gestiona los datos estáticos.

Hay dos botones, el del juego del helado y el del juego del chocolate, por lo tanto, hay una función asociada a cada uno de los botones que te redirige a las respectivas pantallas de los niveles.

d. Script *HeladoNivelesSceneController*:

Cada uno de los niveles de esta pantalla son un botón que por defecto están desactivados (excepto el del nivel 1 que siempre está activado). Estos botones (niveles) se activan cuando en el nivel anterior se han conseguido más de 3 puntos.

Además de activarse el nivel, se desactiva la imagen del candado y se activa el número del nivel, y las estrellas pasan a un color más oscuro.

Al guardarse la puntuación estáticamente, desde este script se calcula cuantas estrellas se han de encender según la puntuación obtenida en las partidas anteriormente jugadas (dos o tres puntos equivalen a una estrella, más de tres puntos y menos de nueve equivalen a dos estrellas, y nueve puntos equivalen a las tres estrellas).

Cada vez que se entra en un nivel, se reinicia la puntuación de la partida a 0, se envía un *string* hacia el script de la pantalla de juego para indicar el número del nivel que ha escogido el usuario para jugar, y se redirige a la escena del juego.

e. Script *HeladoSceneController*:

Este script gestiona el juego del helado. Para empezar, se comprueba cual es el nivel escogido por el usuario (mediante el *string* recibido del script anterior). Dependiendo del nivel escogido, la escena se cargará con unos datos diferentes, es decir, cambiará el valor final de la barra dónde se tiene que

colocar el helado, y también cambiará el número aleatorio que indica la posición donde colocar el helado.

En los niveles 2, 4 y 6 una de las condiciones es que no se repita en la misma partida el número aleatorio dónde se tiene que colocar el helado, por lo que se crean unas listas que se guardan estáticamente y se reinician cuando acaba la partida.

Cada partida son nueve intentos, por lo tanto, la escena se cargará nueve veces y volverá a la pantalla de niveles, enviando la puntuación de la partida actual, y estos puntos se sumarán al score total. En el caso de pulsar el botón de salir, se dirigirá a la pantalla de niveles, poniendo el contador de los intentos a 0 y sin contar los puntos hechos en la partida no acabada.

El helado contiene el script "MoverHelado" que le permite el movimiento únicamente por la barra.

Al pulsar el botón OK se verifican diferentes aspectos:

- La barra donde se coloca el helado va de -6 a 6 en x. Se hace un offset de la recta para convertirlo en una barra de 0 a 10. Para ello he cogido la posición en X del helado, le he sumado 6 y dividido entre 1,2, y obtengo la posición entre 1 y 10. Para los niveles superiores que la barra va de 0 a 20 he hecho una regla de tres, cogiendo esta posición entre 1 y 10, multiplicándolo por el número final de la barra (por ejemplo, si la barra es de 0 a 20, multiplicándolo por 20) y dividiéndolo entre 10.
- Se comprueba en el nivel en el que se está jugando. Si la posición del helado está cerca del número aleatorio, se da por correcto: se suma un punto a la puntuación actual de la partida, se lanza la animación de ganar el helado y se encienden todos los fuegos excepto el situado en la posición correcta. En cambio, si no es correcto, el helado se derretirá y durante los dos primeros niveles se encenderán todos los fuegos menos el correcto, para dar pistas de dónde tendría

que haber ido. En los siguientes niveles se encenderá el fuego que está justo debajo.

- Cuando el contador haya sumado 9 intentos, se cargará la pantalla de los niveles. Se mirará el nivel en el que se ha jugado la partida y se enviará un *int* hacia la pantalla de niveles con los puntos ganados (para poder hacer la gestión de las estrellas y desbloqueo de niveles). También se sumarán los puntos conseguidos al total de puntos de la pantalla de “Menú”.

Al pulsar el botón de la pregunta, se activa la función de ayuda, que pertenece a la misma escena del helado. Esta función activa una imagen con sus respectivas animaciones y textos, y al pulsar OK se desactivan y se sigue jugando en el punto donde se había quedado.

f. Script *ChocolateNivelesSceneController*.

Cada uno de los niveles de esta pantalla, igual que en la pantalla de niveles del helado, son un botón que por defecto están desactivados (excepto el del nivel 1 que siempre está activado). Estos botones (niveles) se activan cuando en el nivel anterior se han conseguido más de 3 puntos.

Además de activarse el nivel, se desactiva la imagen del candado y se activa el número del nivel, y las estrellas pasan a un color más oscuro.

Al guardarse la puntuación estáticamente, desde este script se calcula cuántas estrellas se han de encender según la puntuación obtenida en las partidas anteriormente jugadas (dos o tres puntos equivalen a una estrella, más de tres puntos y menos de nueve equivalen a dos estrellas, y nueve puntos equivalen a las tres estrellas).

Cada vez que se entra en un nivel, se reinicia la puntuación de la partida a 0, se envía un *string* hacia el script de la pantalla de

juego para indicar el número del nivel que ha escogido el usuario para jugar, y se redirige a la escena del juego.

g. Script *ChocolateSceneController*:

En este script se gestiona el juego del chocolate. Igual que en el juego del helado, para empezar, se comprueba cual es el nivel escogido por el usuario mediante el *string* recibido. Depende del nivel escogido, la escena se cargará con diferentes valores, es decir, cambiarán las series de la tableta y la distancia entre los números de la serie, y por consecuencia, también cambiarán los números de las onzas que hay que colocar en la tableta.

Cada vez que se carga la escena, las tres onzas de chocolate que hay en la parte inferior cambian de posición y se colocan aleatoriamente para evitar que el número correcto este siempre en el mismo sitio.

Se crea una lista para cada uno de los niveles, y se escoge un numero aleatorio de la lista, y se elimina. Por ejemplo, si el nivel seleccionado es una serie de distancia 1, el primer número de la tableta será el numero aleatorio (ver fig. y el tercer numero de la tableta será el numero aleatorio +2. Las onzas de chocolate, una de ellas será el número aleatorio +1 (será la onza correcta), que además tendrá un “tag” llamado “resultado”. Los números de las otras 2 onzas serán escogidos de una lista en la que habrá el numero aleatorio -1, +4 y +5 (para evitar que siempre sea igual).

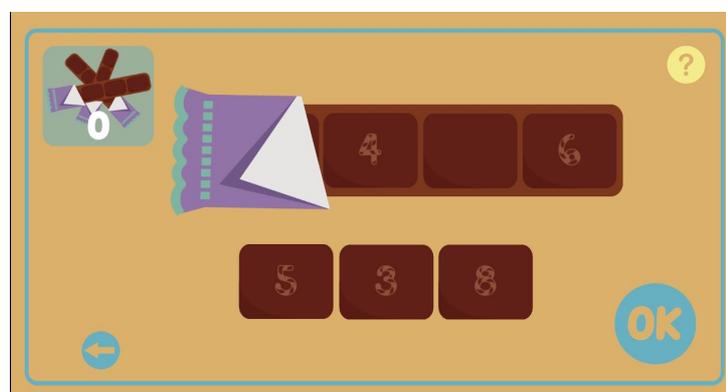


Fig. 28: Pantalla del juego del chocolate

En los niveles que las series sean de distancia 2, 3 o 5, para escoger los números de las 2 onzas de chocolate que no son la correcta se escogen de una lista con más opciones. Por ejemplo, si es de distancia 2, la lista estará compuesta por el número aleatorio -1, +1, +3 y +5. Y así con los siguientes niveles.

Las onzas inferiores i la superior que esta vacía contienen dos scripts (Slot y DragHandler) para permitir el *drag and drop* que se explicarán a continuación.

El botón OK se mantiene desactivado mientras la onza a rellenar esta vacía, y se vuelve a activar una vez tiene algún elemento dentro.

Cada partida son nueve intentos, por lo tanto, la escena se cargará nueve veces y volverá a la pantalla de niveles, enviando la puntuación de la partida actual, y estos puntos se sumarán al score total. En el caso de pulsar el botón de salir, se dirigirá a la pantalla de niveles, poniendo el contador de los intentos a 0 y sin contar los puntos hechos en la partida no acabada.

Igual que en la pantalla del helado, cada partida son nueve intentos, por lo tanto, la escena se cargará nueve veces y volverá a la pantalla de niveles, enviando la puntuación de la partida actual, y estos puntos se sumarán al score total. En el caso de pulsar el botón de salir, se dirigirá a la pantalla de niveles, poniendo el contador de los intentos a 0 y sin contar los puntos hechos en la partida no acabada.

Cuando se pulsa el botón de OK para verificar el resultado:

- Primero se identifica cual es el nivel que se esta jugando y seguidamente se comprueba si la onza que está colocada lleva el “tag” de “resultado” mencionado anteriormente. Si es así, se suma un punto y se lanza el sonido y la animación de ganar.

- Si es incorrecto, se lanza el sonido y la animación de perder.

Al pulsar el botón de la pregunta, se activa la función de ayuda, que pertenece a la misma escena del chocolate. Esta función activa una imagen con sus respectivas animaciones y textos, y al pulsar OK se desactivan y se sigue jugando en el punto donde se había quedado.

- Lanzadores de Animaciones

- h. Script *AnimacionHelado*:

Si se llama a este script se pueden lanzar la animación de ganar un helado (ganar), o la animación de deshacerlo (perder).

- i. Script *AnimacionHeladoAyuda*:

Este script solo es llamado cuando la pantalla de ayuda esta activo, y se ejecuta repetidamente.

- j. Script *AnimacionChocolate*:

Igual que la animación del helado, hay una animación de perder y otra de ganar que se lanza cada vez que este script es llamado.

- k. Script *AnimacionFuego*:

Este script se lanza cada vez que se carga la pantalla del helado, encendiendo los fuegos pequeños. Cuando se gana, enciende todos los fuegos menos el de la posición ganadora. En cambio, cuando se pierde, se enciende el de la posición perdedora.

- Controladores de objetos:

- l. Script *MoverHelado*:

Controla el movimiento del helado. Se define la velocidad en la que va el helado al hacer el movimiento, y los límites de la pantalla por los que puede moverse, en este caso, solo se puede mover sobre la barra.

Scripts que permiten el movimiento *drag and drop* de las onzas de chocolate, estas onzas están compuestas por 2 paneles y un texto (el número):

m. Script *DragHandler* [7]:

Este script se vincula al panel 2 (hijo).

n. Script *Slot* [7]:

Este script se vincula al panel 1 (padre).

- Gestión de datos estáticos

o. Script *GuardarDatos*:

Este script guarda todos los datos estáticamente. Es el único que no está asociado a ningún objeto ni escena. Lo he creado porque así he podido acceder a él desde todos los scripts y modificar los datos necesarios. Estos datos son:

- Intentos jugados en las partidas
- Puntuación general (menú).
- Puntuación diferentes niveles para poder gestionar las estrellas obtenidas.
- Listas numéricas para gestionar que no haya repeticiones a la hora de elegir los números aleatorios.

4) Cuando ya estaba toda la aplicación creada, me he dedicado a poner sonidos en todos los botones de la aplicación, añadir un sonido en el momento de ganar un objeto, ya sea un helado o chocolate, y otro sonido de perder. Como he dicho antes los sonidos son los únicos elementos que me he descargado desde la “Assets Store” [5] de Unity.

5) Para finalizar, el último mes me he dedicado a solucionar fallos y mejorar la aplicación.

Estudio económico

Cálculo y planificación de todos los gastos en el proyecto de “Funny Maths”.



Costes de desarrollo:

	€/hora	Horas	Euros
Rol <i>Game Designer</i>	15	15	225€
Rol Diseñadora	15	50	300€
Rol Desarrolladora	20	160	3200€
Rol Documentación	10	50	500€
Total			4225€

Costes fijos:

En los costes fijos sólo se computan las licencias de software que no son gratuitas (que no es el caso de Adobe Color, Unity y Visual Studio).

	€/mes	Meses	Euros
Adobe Illustrator	24,19	5	120,95 €
Macbook Pro	158,33	5	791,66 €
Total			912,61 €

Los costes de desarrollo finales han sido de 4.225€. Sumándole los costes fijos del ordenador y la licencia de Adobe Illustrator el proyecto suma a **5137,16€**.

Conclusiones

Como se ha podido ver en esta memoria, dónde se explica el procedimiento y el resultado final de “Funny Maths”, una aplicación de minijuegos para niños con discalculia, he podido cumplir con los objetivos propuestos en el inicio del proyecto.

Hacer este proyecto ha supuesto para mi un reto personal, ya que desde mediados del grado multimedia me planteé poder crear una aplicación desde cero, diseñándola y programándola. Ahora he visto que soy capaz de hacerlo y gracias a ella me han ido surgiendo ideas para el futuro.

Además, este proyecto será la parte más importante de mi portfolio, dónde demuestro capacidades para la gamificación, el diseño de interfaces y de objetos y por último mi faceta de desarrolladora.

La parte más motivadora de este proyecto ha sido hacerla para que los niños puedan mejorar en algo, en este caso, mejorar su dificultad de aprender matemáticas.

Futuras mejoras

Más adelante, cuando se compruebe que este demostrador puede funcionar, el siguiente paso sería hacer un minijuego por cada uno de los ejercicios mencionados en el apartado de contextualización: subitizing, identificación numérica, transcodificación, memoria de trabajo numérica, sistema base 10 y cálculo mental.

Para tratar la parte más sociable del juego, los niños podrían crear amigos (más avatares) en el juego. De esta manera todas las golosinas que van ganando las pueden repartir con sus amigos. El comportamiento de estos amigos podría ser como el de un *tamagochi*, si los alimentas de golosinas estarán contentos, y de lo contrario se pondrán tristes. Por lo tanto, entre más golosinas ganen (en consecuencia, entre más jueguen y mejoren), más amigos podrán crear. De esta manera también se podría investigar si los niños son generosos a la hora de repartir las golosinas, o se las quedan para alimentar a su avatar propio.

Para poder comprobar si funciona “Funny Maths”, el siguiente paso sería enviar toda la información del juego a una base de datos para después poder hacer métricas y saber si los niños están mejorando o no.

Por último, traducir “Funny Maths” a varios idiomas, para poder llegar a un número mas amplio de niños.

Web grafía

Gamificación:

[1] <https://www.educaciontrespuntocero.com/noticias/gamificacion-que-es-objetivos/70991.html>

Serious Games:

[2] <https://www.game-learn.com/lo-que-necesitas-saber-serious-games-game-based-learning-ejemplos/>

Discalculia:

[3] <https://integratek.es/que-es-la-discalculia/>

[4] <http://www.aboutdyscalculia.org>

Assets de sonido de Unity:

[5] <https://assetstore.unity.com/>

Documentación de Unity (utilizada en general para el desarrollo del código de la aplicación):

[6] <https://docs.unity3d.com/Manual/index.html>

Drag and Drop (pantalla de chocolate):

[7] https://www.youtube.com/watch?v=Pc8K_DVPgVM