



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Desarrollo del juego Sushi Go para móviles Android

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Muñoz Jiménez, Rosana

Tutor/a: Pelechano Ferragud, Vicente

CURSO ACADÉMICO: 2023/2024



Resumen

Este proyecto se centra en adaptar el juego de cartas "Sushi Go" para dispositivos Android utilizando Unity y programando en C#. El objetivo es crear una experiencia de juego atractiva y fácil de usar. El proceso de desarrollo implica etapas incrementales, abordando desafíos gráficos y de programación, todo ello realizado mediante el uso de metodologías ágiles. Los resultados incluyen un juego funcional que mantiene la esencia del original, destacando la interfaz intuitiva y la jugabilidad rápida y casual.

Palabras clave: Sushi Go, Android, Unity, C#, juegos de mesa, desarrollo de juegos, Kanban, adaptación, diseño gráfico, interfaz de usuario, experiencia de juego,, aplicaciones móviles.

Abstract

This project focuses on adapting the card game "Sushi Go" for Android devices using Unity and C# programming. The goal is to create an attractive and user-friendly gaming experience. The development process involves incremental stages, with graphic and programming challenges, all managed using agile methodologies. The results include a functional game that preserves the essence of the original, with an emphasis on an intuitive interface and quick, casual gameplay.

Keywords: Sushi Go, Android, Unity, C#, board games, game development, Kanban, adaptation, graphic design, user interface, gaming experience, mobile applications.



Índice general

1. Introducción	13
1.1. Motivación.....	14
1.2. Objetivos.....	16
1.3. Metodología.....	16
1.4. Estructura.....	18
2. Sushi Go! : El juego de cartas	20
2.1. Cómo se juega.....	20
2.2. Tipos de sushi.....	21
2.3. Estética.....	23
3. Estado del arte	24
3.1. Juegos similares.....	24
3.2. Propuesta.....	29
4. Tecnología utilizada	30
4.1. Diseño gráfico.....	30
4.2. Animación.....	32
4.3. Sonido.....	33
4.4. Motor gráfico.....	34
4.4.1. Conceptos básicos de Unity.....	35



4.5. Gestión de versiones	37
4.6. Gestión de autenticación de usuarios.....	38
4.7. Editor de código.....	38
4.8. Lenguaje	40
5. Análisis de requisitos	41
5.1. Identificación y análisis de soluciones posibles.....	43
5.1.1. Enfoque.....	43
5.1.2. Orientación del dispositivo móvil.....	44
5.1.3. Puntuaciones	44
5.2. Solución propuesta	45
5.3. Fases del desarrollo	46
6. Diseño de la solución	47
6.1. Diseño gráfico.....	47
6.2. Diseño del sonido	48
6.3. Diseño de la interfaz	49
6.3.1. Menú principal.....	49
6.3.2. Explicación de los combos.....	50
6.3.3. Explicación del juego	51
6.3.4. Juego	51
6.3.5. Puntuaciones	54
6.3.6. Selección de jugadores.....	54



6.3.7. Registro de usuarios	55
6.4. Diseño detallado	56
6.4.1. MenuScene	58
6.4.2. CombosScene	59
6.4.3. HowToPlayScene	61
6.4.4. LoginScene	62
6.4.5. SelectPlayersScene	64
6.4.6. VideoRoundScene	65
6.4.7. GameScene	67
6.4.8. TitleRoundScene	73
6.4.9. ScoreRoundScene	73
6.4.10. WinnerLoserScene	74
7. Incrementos del software	76
7.1. Primer incremento	76
7.2. Segundo incremento	77
7.3. Tercer incremento	79
7.4. Cuarto incremento	80
7.5. Quinto incremento	82
7.6. Sexto incremento	84
7.7. Séptimo incremento	86
8. Problemas y desafíos técnicos	89



9. Conclusión	93
9.1. Relación con los estudios cursados	94
10. Trabajos futuros	96
11. Referencias	97



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Escola Tècnica
Superior d'Enginyeria
Informàtica



etsinf



Índice de figuras

Figura 1: Juego de cartas Sushi Go!	19
Figura 2: Cartas del Sushi Go!.....	19
Figura 3: Tipo de sushi en Sushi Go!.....	22
Figura 4: Gameplay de Crossy Road.....	23
Figura 5: Arte de Crossy Road.....	23
Figura 6: Logo de Solitario.....	24
Figura 7: Gameplay de Solitario	24
Figura 8: Logo de Uno!	25
Figura 9: Gameplay de Uno!.....	25
Figura 10: Logo de Piffle	25
Figura 11: Gameplay de Piffle.....	25
Figura 12: Logo de Tsuki Adventure	26
Figura 13: Gameplay de Tsuki Adventure	26
Figura 14: Logo de Flappy Bird.....	27
Figura 15: Gameplay de Flappy Bird	27
Figura 16: Logo de Adobe Illustrator.....	30
Figura 17: Programa Adobe Illustrator	30
Figura 18: Logo de Figma.....	31
Figura 19: Página web de Figma	31
Figura 20: Logo de Adobe After Effects	33
Figura 21: Programa Adobe After Effects	33



Figura 22: Logo de GarageBand	34
Figura 23: Programa GarageBand.....	34
Figura 24: Logo de Unity.....	35
Figura 25: Programa Unity.....	35
Figura 26: Logo de Git	37
Figura 27: Logo de GitHub.....	37
Figura 28: Programa GitHub Desktop.....	37
Figura 29: Logo de Firebase Authentication	38
Figura 30: Consola de Firebase Authentication	38
Figura 31: Logo de Visual Studio Code	40
Figura 32: Programa Visual Studio Code	40
Figura 33: Logo de C#	40
Figura 34: Ejemplo de código en C#.....	40
Figura 35: Historias de usuario	42
Figura 36: Backlog	42
Figura 37: Paleta de colores	48
Figura 38: Patrón japonés.....	49
Figura 39: Boceto menú principal	51
Figura 40: Boceto combos	51
Figura 41: Boceto explicación del juego	52
Figura 42: Propuesta nº1 - pantalla de juego	53
Figura 43: Propuesta nº2 - pantalla de juego	53



Figura 44: Propuesta nº3 - pantalla de juego	54
Figura 45: Propuesta nº4 - pantalla de juego	54
Figura 46: Boceto pantalla de juego	55
Figura 47: Boceto puntuaciones	55
Figura 48: Boceto seleccionar número jugadores	56
Figura 49: Boceto registro de usuario	57
Figura 50: Diagrama de flujo de escenas	58
Figura 51: Diagrama de flujo de una partida.....	59
Figura 52: Interfaz MenuScene.....	60
Figura 53: Interfaz ComboScene - menuCombo	62
Figura 54: Interfaz ComboScene - combo	62
Figura 55: Interfaz HowToPlayScene.....	63
Figura 56: Interfaz LoginScene - inicio de sesión	64
Figura 57: Interfaz LoginScene - registro.....	65
Figura 58: Interfaz LoginScene - recuperar contraseña	65
Figura 59: Interfaz LoginScene - perfil.....	66
Figura 60: Interfaz SelectPlayersScene.....	67
Figura 61: Interfaz VideoRoundScene	68
Figura 62: Diagrama de flujo de una partida.....	69
Figura 63: Interfaz GameScene - partida.....	70
Figura 64: Interfaz GameScene - platos	70
Figura 65: Interfaz GameScene - sushi seleccionado	71
Figura 66: Diagrama de clases UML.....	72



Figura 67: Interfaz TitleRoundScene	75
Figura 68: Interfaz ScoreRoundScene.....	76
Figura 69: Interfaz WinnerLoserScene	77
Figura 70: GameScene - 1 ^{er} incremento	80
Figura 71: GameScene - 2 ^o incremento.....	80
Figura 72: TitleRoundScene - 2 ^o incremento	81
Figura 73: ScoreRoundScene - 2 ^o incremento.....	81
Figura 74: GameScene - 3 ^{er} incremento	82
Figura 75: GameScene 2 - 3 ^{er} incremento	83
Figura 76: SelectPlayersScene - 4 ^o incremento.....	83
Figura 77: VideoRoundScene - 4 ^o incremento.....	84
Figura 78: GameScene - 4 ^o incremento.....	84
Figura 79: GameScene - 5 ^o incremento.....	85
Figura 80: GameScene 2 - 5 ^o incremento.....	85
Figura 81: ScoreRoundScene - 5 ^o incremento.....	86
Figura 82: GameScene 3 - 5 ^o incremento.....	86
Figura 83: GameScene - 6 ^o incremento.....	87
Figura 84: ScoreRoundScene - 6 ^o incremento.....	87
Figura 85: MenuScene - 6 ^o incremento.....	88
Figura 86: SelectPlayersScene - 6 ^o incremento.....	88
Figura 87: GameScene - 6 ^o incremento.....	88
Figura 88: WinnerLoserScene - 7 ^o incremento	89
Figura 89: HowToPlayScene - 7 ^o incremento.....	89



Figura 90: CombosScene - 7º incremento	90
Figura 91: Detalle en MenuScene - 7º incremento	90
Figura 92: LoginScene - 7º incremento.....	91



CAPÍTULO 1

Introducción

Los smartphones y demás dispositivos móviles se han convertido en parte de nuestro día a día con gran rapidez. En la actualidad nos sería muy difícil concebir nuestra rutina habitual sin uno de estos aparatos, ya que nos pueden sacar de problemas rápidamente o hacernos más amenos algún rato muerto como esperar el transporte público o desconectar un rato del estudio.

Los juegos para móviles han sido una tendencia al alza. Ya sean aficionados a los videojuegos o jugadores casuales, sería raro encontrar a alguien sin algún tipo de juego instalado en su móvil que les ayude a evadirse un rato.

A pesar de la creciente variedad de géneros que podemos encontrar cuando vayamos a escoger una de estas aplicaciones, los más populares son los juegos casuales [1], típicamente distinguibles por sus reglas simples y el poco compromiso que requieren del jugador, en contraste con otros juegos, más complejos, que requieren una dedicación de tiempo a largo plazo o habilidades especiales para jugar. En resumen, dedicado a la gente que quiere jugar una o dos partidas rápidas en un momento en concreto, sin tener que preocuparse por dedicarle un esfuerzo adicional, perfecto para desconectar por un momento y sin tener más preocupaciones [2].

Todas estas características las podemos ver reflejadas también en algunos juegos de mesa, partidas rápidas para divertirse un rato sin tener que aprender instrucciones muy complicadas para pasar un rato agradable



con familiares y amigos.

El Sushi Go! [3] es uno de ellos, un juego de cartas rápido y de reglas sencillas. En cada partida tendremos que conseguir la máxima puntuación, recogiendo las piezas de sushi que nos aparezcan en cada turno para hacer combinaciones y conseguir mayor cantidad de puntos que nuestros oponentes.

Debido a la facilidad para dominarlo y su diseño cuidado y atractivo, el Sushi Go! ha alcanzado cierto nivel de fama [4], lo que se ha traducido en nuevas entregas, como expansiones para el juego de cartas y juegos de mesa [5][6][7][8].

Es por todo esto que la idea de realizar una adaptación para Android parecía el paso lógico para este aclamado juego de cartas.

1.1. Motivación

Desde pequeña he sido una gran aficionada a los juegos de mesa, desde los juegos más complicados donde se requieren varias partidas para lograr dominar el juego, hasta los juegos sencillos para pasar un breve pero agradable rato jugando con familiares y amigos. Esta pasión fue compartida más adelante por los videojuegos, en los que siempre he podido encontrar un refugio y he disfrutado pasando el tiempo tanto sola como acompañada.

En una reunión de amigos, pude descubrir el juego de cartas Sushi Go!, juego que desconocía pero del que rápidamente aprendí las mecánicas. Por el camino fácil, podías ir viendo las manos que te llegaban y coleccionar los pequeños puntos que te otorgaban, por el contrario, cuando todo esto ya lo tenías dominado, podías centrarte en las estrategias de los otros jugadores e intentar ir quitándoles las cartas vitales para su victoria, aumentando las posibilidades de la tuya. Esa tarde, en la que se planeaba un tour por varios



juegos de mesa más, acabó monopolizada por el Sushi Go!.

Tras este primer encuentro, el Sushi Go! se convirtió en un juego recurrente con amigos. Cuál fue mi sorpresa cuando, una vez en mi casa, intenté buscar este juego en la tienda de aplicaciones y no encontré nada.

Por ello, un amigo de la carrera y yo decidimos emprender la aventura de crear nuestra propia adaptación al móvil durante el primer año del grado. Para el diseño y el arte podríamos recurrir a mis conocimientos de diseño gráfico y él añadiría al proyecto su experiencia creando pequeños juegos y programas en su adolescencia. Desgraciadamente, debido a nuestra gran carga de trabajo y nuestros aún no suficientes conocimientos en el campo, tuvimos que abandonar el proyecto por el momento.

Cuando decidí retomarlo, mi antiguo compañero de proyecto ya se encontraba embarcado en otro desarrollo. Viendo la gran cantidad de tutoriales e información que podía obtener fácilmente sobre el desarrollo en Unity, me aventuré a volver empezarlo de cero yo sola, reutilizando únicamente algunos diseños del juego original que ya había conseguido vectorizar anteriormente para usarlos como sprites.

Debería combinar mis habilidades como diseñadora gráfica para crear una interfaz atractiva para el usuario a la par que intuitiva debido a la naturaleza casual del videojuego que quería conseguir, sin perder la temática de restaurante japonesa que desprendía el juego original, Por otro lado, también podría necesitar de la creación de animaciones para dotar al videojuego de esa esencia adorable que desprende el original y, en la parte más técnica, debería aprender a manejar con soltura el motor de Unity y la programación en C#. Era sin lugar a dudas un reto que pondría a prueba mis habilidades y me daría como resultado una prueba visible de mi valía y de la que estar orgullosa.



1.2. Objetivos

Este proyecto tiene como objetivo principal desarrollar el videojuego del Sushi Go!, manteniendo su esencia, tanto estética como conceptualmente, creando una interfaz fácil de entender para el usuario, siguiendo la línea de partidas rápidas y casuales de su versión física.

1.3. Metodología

La metodología de desarrollo elegida para este trabajo es la Kanban [9]. Esta palabra proviene del japonés, "Kan" que significa visual y "Ban" que significa tarjeta.

De esta forma, mediante la metodología Kanban se podrá gestionar de manera visual la carga de trabajo usando un tablero. Dicho tablero se divide en columnas, cada una representando una etapa del proyecto y por las cuales se irán trasladando las diferentes tareas. Generalmente se encuentra el tablero dividido en tres columnas: To do (pendiente), In progress (en progreso) y Done (terminado).

Los principios de la metodología Kanban son:

- Calidad garantizada: La idea de esta metodología es no tener que rehacer trabajo, por ello se prima la calidad antes que la rapidez. Este principio esta basado en el hecho de que muchas veces requiere más tiempo arreglar algo que hacerlo bien desde el principio.
- Reducción del desperdicio: El objetivo de esto sería realizar únicamente lo esencial y necesario. Esto implica eliminar todo lo que sea superficial o secundario, siguiendo el principio "You Ain't Gonna Need It" (YAGNI), que sugiere evitar implementar funcionalidades que no son estrictamente necesarias en el momento, centrándose en lo que realmente



agrega valor.

- Mejora continua: Con base en los objetivos que se busca alcanzar, Kanban no solo ayuda a organizar y visualizar tareas, sino que se orienta hacia la optimización constante, adaptándose fácilmente a los cambios que puedan surgir y buscando formas de mejorar la eficiencia y la calidad en el proceso de trabajo.

- Flexibilidad: Esta es una característica importante para la metodología Kanban, ya que se permite poder reordenar el orden de tareas en la cola de espera según las necesidades del momento.

En un videojuego, y sobretodo si es el primero que se desarrolla, es fácil darse cuenta de nuevas funcionalidades que se podrían añadir pero no son estrictamente necesarias, por lo que priorizar correctamente la cola de trabajo ayudaría a centrarse en los más importante.

Se ha seleccionado esta metodología porque, al ser un desarrollo realizado por una persona, no se van a realizar tareas a la vez en cooperativo ni se van a concluir simultáneamente, por lo que una única cola de tareas daría la información necesaria.

También, poder visualizar la columna "Done" o de tareas terminadas, ayudará a la hora de probar el juego y encontrar errores, tanto en los nuevos aspectos implementados como en la comprobación de que siguen funcionando correctamente los anteriores.



1.4. Estructura

Esta memoria del TFG consta de diez capítulos. A continuación se realizará un breve resumen de cada uno de estos apartados para explicar qué se puede encontrar en cada uno de ellos:

- **Introducción:** En este capítulo se explica el contexto de la idea del proyecto, así como la motivación que llevó a su desarrollo y los objetivos que se esperan alcanzar.
- **Sushi Go!:** Este capítulo sirve para conocer con más detalle el juego de cartas en el que se basa este desarrollo, cuya información facilitará la comprensión del resto de la memoria.
- **Estado del arte:** A continuación se hablará de juegos existentes similares al que se ha buscado desarrollar, estableciendo al final una propuesta clara de la aplicación que se busca.
- **Tecnología utilizada:** En este capítulo se razonará sobre la tecnología utilizada en los diferentes apartados de desarrollo del videojuego. En el apartado del motor gráfico se hará una breve introducción a sus conceptos básicos para comprender mejor sus particularidades.
- **Análisis de requisitos:** Una vez clara la propuesta y la tecnología que se va a utilizar, en este capítulo se procederá a realizar una propuesta más sólida basándose en la funcionalidades que deberá abarcar la aplicación.
- **Diseño de la solución:** Este capítulo estará dedicado a comprender la estructura del proyecto en detalle, cómo se ha diseñado y a entender la lógica que lo compone.
- **Incrementos:** En este capítulo se comentarán los incrementos realizados durante el desarrollo, explicando paso a paso la evolución que ha ido sufriendo el producto.



- **Desafíos y problemas técnicos:** En este capítulo se hablará de algunas dificultades que se han tenido durante el desarrollo de la aplicación.
- **Trabajos futuros:** A continuación se expondrán los planes de futuro para el desarrollo del proyecto.
- **Conclusión:** Se valorará el resultado final de este desarrollo, así como la experiencia personal obtenida y la relación de este proyecto realizado con los estudios cursados durante la carrera.



CAPÍTULO 2

Sushi Go! : El juego de cartas

Debido a que el objetivo de este proyecto es el desarrollo de una adaptación de un juego de cartas físico, lo primero de todo sería tener claros los conceptos del juego de mesa que se va a adaptar.

En el juego de cartas Sushi Go, el jugador se encuentra ambientado en un kaitenzushi [10], un restaurante de sushi japonés donde la comida es transportada en cintas y es el cliente el que va escogiendo a su gusto qué platos sacar de ella para comer.



Figura 1: Juego de cartas Sushi Go!



Figura 2: Cartas del Sushi Go!

2.1. Cómo se juega

Para emular la ambientación de este tipo de restaurante, en el juego se reparten cartas a todos los jugadores y, de esta mano, los jugadores escogerán una carta que revelarán todos al mismo tiempo. La carta escogida será puesta enfrente del jugador correspondiente y el resto del mazo se



pasará al jugador de la derecha. De esta forma, cada turno obtendrán un mazo del que escoger una pieza de sushi con una carta menos.

Una vez se acaben las cartas, se habrá acabado la ronda y se calcularán y anotarán los puntos ganados por cada uno de los jugadores. El juego consta de tres rondas, al final de las cuales se sumarán todos los puntos y se descubrirá el ganador.

En todo momento se podrá ver las cartas escogidas por todos los jugadores, que permanecen bocarriba delante suyo hasta el final de la ronda.

2.2. Tipos de sushi

Algunas cartas dan puntos por sí mismas, pero la mayoría se ven potenciadas por la combinación de varias, lo que hará que el jugador sume todavía más puntos. También hay determinadas cartas que solo suman al jugador que consiga más al final de la ronda o incluso pueden restar al que menos tenga al final del juego [11]. A continuación se detallan en profundidad los tipos de cartas existentes.

- Wasabi: Triplica el valor del siguiente nigiri que se selecciones. Si no se selecciona ningún nigiri después del wasabi, no da puntos.
- Nigiri de calamar: Otorga tres puntos. Si es primer nigiri seleccionado tras un wasabi, otorga nueve puntos.
- Nigiri de salmón: Otorga dos puntos. Si es primer nigiri seleccionado tras un wasabi, otorga seis puntos.
- Nigiri de tortilla: Otorga un punto. Si es primer nigiri seleccionado tras un wasabi, otorga tres puntos.



- Gyoza: Otorga puntos según el número de gyozas que se posean. Una gyoza otorga un punto, dos gyozas otorgan tres puntos, tres gyozas otorgan seis puntos, cuatro gyozas otorgan diez puntos y, de cinco gyozas en adelante, quince puntos.
- Palillos: Una vez hayas seleccionado los palillos, en un turno posterior podrás seleccionar dos piezas de sushi en tu turno en vez de una. En ese momento los palillos volverán a esa baraja de la que hayas seleccionado ambas piezas. Una carta de palillos no da puntos.
- Maki: El jugador con más maki al final de la ronda gana seis puntos, el segundo jugador con más rollos de makis, ganará tres.
- Tempura: Una pareja de tempura otorga cinco puntos. Una única tempura no da puntos. Se pueden puntuar varias parejas de tempura en la misma ronda.
- Sashimi: Un trío de sashimi otorga diez puntos. Un único sashimi o pareja de sashimis no dan puntos. Se pueden puntuar varios tríos de sashimi en la misma ronda.
- Puddin: Al finalizar las tres rondas, el que más puddin tenga gana seis puntos y el que menos tenga pierde seis puntos.



Figura 3: Tipo de sushi en Sushi Go!

2.3. Estética

El diseño artístico de las cartas refleja en el fondo la ambientación de las cintas del restaurante. Aunque las ilustraciones son caricaturescas, se incorporan elementos japoneses en el diseño. Esto incluye patrones y detalles que evocan la estética cultural japonesa.

Cada una de las cartas representa el tipo de sushi con un personaje de estética simpática que los hace muy reconocibles y atractivos para el jugador. Esto crea un ambiente relajado, lo cual es coherente con la naturaleza ligera del juego. Además, cada tipo de sushi tiene un esquema de colores distintivo, facilitando la identificación de las cartas durante el juego.



CAPÍTULO 3

Estado del arte

3.1. Juegos similares

Otros proyectos con los que se puede comparar este juego cuentan con partidas de corta duración, fáciles de entender, sin mucho avance o progreso entre partidas, ya que no se busca una gran implicación o habilidad por parte del usuario y, sobretodo, una interfaz atractiva que llame la atención del jugador.

3.1.1. Crossy Road

Crossy Road [12] es un juego casual de saltos del estudio Hipster Whale, lanzado en 2014 y desarrollado en Unity. En él se debe guiar al personaje principal para llegar lo más lejos posible, evitando árboles, trenes o diversos obstáculos que se encuentran en el camino. Cada paso que se logre avanzar, sumará un punto. También se pueden recoger monedas, que ayudarán a desbloquear nuevas apariencias o skins para nuestro personaje.



Figura 4: Gameplay de Crossy Road



Figura 5: Arte de Crossy Road



De este juego cabría destacar su apariencia y sencillez. Se parte de una premisa muy básica, muy sencilla, de la misma forma que la aplicación que se ha desarrollado pero, con una estética muy cuidada, lo que la hace atractiva visualmente. En ninguna de las dos hay muchos menús que estorben cuando se va a jugar una partida, lo que le da mucha ligereza al entrar a la aplicación.

3.1.2. Solitario

El Solitario [13] para dispositivos móviles presenta el famoso juego de cartas, de una forma fácil, rápida y accesible. En él es posible cambiar su apariencia, comparar puntuaciones con otros jugadores y realizar retos diarios.



Figura 6: Logo de Solitario



Figura 7: Gameplay de Solitario

Un juego sencillo, conocido por todo el mundo y que da al jugador justo lo que necesita, sencillez y partidas rápidas, características deseables también para la aplicación del SushiGo.

3.1.3. Uno!

El Uno! [14] es una adaptación del famoso juego de cartas para



dispositivos móviles. En él podemos jugar tanto en solitario como con amigos, diversos modos de juego, torneos online, etc.



Figura 8: Logo de Uno!



Figura 9: Gameplay de Uno!

Este juego presenta una interfaz llamativa para el jugador, muy acorde con la estética del conocido juego de cartas. Del mismo modo, también se ha tratado de dar la misma ambientación a la aplicación que se ha desarrollado que a su versión física.

3.1.4. Piffle

Piffle [15] es un juego para móvil y Nintendo Switch donde tu objetivo es destruir bloques lanzando pelotas y recolectar gatos. Cada bloque tiene un número que representa los golpes necesarios para destruirlo. Además de superar niveles, puedes personalizar a tu personaje, mejorar gatos y usar potenciadores.



Figura 10: Logo de Piffle



Figura 11: Gameplay de Piffle



Piffle combina habilidades estratégicas con elementos de recolección y personalización para una experiencia de juego entretenida.

Tanto la estética de Piffle como la del SushiGo, presentan gráficos atractivos. Su fácil aprendizaje y sencillez también es una característica común entre ambos.

3.1.5. Tsuki Adventure

Tsuki Adventure [16] es un juego de simulación y aventuras donde los jugadores siguen la vida de Tsuki, un conejo que se muda a un pueblo. Los jugadores experimentan la vida cotidiana de Tsuki, realizan actividades como pescar y cultivar, interactúan con otros personajes, y participan en eventos estacionales.



Figura 12: Logo de Tsuki Adventure



Figura 13: Gameplay de Tsuki Adventure

Ambos juegos tienen una estética japonesa y destacan por su carácter desenfadado.

3.1.6. Flappy Bird

Flappy Bird [17] fue un juego para móviles simple pero desafiante. Los jugadores controlaban un pájaro y debían dirigirlo a través de tuberías sin chocar. La mecánica de juego era un toque en la pantalla para hacer saltar el pájaro



Figura 14: Logo de Flappy Bird

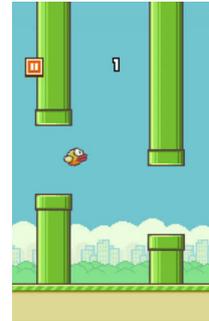


Figura 15: Gameplay de Flappy Bird

Destaca por su diseño sencillo, tanto visual como mecánicamente, y su enfoque en la repetición constante sin una progresión. Este enfoque minimalista fue parte de la razón de su atractivo y su adicción para muchos jugadores, por lo que se ha tomado como ejemplo para este desarrollo, ya que ambos juegos carecen de una progresión real.

Los juegos vistos anteriormente son un ejemplo para el desarrollo de esta adaptación. Posibles cosas a destacar y que servirían como puntos de referencia para este videojuego serían:

- Estética: Muchos de estos ejemplos presentan una estética muy cuidada y vistosa, cosa que le da un apariencia muy profesional a un juego tan sencillo. El juego del Uno!, al que se podría tomar también como una referencia por ser una adaptación de un juego de cartas, presenta una interfaz que hace recordar completamente el juego original, han sabido transmitir la misma ambientación y el resultado ha sido muy llamativo.

- Sin progreso: Casi ninguno de estos juegos presenta un avance real, una historia que seguir o un progreso, eso podría hacer que el jugador se canse de la aplicación y no vuelva tras usarla un par de veces. En estos casos, se las han ingeniado para dar al jugador algo a cambio de jugar, sin llegar a obligarlo a dedicarle demasiado tiempo.

- Sencillez: Los jugadores casuales no suelen haber tenido mucha experiencia con videojuegos complejos por lo que, si ese es el público al que va dirigido el



juego, es importante no marearlo con muchos menús u obstáculos antes de poder empezar una partida rápidamente. Todos los juegos anteriormente descritos son un ejemplo de accesibilidad, ya que sus menús son fáciles de entender y sencillos.

3.2. Propuesta

En resumen, el SushiGo es un juego de cartas que, aún tras 10 años de su lanzamiento, ha sacado contenido adicional y es posible encontrarlo fácilmente en establecimientos de venta de juegos de mesa. Por todo ello, es posible considerar el paso lógico de este juego a los dispositivos móviles, siempre conservando su frescura, ambientación japonesa y simpatía de los personajes.

Sería importante conseguir una interfaz de usuario sencilla y atractiva como en el resto de juegos casuales, ya que debería guiar al usuario para poder jugar partidas rápidas, además de darle una información concisa y clara.



CAPÍTULO 4

Tecnología utilizada

Para este apartado se diferenciará entre las diferentes tecnologías utilizada por las distintas partes de desarrollo del videojuego, por ello se hará un análisis de la tecnología a nivel diseño, animación, sonido y motor gráfico.

4.1. Diseño gráfico

En el apartado gráfico se necesita un programa para realizar los sprites del juego y diseñar las interfaces de usuario. La realización de sprites vectoriales resultará de gran ayuda, ya que no perderán calidad a la hora de una posible reescala. Su desarrollo por capas ayudará posteriormente la separación de los elementos de cada ilustración para poder editarlos por separado. También se usará una tableta gráfica para obtener los primeros bocetos de dichas ilustraciones.

Tras tener claro cuáles son las necesidades de diseño que se necesitan cubrir, estudian diversos programas que podrían cumplir con el propósito:

Inkscape [18] es un editor de gráficos vectoriales libre y de código abierto. Es un programa ligero, gratuito y todavía en desarrollo, se utiliza ampliamente en todo el mundo tanto por diseñadores profesionales como por aficionados para generar una diversidad de gráficos que emplea el formato SVG como su formato principal.

Gimp [19] es una aplicación de software de edición de imágenes de código



abierto y gratuito que se utiliza para crear, editar y retocar imágenes y gráficos. Cuenta con numerosas opciones de personalización y la posibilidad de utilizar complementos de terceros.

Adobe Illustrator [20] es una aplicación de software de diseño vectorial desarrollada por Adobe Inc. Se utiliza ampliamente en el mundo del diseño gráfico y la creación de ilustraciones vectoriales. Los archivos en Illustrator se guardan en formatos como AI (el formato nativo de Illustrator), PDF, SVG y otros, lo que facilita su uso en una variedad de aplicaciones y medios.

Teniendo en cuenta lo estudiado, se descarta Gimp, ya que está más enfocado en edición de imágenes de mapa de bits (raster). Esto significa que se centra en la manipulación de píxeles individuales y no trabaja con vectores, poco favorable a nuestros objetivos.

Entre Inkscape y Adobe Illustrator se ha elegido el éste último, debido a que ya ha sido utilizado con anterioridad, presenta una interfaz sencilla, tiene un buen soporte para el dibujo con tableta gráfica y, tiene buena sinergia con otros programas de Adobe en caso de necesitarlo.



Figura 16: Logo de Adobe Illustrator

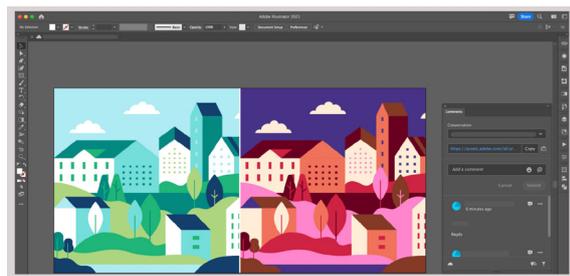


Figura 17: Programa Adobe Illustrator

4.2. Diseño de las interfaces

En el diseño de las interfaces se requiere un programa fácil de utilizar y bastante cómodo para poder tener la libertad de bocetar diversas propuesta de una misma interfaz sin dedicar mucho tiempo y pero dejando los conceptos clave claros para poder elegir siempre la elección más óptima y cómoda para el jugador.

Se estudiaron diversos programas para esta función:

Figma [21] es una plataforma de diseño colaborativo en la nube utilizada para crear interfaces de usuario, prototipos y diseño gráfico. Ofrece herramientas para diseño de interfaz de usuario y experiencia del usuario, prototipado interactivo, componentes reutilizables y comentarios integrados.

Adobe XD [22] es una herramienta de diseño y prototipado de interfaces de usuario creada por Adobe. Permite a los diseñadores crear elementos visuales, prototipos interactivos y colaborar en tiempo real.

Pese a ser los dos muy buenos programas, finalmente se escogió Figma, debido a su utilización durante la carrera en distintas ocasiones.

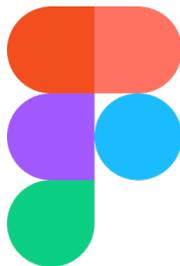


Figura 18: Logo de Figma

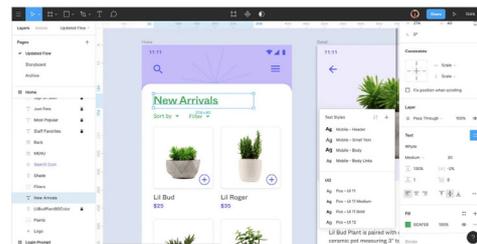


Figura 19: Página web de Figma

4.3. Animación

Para realizar las animaciones que se necesiten en la aplicación se necesita un programa al que se puedan importar ilustraciones por capas, lo que permitirá animar cada una de las partes por separado, dotándolas de mayor realismo,



teniendo en cuenta también los 12 principios de la animación, para dar un acabado más profesional.

Se estudian varios programas de animación para luego decidir cuál cumple mejor dichos requisitos:

Blender [23] es una aplicación de software que funciona en diversas plataformas y está diseñada principalmente para tareas como modelado, iluminación, renderizado, animación y creación de gráficos en 3D. También permite realizar composiciones digitales mediante la técnica de nodos, editar vídeos, esculpir y pintar digitalmente.

Adobe After Effects [24] es una aplicación que se utiliza principalmente para crear composiciones, aplicar efectos especiales y generar gráficos en movimiento de alta calidad. Desde su inicio, esta herramienta ha estado centrada en la superposición de capas, lo que permite combinar elementos visuales de manera creativa y profesional.

DaVinci Resolve [25] es una herramienta integral que abarca desde la edición y corrección de color hasta la mezcla de sonido y la postproducción en un solo paquete. Se utiliza en la industria del cine y la televisión, así como por profesionales y aficionados para editar vídeos, crear efectos visuales, graduar el color y realizar diversas tareas relacionadas con la postproducción de contenido audiovisual.

Tras estudiar las posibles opciones, primero se descartó Blender, ya que está más destinado a la creación desde cero, no a la edición de elementos ya existentes. Esto es un inconveniente ya que las ilustraciones para animar se trabajan en otro programa.

Después se eliminó DaVinci Resolve el que, aunque ofrece muchas características interesantes, Adobe After Effects puede ofrecer una combinación con el otro programa Adobe que vamos a utilizar para el diseño gráfico. Además, este software ha sido estudiado en la carrera y, como todos los programas Adobe,



presenta una interfaz muy similar.



Figura 20: Logo de Adobe After Effects



Figura 21: Programa Adobe After Effects

4.4. Sonido

Para editar el sonido del videojuego simplemente se necesita gestionar la duración de clips de audio y la creación de loops para tener la música del juego y los efectos de sonido listos para importar en Unity.

Los programas que se han estudiado para realizar estas tareas son:

GarageBand [26] es una aplicación de software de creación musical desarrollada por Apple Inc., se utiliza para crear, grabar y editar música y pistas de audio de manera intuitiva.

Audacity [27] es una aplicación de software de edición de audio de código abierto y gratuita. Es ampliamente utilizado para grabar y editar audio en una variedad de proyectos, desde grabaciones de podcasts y música hasta trabajos de postproducción de sonido.

Por su interfaz intuitiva y fácil de utilizar, además de por su gestión de loops, GarageBand ha sido el programa elegido para la edición del sonido.



Figura 22: Logo de GarageBand



Figura 23: Programa GarageBand

4.5. Motor gráfico

Se busca hacer un juego en 2D sin gráficos muy demandantes. Su desarrollo va a estar enfocado en Android pero en el futuro se podría valorar sacar el videojuego también disponible para iOS, por lo que es una características también a valorar.

Teniendo claras las necesidades de desarrollo, se van a estudiar los principales motores gráficos actualmente.

Unreal Engine [28], desarrollado por Epic Games, cuenta con un apartado gráfico mucho más potente que Unity pero un desarrollo más enfocado a usuarios experimentados, con herramientas avanzadas y el C++ como lenguaje de programación.

Por otro lado, Unity [29], desarrollado por la empresa Unity Technologies, presenta una curva de aprendizaje más sencilla, el uso de los lenguajes C# y Javascript y un desarrollo multiplataforma, lo cual podría representar una fuerte ventaja de cara a realizar un lanzamiento tanto para iOS como para Android en el futuro.

Debido a estos puntos y la poca necesidad de un motor con gráficos muy potentes para el videojuego que se pretende realizar, Unity fue el motor elegido.



Figura 24: Logo de Unity



Figura 25: Programa Unity

4.5.1. Conceptos básicos de Unity

A continuación se van a explicar algunos conceptos básicos de Unity para su mejor comprensión.

- Interfaz

La interfaz de Unity [30] cuenta con varias ventanas integradas que te permiten interactuar con diferentes aspectos de tu proyecto. Puedes organizar estas ventanas según tus preferencias y necesidades específicas. Las ventanas principales que Unity ofrece son:

- Ventana del proyecto: Aquí puedes ver y organizar todos los archivos de tu proyecto, como sprites, música o scripts. Es básicamente tu área de trabajo para gestionar los recursos.

- Ventana de la jerarquía: Muestra una lista de todos los objetos en tu escena. Se representa como una estructura de árbol que refleja la relación entre los diferentes elementos del juego.

- Inspector: Muestra detalles específicos sobre el objeto seleccionado en la jerarquía y es donde se pueden ajustar sus propiedades y componentes.



- Ventana de la escena: Proporciona una vista visual de la escena en la que estás trabajando. Es el lugar donde se diseña y construye el juego.

- Ventana de juego: Muestra una vista previa del juego en tiempo real. Se muestra cómo se verá y se comportará el juego mientras está en desarrollo. Es especialmente útil para probarlo y depurarlo.

- Escenas

Las escenas en Unity [31] son los diferentes escenarios del juego. Conceptualmente, una escena actúa como un lienzo virtual en el cual se disponen y organizan los diversos elementos de ese escenario. La implementación de escenas en Unity favorece una estructuración modular del proyecto, lo que facilita la carga y combinación selectiva de estas escenas para construir la experiencia global del juego.

- GameObject y sus componentes

Un GameObject [32] es la unidad básica que conforma Unity. Cada uno de ellos posee una serie de componentes que definen su comportamiento. Dichos comportamientos se pueden habilitar o deshabilitar desde el inspector, lo que proporciona una gran flexibilidad para controlar el comportamiento de un GameObject. Del mismo modo, también es posible habilitar o deshabilitar los GameObject del mismo modo.

- Tags

Un Tag [33] es una palabra que vincula a uno o más GameObjects. Las etiquetas están destinadas a identificar GameObjects para propósitos de scripting.

- Prefab

Un prefab [34] es un tipo de recurso que te permite almacenar un objeto

GameObject junto con todos sus componentes y propiedades. Este prefab funciona como una especie de plantilla a partir de la cual puedes crear nuevas instancias del objeto en la escena. La particularidad del prefab radica en que cualquier modificación realizada en el recurso del prefab se reflejará de inmediato en todas las instancias que se creen a partir de él. No obstante, también ofrece la opción de anular componentes y ajustes específicos para cada instancia individual, lo que proporciona un grado de personalización independiente de la plantilla original.

4.6. Gestión de versiones

Para alojar el proyecto se ha usado GitHub [35], un servicio basado en la nube que aloja un sistema de control de versiones (VCS) llamado Git [36]. Una forma fácil para crear diferentes ramas de trabajo y acceder al proyecto desde varios dispositivos.

También se ha contado con la ayuda de GitHub Desktop [37], una aplicación que crea un acceso más rápido y fácil a nuestro proyecto en Github , pudiendo gestionar desde ahí creaciones de ramas, subir un cambio o ver los cambios producidos entre otras muchas funcionalidades.



Figura 26: Logo de Git

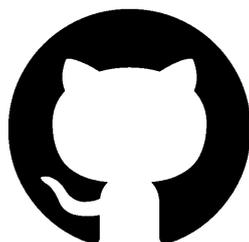


Figura 27: Logo de GitHub

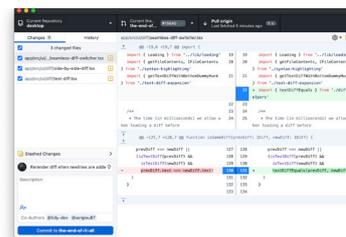


Figura 28: Programa GitHub Desktop

4.7. Gestión de autenticación de usuarios

Para la creación e ingreso de usuarios a la aplicación se han estudiado varios servicios:

Auth0 [38] es un servicio de gestión de identidad y acceso que se utiliza para la autenticación y autorización de usuarios en aplicaciones. Se pueden utilizar los user metadata en Auth0 para almacenar datos personalizados relacionados con un usuario autenticado. Al inicio la curva de aprendizaje puede ser pronunciada.

Firebase Authentication [39] es un servicio de autenticación proporcionado por Firebase, que facilita la autenticación de usuarios en aplicaciones web y móviles. Ofrece autenticación con múltiples proveedores, incluyendo correo electrónico, redes sociales y sistemas de terceros.

Debido a su sencillez y a su posible integración en el futuro con la base de datos de Firebase para agregar más funcionalidades, se ha escogido finalmente Firebase Authentication.



Firebase Authentication

Figura 29: Logo de Firebase Authentication

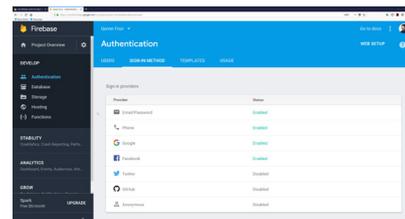


Figura 30: Consola de Firebase Authentication

4.8. Editor de código

A pesar de que el editor de código por defecto en Unity es Notepad, es posible cambiarlo fácilmente en las preferencias. Por ello, se van a estudiar varios editores de código para ver si hay alguna opción mejor.



Notepad++ [40], como opción predefinida, también la vamos a estudiar. Es un editor de código apreciado por su simplicidad y utilidad para tareas rápidas de edición de texto sin formato. Es útil para abrir y ver archivos de registro, escribir scripts simples, realizar anotaciones rápidas y realizar otras tareas simples de edición de texto.

Visual Studio Code [41] es muy popular en la comunidad de desarrollo de software debido a su capacidad de personalización, sus extensiones, su rendimiento y su versatilidad. Es una herramienta poderosa y altamente recomendada para la escritura y edición de código en una variedad de lenguajes de programación y para diferentes tipos de proyectos.

Eclipse [42] es especialmente popular entre los desarrolladores de Java. Es un entorno de desarrollo integrado de código abierto ampliamente utilizado para la creación, el desarrollo y la administración de aplicaciones de software.

Visual Studio [43] es otro entorno de desarrollo integrado creado por Microsoft que se utiliza para desarrollar aplicaciones de software en una variedad de plataformas, aplicaciones web, aplicaciones móviles, aplicaciones de escritorio y mucho más. Visual Studio es una de las suites de desarrollo más completas y ampliamente utilizadas en la industria de la programación y la ingeniería de software.

En este caso se descartó Eclipse, no solo por estar más enfocado a Java, sino por tener menos integración para Unity como otras opciones. Notepad++ también fue descartado por estar más destinado a tareas de edición de texto simples, que carece de muchas características necesarias para un desarrollo de software eficiente, como resaltado de sintaxis y depuración.

Entre Visual Studio y Visual Studio Code la decisión es más difícil, ya que ambos tienen buenas características, por lo que finalmente se decidió utilizar el Visual Studio Code, debido a su ligereza y capacidad de personalización.



Figura 31: Logo de Visual Studio Code

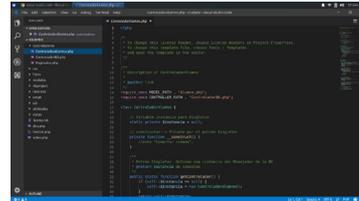


Figura 32: Programa Visual Studio Code

4.9. Lenguaje

Los principales lenguajes que admite Unity son:

C# [44] es el lenguaje de programación principal para el desarrollo de juegos en Unity. Es ampliamente utilizado y recomendado debido a su rendimiento, facilidad de uso y amplia compatibilidad con la plataforma.

JavaScript (UnityScript) [45] cada vez tiene menos soporte en las nuevas versiones de Unity, por lo que se ha desaconsejado su uso en favor de C#.

Boo [46] es otro lenguaje de programación que se puede utilizar con Unity, pero su uso es menos común en comparación con C#, por lo que sería mucho más complicado encontrar documentación o ejemplos basados en este lenguaje.

Por razones evidentes, se escogió C# como lenguaje para implementar este videojuego.

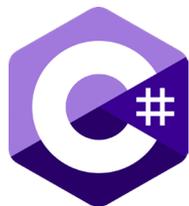


Figura 33: Logo de C#

```
Program.cs > ...
1 using System;
2
3 namespace HolaMundoApp
4 {
5     references
6     class Program
7     {
8         references
9         static void Main(string[] args)
10        {
11            Console.WriteLine("¡Hola Mundo!");
12        }
13    }
14 }
```

Figura 34: Ejemplo de código en C#



CAPÍTULO 5

Análisis de requisitos

Con el fin de establecer los requerimientos que debe reunir el software se va a utilizar el método de las historias de usuario.

Una historia de usuario [47] es una descripción general y no formal de una funcionalidad de software, redactada desde la perspectiva del usuario final o cliente. En lugar de detallar aspectos técnicos, se centra en las necesidades y expectativas del usuario, proporcionando una comprensión clara y directa de lo que se espera lograr con la funcionalidad en cuestión.

Este método encaja perfectamente en metodologías ágiles como Kanban, donde sus equipos incorporan las historias de usuario en su backlog y las ejecutan siguiendo su flujo de trabajo.

Las historias de usuario constan de breves descripciones en lenguaje natural que explican el resultado deseado sin entrar en detalles específicos. Suelen expresarse con una frase simple con la siguiente estructura:

“Como [perfil], [quiero] [para].”

De esta forma, se van a generar necesidades desde la perspectiva del cliente y, tras una fase de análisis y valoración para determinar su coste a nivel tiempo y prioridad, pasarán a formar unidades de trabajo para añadir al flujo de trabajo.

A lo largo del proyecto es posible cerrar, reabrir o descartar historias

de usuario, debido a imprevistos que puedan surgir.

A continuación se pueden observar las historias de usuario generadas y su colocación como unidades de trabajo en el backlog ordenadas descendentemente:

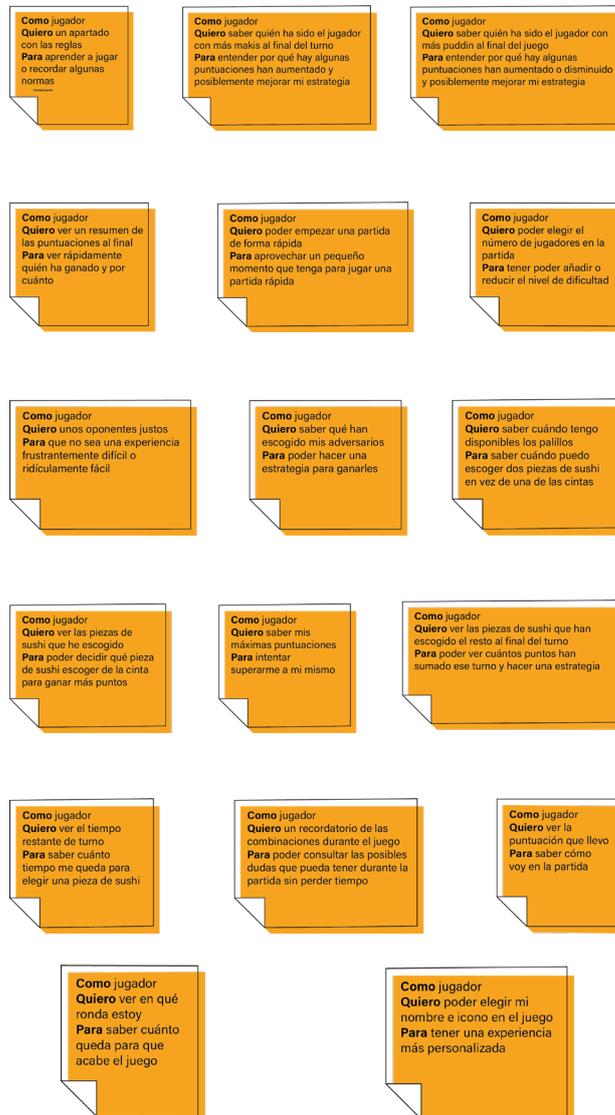


Figura 35: Historias de usuario

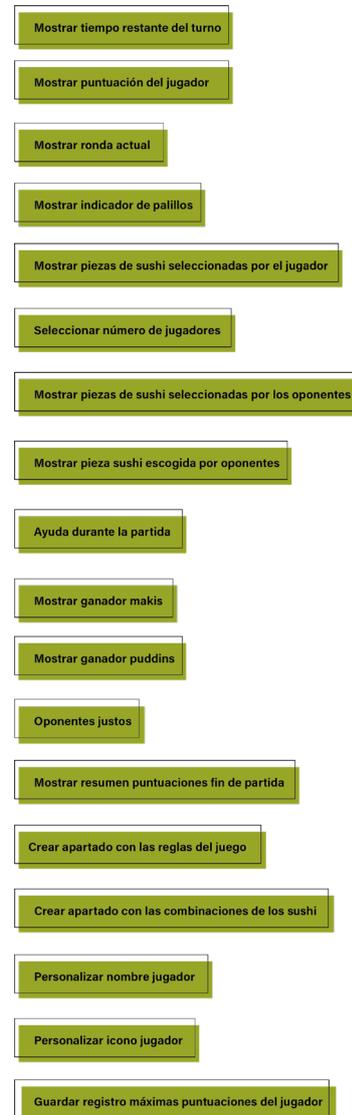


Figura 36: Backlog



Posteriormente se podrán ir añadiendo nuevas tareas y reorganizándolas en función de su importancia, gracias a la metodología ágil con la que se trabaja.

Primero se han colocado las unidades de trabajo que cumplen un equilibrio entre dedicación en tiempo e importancia para la jugabilidad del videojuego y al final las más triviales para la funcionalidad, cumpliendo los principios de la metodología Kanban de entrega temprana, desarrollo evolutivo y la mejora continua.

Cabe destacar que las dos últimas unidades de trabajo no han entrado en el producto final que aquí se presenta debido a falta de tiempo, por priorización de otras tareas nuevas que han ido surgiendo en el desarrollo.

5.1. Identificación y análisis de soluciones posibles

Una vez claras las unidades de trabajo y funcionalidades que se deberían implementar, se definieron algunos conceptos importantes para poder seguir con el desarrollo y diseño de la aplicación.

5.1.1. Enfoque

Lo primero que se tenía que decidir a la hora de empezar este desarrollo, es el enfoque que se quería tomar. Al ser una adaptación de un juego de cartas, se podría pensar que la solución más lógica sería trasladarlo al móvil también en este formato, como ya se ha hecho en otros juegos como el Uno! o el Solitario. Esta vez se decidió innovar en este aspecto, por lo que, tomando como referencia el tipo de restaurante en el que está basado el juego de cartas, este videojuego simulará estar en dicho restaurante y siendo el jugador un comensal queriendo seleccionar su comida.



Esto aportará una frescura nueva a la adaptación y una perspectiva nueva respecto al juego de cartas que adapta, lo que enriquecerá la experiencia para los jugadores habituales del juego.

5.1.2. Orientación del dispositivo móvil

Otra decisión a estudiar fue la orientación del móvil con la que se quería trabajar. Para los dispositivos móviles actuales es posible encontrar aplicaciones desarrolladas para ser reproducidas con el móvil en orientación vertical, otras en horizontal y otras donde se puede pasar de una orientación a otra sin problemas.

Para elegir la de este juego, se analizó el jugador al que va dirigido. Como se ha explicado en la introducción, los juegos casuales se dirigen a un público que no espera dedicar mucho tiempo al juego, no pueden o no desean invertir un gran compromiso y, normalmente, son utilizados en breves espacios de tiempo que tienen libre. Por lo tanto, lo razonable sería escoger una orientación cómoda y práctica, que no requiera grandes esfuerzos. Debido a esto, se escogió la orientación vertical, donde el usuario podría navegar por la aplicación con facilidad utilizando solo una mano, perdiendo el menor tiempo posible.

5.1.3. Puntuaciones

En la pantalla donde se desarrolla el juego, un elemento vital que debe mostrar la interfaz es la puntuación del jugador. Por ello, se estudió si era mejor mostrar la puntuación total del jugador en la partida o, únicamente mostrar en cada ronda los puntos que se habían conseguido en ella.

Como propuesta, era interesante mostrar la mayor cantidad de



información posible al usuario, pero realmente, la única puntuación que importaba era la total. Mostrar los puntos conseguidos en una ronda podría resultar curioso pero, finalmente, solo añadiría ruido a la interfaz y no aportaría nada de valor, así que únicamente se mostraría la puntuación acumulada durante toda la partida.

5.2. Solución propuesta

El sistema de la partida sería lo más importante, siendo el centro del juego, pero también hay que tener en cuenta algunos elementos para hacer la aplicación más completa.

Es necesario asegurarse de que los jugadores puedan ver su plato y el de los oponentes, de una manera fácil e intuitiva. Al final de cada turno, los jugadores deben poder ver qué pieza ha escogido cada jugador, como pasaría en el juego de cartas. Para completar la interfaz, el usuario debería poder ver cuánto tiempo le queda a ese turno, un recordatorio de la ronda en la que se encuentra y su puntuación.

Las puntuaciones deben ofrecer información clara al jugador acerca de aquellas cartas cuyo computo se realice de forma diferente, cómo es el caso de los maki y los puddin que, en caso del primero suman puntos a los dos jugadores con más sushi de ese tipo y, en el segundo, suman puntos al jugador con más puddin y restan al participante con menos. En el caso del maki a final de ronda, mientras que es al final del juego cuando se cuentan los puddin. Esto es esencial para mantener a los jugadores comprometidos y proporcionarles una comprensión clara de su desempeño.

Sería necesario añadir una guía con las reglas, ya que eso haría más accesible el juego para todo el mundo y no solo para el público ya familiarizado con el juego de cartas previamente. Además, un recordatorio de las posibles combinaciones de piezas que se puede hacer para sumar



más puntos, tanto para buscar alguna combinación de la que se tengan dudas como para los nuevos usuarios que no las conozcan y quieran ojear todas las opciones.

Añadir un registro de usuario abre oportunidades para la personalización y el seguimiento de estadísticas en el futuro, aunque por el momento sería conveniente implementarlo para que aparezca el nombre del jugador en la partida.

5.3. Fases del desarrollo

Una vez ya se ha establecido la el enfoque que se le va a dar al juego, se han generado una serie de historias de usuario para poder crear un juego desde la perspectiva del usuario, para saber cuáles son sus necesidades y deseos. Estas historias de usuario se han analizado y, a partir de ellas, se han generado las unidades de trabajo que guiarán el desarrollo de la aplicación.

Tras esto, se han estudiado las escenas que sería necesario implementar para el proyecto, así como la orientación del dispositivo, lo cual afectará directamente en la forma en la que usuario interactúe con el dispositivo.

Teniendo en cuenta todo ello, se bocetarán las interfaces de las escenas, asegurándose de que poseen una organización adecuada para incluir todas las funcionalidades necesarias, aunque al principio no estén todas ellas implementadas.

El diseño de la lógica del juego primero se centrará en la escena donde se desarrollará la partida y, paulatinamente, se irán añadiendo funcionalidades hasta obtener un producto completo y consistente.

Durante su transcurso, se irán construyendo versiones del producto cuando se haya alcanzado un paquete considerable de funciones y se probarán para garantizar la calidad y abordar problemas a medida que surgen.



Para los dibujos de las piezas de sushi se van a utilizar los mismos que hay en las cartas, por lo que es necesario vectorizarlos uno a uno. Habrá que crear también una ambientación que recuerda a un restaurante japonés, crear los platos, la interfaz de usuario,... La creación de dicha ambientación contribuirá a la inmersión del jugador.

En el desarrollo de algunas ilustraciones se va emplear también una tableta gráfica, con la cual se dibujarán, se colorearán y se les dará sombreado para, posteriormente, animarlas en Adobe AfterEffects.

Los efectos de sonido y la música se aplicarán una vez la aplicación ya esté avanzada, en beneficio de entender qué sonidos son los apropiados en cada caso. Se recurrirá a recursos de internet que serán tratados posteriormente para adaptarse a las necesidades del juego.



CAPÍTULO 6

Diseño de la solución

Para este videojuego lo principal será crear una buena experiencia de juego durante la partida. Que mantenga los mismos componentes que lo hacen funcionar en el juego de cartas físico y hacen la experiencia divertida. Por lo tanto, es importante conseguir un aspecto atractivo y agradable.

6.1. Diseño gráfico



Figura 37: Paleta de colores

Definir una paleta de colores es muy importante para el diseño gráfico del videojuego. Esto ayuda a crear una identidad visual y crea un aspecto más homogéneo y de cohesión. Para ello, se han escogido colores que característicos del juego y se ha comprobado la legibilidad de los textos al escoger los colores de la tipografía y los fondos.

Se ha utilizado el patrón japonés con diversas variantes, como se muestra en la figura 38 en diferentes partes del juego para dotar a su interfaz de cierta cohesión visual, a la par que mantienen la temática japonesa que desprende el juego.



Figura 38: Patrón japonés

Lo más importante es que toda la aplicación tenga una coherencia, su interfaz guíe al usuario a través de las pantallas con facilidad y las partidas tengan una estructura consistente que no dé lugar a errores. Los jugadores casuales no deben verse frustrados ni confusos en la aplicación.

6.2. Diseño del sonido

La música es un elemento que ayudará a completar la temática japonesa del juego, ambientará tanto en los menús como en las pantallas del juego y ayudará a diferenciar las diferentes fases de este. Por ello, se contará con una música ambiental para las escenas de menú, otra música para la escena de los cocineros donde se anuncia la ronda que va a empezar, una para el transcurso de la partida y otra para cuando se muestren las puntuaciones.

Al igual que la música, los efectos de sonidos son esenciales en los videojuegos y en cualquier aplicación, ya que da al usuario feedback de sus actos. Debido a esto, todos los botones emiten un sonido al pulsarse, de la misma forma que reproducen una pequeña animación simulando haber sido pulsados, los sushi en la cinta, al seleccionarse, emiten otro sonido más agudo para diferenciarlo de un botón normal, queriendo simbolizar una recompensa. Por otro lado, al seleccionar los palillos, ya que se usan de manera diferente al resto de sushi al seleccionarlos, tienen también un sonido diferente.



De igual forma, al cambiar de escena para ver los platos del resto de jugadores se ha aplicado un efecto de desplazar y, al abrir el panel de ayuda de combos, un sonido parecido al de mover una hoja, queriendo emular que se está observando lo que sería la carta del restaurante en físico.

6.3. Diseño de la interfaz

La interfaz de las pantallas se ha diseñado teniendo en cuenta la información que debería incluir cada pantalla y la forma en la que se quiere que el usuario interactúe con ella.

En el diseño de interfaces para dispositivos móviles es esencial tener en cuenta la Ley del Pulgar [48], donde se explica que los elementos más usados deben colocarse en las zonas más accesibles, es decir, en la parte central o inferior de la pantalla.

De esta forma e intentando que la interfaz de usuario sea lo más cómoda e intuitiva posible, se realizarán bocetos de las pantallas principales que iba a necesitar la aplicación.

6.3.1. Menú principal

A todas las pantallas del juego se accederá desde un menú sencillo e intuitivo, pero también atractivo visualmente, para llamar la atención del jugador. Este menú será lo primero que se verá al abrir la aplicación. En la figura 39, se puede ver un esquema muy básico de las funciones de este.

En la versión final, se aumentará el tamaño del botón "Jugar" en comparación con los demás, con el propósito de destacarlo y guiar de manera intuitiva al usuario. De esta manera, cuando desee iniciar una

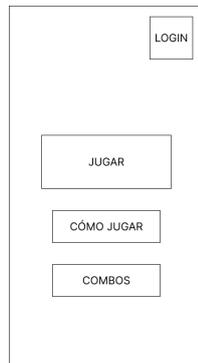


Figura 39: Boceto menú principal

partida rápida, le resultará más fácil encontrarlo de manera inconsciente.

En la parte estética, se decidió añadir una animación de todos los tipos de sushi existentes en el juego, pasando por el menú montados en un plato con ruedas. Esto le dará más dinamismo al menú principal, una mejor presentación al abrir el juego y ayudará al usuario a familiarizarse con los sushi disponibles.

6.3.2. Explicación de los combos

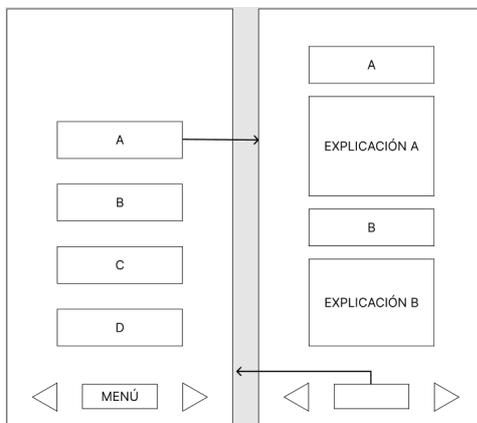


Figura 40: Boceto combos

Se hará un apartado para que los jugadores puedan consultar qué combinaciones hacer con los sushi para conseguir más puntos. Pensando tanto en los jugadores más experimentados como los nuevos, se dispondrá de dos pantallas.

En la primera se dará la opción de buscar entre un listado con todos los tipos de sushi. Cuando se pulse en uno de ellos, se llevará a la segunda pantalla, donde se encontrará la explicación del combo. Se colocarán dos flechas para navegar por el listado, así como un botón para volver al menú principal.

Como ya se ha comentado, en la segunda pantalla se encontrarán las explicaciones de los combos y será accesible al pulsar un tipo de sushi en la primera pantalla. Una vez en esta pantalla, también se puede navegar entre las explicaciones sin volver a la pantalla del listado. De todas formas,



también contará con un botón para volver a la pantalla de listado.

En lo que respecta a la parte estética, se ha decidido dar un aspecto similar a una carta de restaurante, debido al listado que contiene.

6.3.3. Explicación del juego

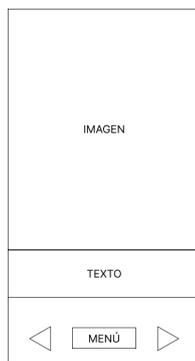


Figura 41: Boceto explicación del juego

Por otro lado, también se creará un apartado para que los jugadores nuevos puedan aprender a jugar. Para ello, se dispondrá una pantalla con flechas para ir pasando de regla en regla. Encima de cada una de ellas, un dibujo acompañará el texto.

Aparte de las flechas de navegación, también se encontrará un botón para volver al menú principal.

6.3.4. Juego

Esta pantalla resulta la más compleja, ya que en ella reside la función más importante de la aplicación y debe dar cabida a todas las necesidades del jugador. Es por ello, y por la gran cantidad de funcionalidades que ha tenido que incluir, que ha sido la más complicada de diseñar.

La posición de la cuenta atrás, el marcador de ronda y las puntuación estaba clara, se colocarían arriba del todo de la pantalla. De esta forma se dejaría libre el resto de espacio en pantalla, mucho más accesible para la interacción del usuario.

El espacio central de la pantalla estaría reservado para el sushi, las piezas centrales del juego, lo que le daría más importancia y protagonismo.

Se hizo un estudio para analizar cuál era la mejor manera de mostrar los platos del jugador y los oponentes, ya que toda esa información ocuparía mucho espacio en pantalla y, junto a todo lo demás, podría abrumar al jugador.

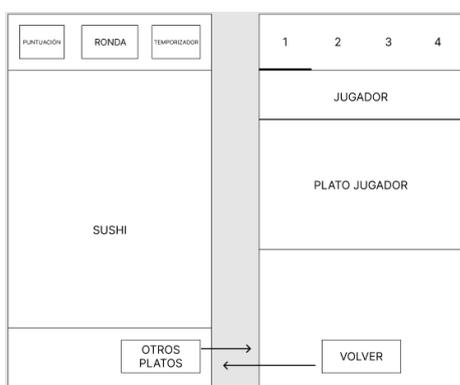


Figura 42: Propuesta n°1 - pantalla de juego

Una idea fue no mostrar ningún plato en la pantalla, teniendo acceso a una segunda pantalla mediante un botón en la que se podrían ver todos los platos, incluido el del jugador. Esta propuesta fue descartada debido a que resultaría tedioso al usuario tener que moverse de pantalla cada vez que quisiera tener controlado lo que había escogido.

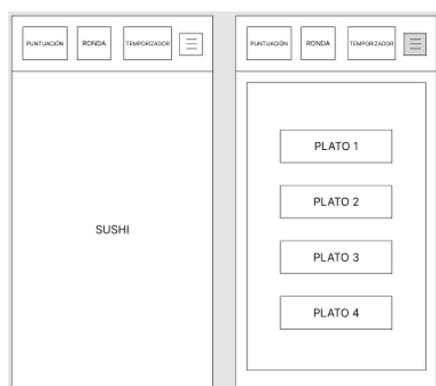


Figura 43: Propuesta n°2 - pantalla de juego

Otra idea similar, proponía que, en vez de acceder a una segunda pantalla mediante un botón, los platos apareciesen en un panel flotante mientras se mantenía pulsado un botón. Esta propuesta también fue descartada debido a que añadiría demasiada información en la pantalla cada vez que apareciese el botón flotante, lo que podría acabar resultando abrumador.

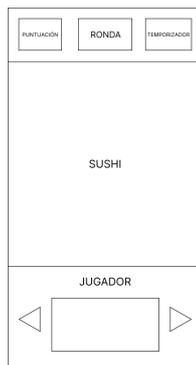


Figura 44: Propuesta n°3 - pantalla de juego

Una propuesta también descartada fue la opción de navegar entre platos bajo la sección de la pantalla donde aparecerían el sushi. Esta propuesta no convencía debido a que resultaría confuso identificar cuál era el plato del jugador y podría resultar en equivocaciones.

Una alternativa que se implementó de forma experimental consistía en añadir unos círculos entre el plato del jugador y el sushi. Cada uno de estos círculos iría identificado como cada uno de los oponentes y, si se mantenía pulsado, aparecería el plato de dicho jugador. Pese a parecer una idea ingeniosa en un primer momento, finalmente se cambió, porque durante el juego era preferible ver de un vistazo todos los platos juntos.

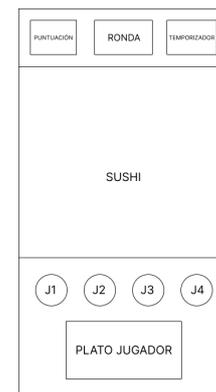
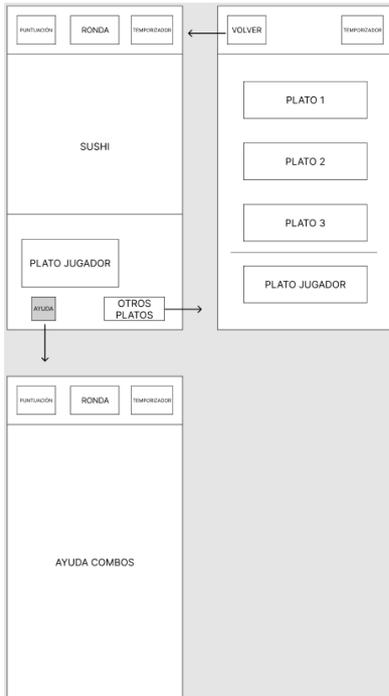


Figura 45: Propuesta n°4 - pantalla de juego

Finalmente, la propuesta ganadora fue la que mostraba siempre el plato del jugador bajo el sushi y, bajo este dos botones.

En el botón de la derecha se mostraría en otra pantalla una vista de todos los platos, incluido el del jugador, para poder compararlo con el resto, y el temporizador, en el mismo sitio que la pantalla anterior, para poder tener en cuenta el tiempo y no perder el turno. Esta pantalla también contaría con un botón para volver a la pantalla principal del juego. Debido a la importancia que se le quería dar a los platos, este botón quedaría alojado arriba a la izquierda, lo que, según la ley del pulgar vista anteriormente, era un lugar de difícil acceso si se decidía jugar solo con una mano. Para hacer más fácil



este cambio, se habilitó también el gesto de deslizar la pantalla hacia los lados, dejando así al jugador elegir como prefería moverse entre ambas pantallas.

En la zona inferior izquierda de la pantalla principal de juego, se podía habilitar un botón que, al mantenerlo pulsado, abriría un breve resumen de las combinaciones de sushi disponibles en el juego.

Figura 46: Boceto pantalla de juego

6.3.5. Puntuaciones

En esta pantalla se mostrarían las puntuaciones de todos los jugadores, para así poder valorar su desarrollo en la partida. Se buscaba una pantalla simple, con los datos necesarios esquematizados, sin demasiado ruido. Por ello, se decidió mostrar en el centro de la pantalla a todos los usuarios junto a su puntuación. Estos estarían en el mismo orden en el que se habían colocado anteriormente en la pantalla

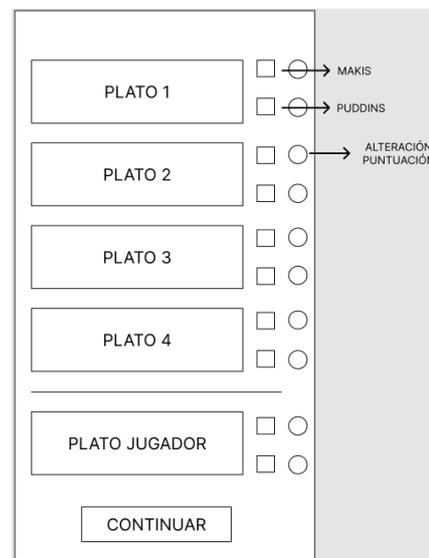


Figura 47: Boceto puntuaciones



para visualizar los platos, lo que le daría coherencia a la interfaz.

Al lado de cada jugador, se podría mostrar el número de makis y puddins, los únicos dos tipos de sushi que modificaban la puntuación fuera de la ronda. Al lado de este, se podría poner un indicador que mostraría el ganador o perdedor en cada uno de los jugadores que fuese necesario.

Abajo del todo se encontraría el botón que permitiría reanudar el desarrollo del juego.

6.3.6. Selección de jugadores

Esta pantalla sería el único trámite que habría que atravesar antes de empezar una partida, por lo que se intentaría que fuera una interfaz lo más sencilla y accesible posible.

Por ello se decidió que los botones para seleccionar el número de jugadores abarcarían todo el ancho de la pantalla y a una altura a la que fuera cómodo acceder con el pulgar.

NÚMERO JUGADORES
3
4
5

Figura 48: Boceto
seleccionar número jugadores

6.3.7. Registro de usuarios

El acceso a este registro de usuario estaría en el menú principal. De él se accedería a la pantalla de iniciar sesión, con la opción de registrarse si aún no tiene cuenta. Sería conveniente cubrir el caso de haber olvidado la contraseña de la cuenta, por lo que esta opción se colocaría debajo del campo para introducirla. Finalmente, al haber iniciado sesión, se accederá a la pantalla de perfil donde, de momento solo estará el nombre de usuario y

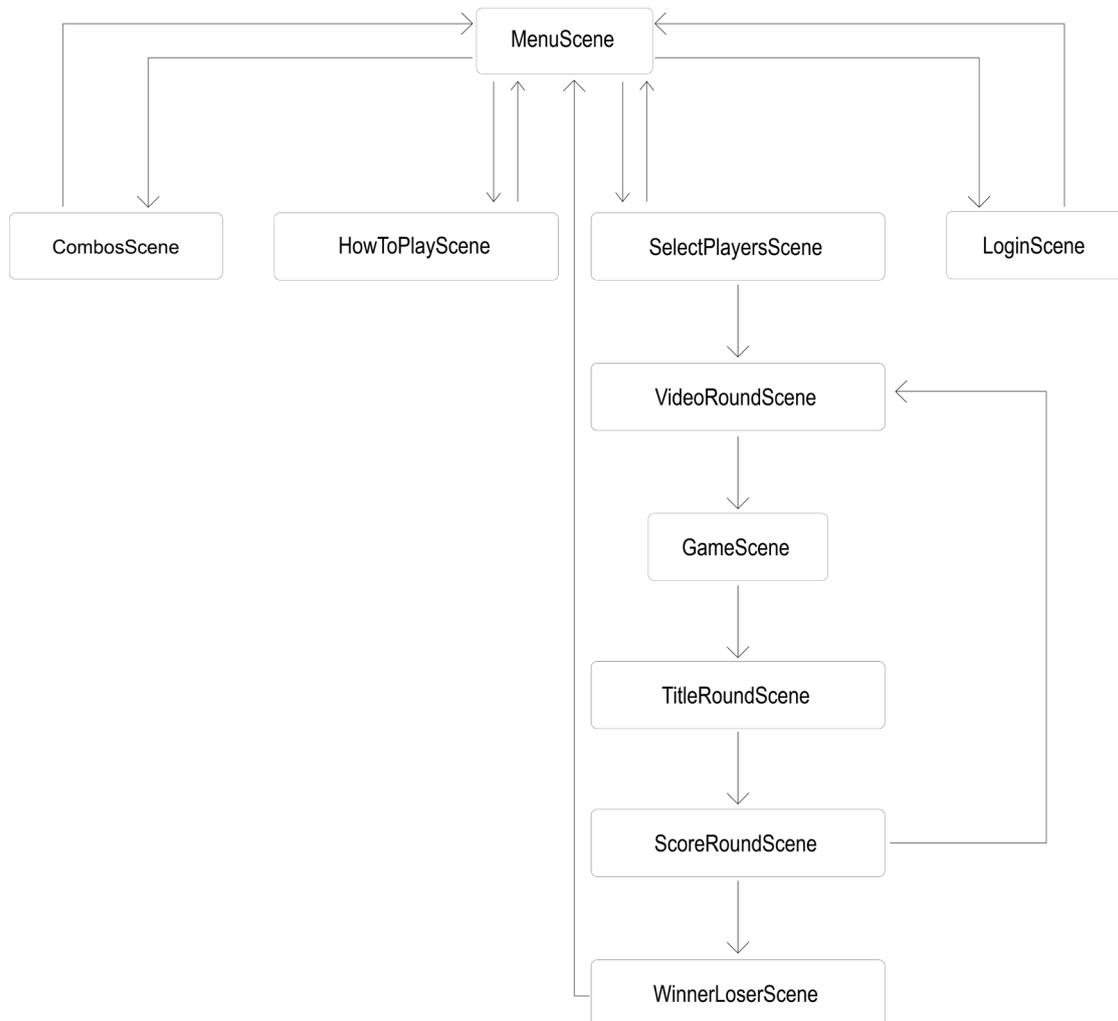


Figura 50: Diagrama de flujo de escenas

Como se puede observar, desde MenuScene, tras acceder a SelectPlayersScene, sería donde comenzaría una partida. Para tener claro el progreso de una partida completa, se ha estudiado en otro diagrama de flujo más detallado.

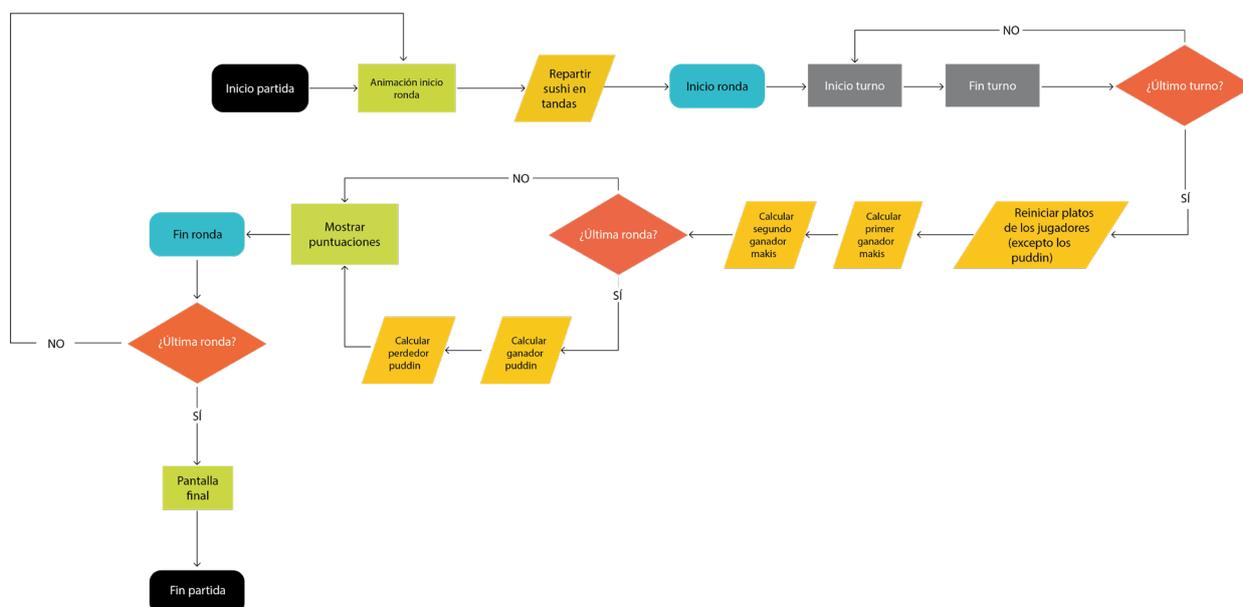


Figura 51: Diagrama de flujo de una partida

Algunas de estas acciones se han separado en diversas escenas para tener un mejor control sobre ellas.

- La animación inicio ronda, indicada en el diagrama de flujo, corresponderá a la escena VideoRoundScene.
- Todas las acciones entre repartir los sushi en tandas hasta mostrar las puntuaciones, se gestionarán en la escena GameScene, es decir, los turnos dentro de cada ronda.
- Previo a la muestra de puntuaciones, se accederá a la escena TitleRoundScene, una escena sencilla donde se mostrará el número de la ronda de la cual se van a ver las puntuaciones.
- Las puntuaciones se mostrarán en ScoreRoundScene.
- Finalmente, la pantalla final, corresponderá a la



escena WinnerLoserScene

Tras tener estos conceptos claros, se procederá a explicar detalladamente cada escena por separado.

6.4.1. MenuScene

MenuScene es la primera escena que se encuentra al abrir la aplicación.

- Interfaz

En esta escena se encuentran dos botones que permiten al jugador ir a la sección que explica las reglas del juego y a la sección con el listado de tipos de piezas de sushi. También hay un botón más grande, encima de los demás, que es el encargado de iniciar una partida.

Arriba a la derecha se encuentra el icono de usuario que lleva al jugador a la pantalla de iniciar sesión y, por último, abajo de los botones, se encuentra, reproduciéndose en bucle, una animación que presenta al jugador los diferentes tipos de sushi.



Figura 52: Interfaz MenuScene

- Scripts

- PlayVideoMenu: Se trata de un script que controla al componente VideoPlayer. En él se lleva a cabo la



reproducción del vídeo de la animación que se encuentra debajo de los botones.

- ButtonManager: Este script se encuentra a lo largo de toda la aplicación. Se encarga de gestionar las funciones básicas de algunos botones, principalmente de navegación entre escenas. En esta escena en concreto gestiona la navegación hasta las escenas ComboScene, HowToPlayScene, LoginScene y SelectPlayersScene.

- User: En este script se ha implementado un Singleton, con el fin de que guarde el valor del usuario registrado, en caso de haber alguno.

- UserNameMenu: Aquí es donde se gestiona el texto que aparece al lado del icono de usuario. Comprueba si hay algún usuario registrado, accediendo al script User. En caso afirmativo, muestra el nombre del usuario y, en caso contrario, muestra el texto "Iniciar Sesión".

- ResetPlay: A la escena MenuScene se puede volver a llegar desde diferentes escenas. Una de ellas es desde la pantalla WinnerLoserScene, lo que quiere decir que ya se ha jugado una partida. Este script se encarga de reiniciar todos los valores de la partida.

- MusicMenu: El patrón Singleton se ha aplicado también en esta entidad lógica. Se encarga de la gestión de un AudioSource y dispone de música ambiental a las escenas MenuScene, HowToPlayScene, CombosScene, LoginScene, SelectPlayersScene y WinnerLoserScene.

6.4.2. CombosScene

En esta escena es donde va el jugador con el objetivo de ver todos los tipos de piezas de sushi que hay y la forma que tiene de combinarlas para conseguir más puntos. Está diseñada para permitir que el usuario pueda,

tanto buscar un tipo de sushi en concreto para consultar una duda ocasional, como para permitirle navegar entre todos para que aprenda a jugar.

- Interfaz

Esta escena cuenta con dos pantalla.

- MenuCombo: Al abrir esta escena se encuentra una pantalla que simula la carta de un restaurante. En ella se puede buscar el tipo de sushi deseado entre las dos páginas que hay a modo de índice. Abajo se encuentran unas flechas de navegación y un botón para volver a MenuScene.



Figura 53: Interfaz ComboScene - menuCombo

- Combo: Al pulsar sobre cualquier pieza de sushi, la pantalla cambia y muestra el detalle del sushi que se haya seleccionado. Esta nueva pantalla también cuenta con unas flechas de navegación que permiten navegar entre las explicaciones de todas las otras piezas y un botón para volver a la pantalla de selección del tipo de sushi.



Figura 54: Interfaz ComboScene - combo

- Scripts

- CombosManager: Se encarga de la gestión del cambio de pantallas para cambiar entre unos grupos de tipo de sushi y otros en la parte del buscador de tipos, así como de mostrar la pantalla de explicación o cambiar de esa explicación a otra.



- **NavigationMenuCombo**: Este script es un componente de cada una de las flechas encargadas de la navegación en la pantalla del buscador. A través del inspector se le da valor a la variable *menuComboPage* que indica a qué página del buscador se desea cambiar. Para ello se llama al método *changeMenuCombo*, perteneciente al script *CombosManager*.

- **NavigationCombos**: Similar al anterior, este script es un componente de cada una de las flechas encargadas de la navegación en la pantalla de explicación de los tipos de sushi. A través del inspector se le da valor a la variable *comboPage* que indica a qué página de las explicaciones se desea cambiar. Para ello se llama al método *changeCombo*, perteneciente al script *CombosManager*.

También es componente de cada uno de los tipos de sushi del menú que deben ser redirigidos a su explicación.

- **ButtonManager**: Se encarga de volver de la pantalla Combo a MenuCombo, así como de volver de esta última a MenuScene.

6.4.3. HowToPlayScene

Esta escena es la encargada de enseñar al jugador las reglas del juego.

- Interfaz

En esta se encuentran dos grandes flechas abajo del todo que permiten al jugador avanzar o retroceder entre las reglas del juego. Entre ellas, se encuentra un botón para volver a la MenuScene.



Figura 55: Interfaz HowToPlayScene



Encima de esta barra de navegación se encuentra un dibujo y, más abajo, un texto. Estos dos irán cambiando según la navegación con las flechas para ir enseñando al usuario cómo se juega.

- Scripts

- HowToPlayManager: En este script se crean y se ordenan cada una de las fases a las que se navega con las flechas. Permite mostrar fases y llevar el control de que se muestren siempre en el orden correcto.

- ButtonManager: Se encarga volver a MenuScene mediante el botón que se encuentra entre las flechas de navegación.

6.4.4. LoginScene

En esta escena el jugador puede iniciar sesión para que aparezca su nombre en la partida. En caso de no tener cuenta puede registrarse para crear una nueva. Si ha olvidado la contraseña, con solo introducir su email, se le enviará un correo para crear una nueva contraseña. Una vez el jugador haya iniciado sesión, será redirigido a la pantalla de perfil, donde podrá cerrar la sesión cuando quiera.

- Interfaz

Esta escena cuenta con varias pantallas.

- Inicio de sesión: En esta pantalla se encuentra dos InputField donde se deberán introducir el email y la contraseña vinculados a la cuenta. Debajo del InputField de



Figura 56: Interfaz LoginScene - inicio de sesión



la contraseña se encuentra el texto "¿Has olvidado la contraseña?", que redirigirá la usuario a otra pantalla para la gestión de ese problema. Tras esto, se encuentra el botón de Iniciar Sesión.

Tanto en caso de éxito como de fallo, aparecerá un aviso dándole la información pertinente al usuario.

Por último, abajo se da la opción de crear una nueva cuenta, lo que redirige a la pantalla de registro.



Figura 57: Interfaz LoginScene - registro

- Registro: En esta pantalla se encuentra cuatro InputField con el fin de introducir nombre, email, contraseña y la repetición de la contraseña. Tras esto, se encuentra el botón para ejecutar el registro.

Tanto en caso de éxito como de fallo, aparecerá un aviso dándole la información pertinente al usuario.

Por último, se le da la oportunidad al usuario de volver a la pantalla de inicio de sesión.

- Recuperar contraseña:
En esta pantalla se encuentra un único InputField, para introducir el email al que se enviará un correo para crear una contraseña nueva. Tras esto, se encuentran dos botones, uno rojo para volver a la pantalla de inicio de sesión con el texto "Volver" y otro verde para ejecutar el envío del correo.

Tanto en caso de éxito como de fallo, aparecerá un aviso dándole la



Figura 58: Interfaz LoginScene - recuperar contraseña



información pertinente al usuario.

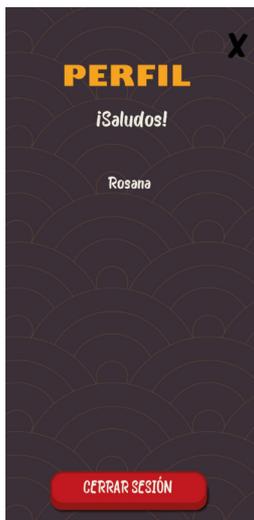


Figura 59: Interfaz LoginScene - perfil

- Perfil: En esta pantalla se encuentra el nombre de usuario, para verificar en qué cuenta se ha registrado.

Al final de la pantalla, se puede acceder al botón de cerrar sesión.

Todas estas pantallas tienen la posibilidad de ser cerradas para acceder directamente a la MenuScene desde el botón del que disponen arriba a la derecha.

- Scripts

- FirebaseController: En este script se encuentra toda la lógica que, junto a Firebase Authentication, se encarga de gestionar el registro e inicio de sesión de usuarios.

Todos los InputFile anteriormente mencionados, se le asignan como valor por medio del Inspector y, desde ahí son leídos sus datos. Con el método IsNullOrEmpty de la clase string, se comprobará que no haya ningún campo necesario vacío. No será necesario comprobar el formato de cada input, ya que los propios InputField tienen la capacidad de restringir el formato.

- ButtonManager: Le da función al botón que se encuentra arriba a la derecha para volver a MenuScene.



6.4.5. SelectPlayersScene

A esta escena es a la que se accede una vez el jugador pulse sobre el botón "Jugar" de MenuScene. En ella, el usuario podrá elegir el número de jugadores en la partida: tres, cuatro o cinco.

- Interfaz

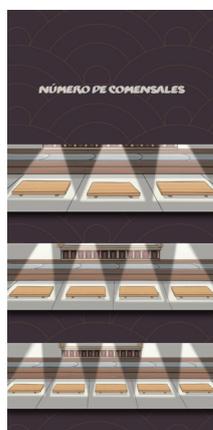


Figura 60: Interfaz
SelectPlayersScene

Arriba del todo se encuentra el texto "Número de comensales" haciendo un juego de palabras con la ambientación del juego.

Abajo se encuentran los tres botones disponibles para elegir el número de jugadores, mostrando los platos correspondientes en cada una de las elecciones.

- Scripts

- NumberPlayers: En este script se ha implementado el patrón Singleton, con el fin de que guarde el valor del número de jugadores.

- PickPlayers: Este script esta asociado como componente a cada uno de los botones para elegir el número de jugadores y, en cada uno se asigna el valor de la variable players según el botón que sea. Una vez es pulsado, este valor se pasa al script NumberPlayers.

6.4.6. VideoRoundScene

Esta escena se reproducirá siempre que se inicie una nueva ronda.

- Interfaz



Figura 61: Interfaz VideoRoundScene

En esta pantalla se reproduce una animación donde se muestran dos cocineros en el restaurante en el que está ambientado el juego. También se pueden observar las cintas por las que irán las piezas de sushi y el plato del jugador. Al final de la animación la cámara se acerca y presentará la ronda que se va a iniciar, primero en japonés y después en español.

Esta escena contiene un componente AudioSource que reproduce automáticamente, al inicio, una música durante la animación.

- Scripts

Una vez se accede a esta escena, el script MusicMenu, del que se ha hablado en el apartado orientado a MenuScene, detiene la música ambiental que estaba sonando en el juego.

- PlayVideoRound: Script que se encarga de gestionar el comportamiento del componente VideoPlayer, el cual debe reproducir el video de la animación.

6.4.7. GameScene

En esta escena es donde transcurrirán los turnos de la partida. Al igual que se ha estudiado antes el diagrama de flujo de una partida, en esta sección se va a profundizar un poco más y se va a analizar con detalle el diagrama de flujo de un turno.

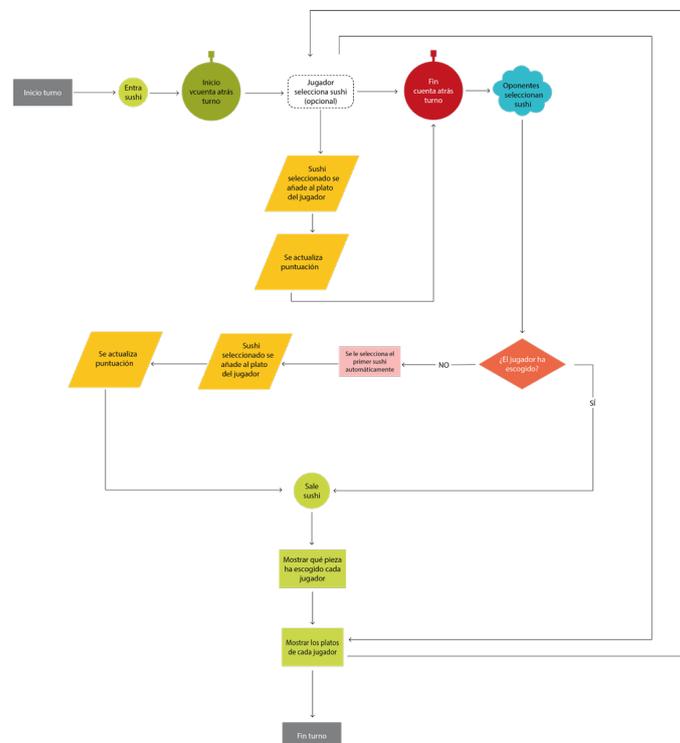


Figura 62: Diagrama de flujo de una partida

Esta escena se ha decidido dividir en dos pantallas, una que representará las cintas y el plato del jugador, donde transcurrirán las acciones desde que entra el sushi hasta que sale, y otra donde se mostrarán las piezas elegidas por cada jugador y sus platos. Se podrá acceder de una a otra durante el turno.

- Interfaz

- En la pantalla donde se encontrarán las cintas, aparte de estas, también se puede encontrar el plato donde se mostrarán las piezas de sushi del jugador. Bajo este, hay dos botones, uno a la izquierda que, si se mantiene pulsado, abre un panel para ver un resumen de las combinaciones que se pueden hacer de sushi y, el otro que llevará al usuario a la pantalla con los platos de todos los jugadores.

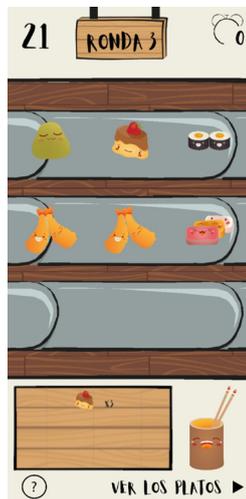


Figura 63: Interfaz GameScene - partida

En la parte de arriba se encuentran, de izquierda a derecha, el contador de la puntuación actual del jugador, un recordatorio de la ronda en la que se encuentra y la cuenta atrás.

- En la otra pantalla, se podrán ver los platos de todos los jugadores, la lado de su nombre y puntuación. El usuario encontrará su plato abajo del todo, diferenciado del resto con un separador.

En la parte de arriba se encuentra el botón que permite volver a la pantalla de juego, donde se encontrarán las cintas con los sushi, y la cuenta atrás, para no perder de vista cuánto

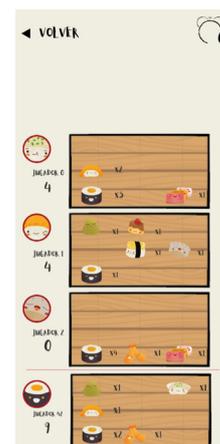


Figura 64: Interfaz GameScene - platos

tiempo queda antes de que se acabe el turno.



Figura 65: Interfaz GameScene -
sushi seleccionado

También se mostrarán en esta pantalla las piezas de sushi seleccionadas en ese turno por cada uno de los jugadores. Aparecerá la pieza elegida en grande en el plato correspondiente junto con la puntuación que se ha sumado por ese movimiento.

- Scripts

Al ser esta escena donde transcurrirá la partida, se ha modelado un diagrama de clases UML con las clases más importantes para tener más claro el diseño visual de la estructura.

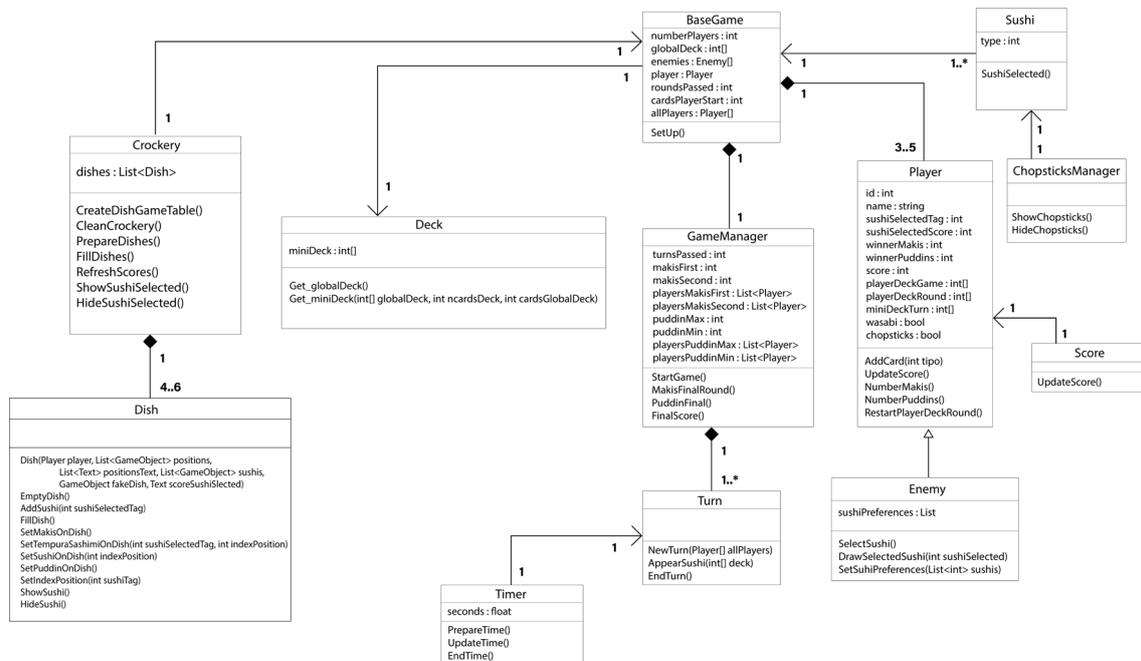


Figura 66: Diagrama de classes UML

- Deck: Su nombre significa "baraja" en inglés. Dado que este videojuego está basado en un juego de cartas, se ha decidido mantener esta nomenclatura para hacer la adaptación más entendible. En él se crea la baraja de cartas del juego real, manteniendo la proporcionalidad de los tipos de sushi, estudiado para hacer el juego más justo, esta baraja se llamará *globalDeck*. Por otro lado, en el juego de cartas físico, al inicio de cada ronda se reparten un número de cartas a cada jugador (dependiendo del número de jugadores). Estas pequeñas barajas que se forman, se han llamado *miniDeck* y también se crearán en la clase *Deck*. Estas *miniDeck* serán las que irán rotando de jugador en jugador hasta que se quede sin cartas, lo que



significará el final de la ronda.

- Turn: Será el encargado de gestionar los turnos de cada ronda. Al inicio de cada turno se deben rotar las *miniDeck* de un jugador a otro y será esta la clase que se encargue de ello. Por otro lado también tendrá que gestionar la colocación de los sushi en la cuadrícula según el *miniDeck* correspondiente, la aparición de estos en pantalla, lo que provocará el inicio de la cuenta atrás del temporizador, y su salida.

- Timer: Llevará el control del temporizador de los turnos. Se encargará de hacer la cuenta atrás para que se acabe el turno y los sushi salgan de la pantalla. Se preparará el temporizador con los segundos elegidos para la duración del turno, esperando la señal de la clase *Turn* para iniciar la cuenta atrás y, al llegar a 0 comprobará si el jugador ha llegado a seleccionar alguna pieza de sushi y, en caso negativo, le hará coger la primera pieza de la cuadrícula, ordenará robar a las IAs y cambiará a otra escena donde se mostrarán los sushi elegidos por todos los jugadores y sus respectivos platos.

- GameManager: Se encargará de la gestión de la progresión de la partida. Controlará los inicios de cada turno y de que no se sobrepasen las tres rondas de juego. Al final de cada ronda se debe calcular qué jugadores han sido los que han logrado obtener el mayor y el segundo mayor número de makis. Sin embargo, al final de la última ronda, también se debe averiguar qué jugador ha sido el que ha obtenido más pudding y el que ha obtenido menos. De todo ello, al igual que de aplicar la correspondiente recompensa o penalización, también se encargará esta clase.

- Player: En ella se encuentran todos los atributos que debe tener un jugador. En ella se tendrá controlada la puntuación, si tiene algún wasabi que aún no ha sido combinado con un nigiri, si tiene palillos sin usar, si ha resultado ganador, segundo o perdedor de makis y pudding, cuál ha sido la última pieza que ha escogido, ... Aparte de todo esto también se deberá llevar un registro de todas las piezas que ha obtenido a lo largo de la partida, las



que ha obtenido en esta ronda y, obviamente de la *miniDeck* que tiene ese turno y de la cual deberá elegir una pieza. En esta clase también se puede encontrar el método que, pasándole como parámetro un tipo de sushi, lo añade a la selección del jugador, los métodos que devuelven el número de makis que tiene y el de puddins, entre otros.

- **Enemy:** Hereda de *Player* y tiene como atributo diferenciador una lista de preferencias de sushi. Este atributo será lo que guiará el comportamiento de los a la hora de seleccionar las piezas de sushi de las cintas. Los métodos que contiene son los necesarios para crear los criterios de selección de piezas de sushi, seleccionar el tipo de sushi siguiendo dichos criterios o preferencias y finalmente, cogerlo de las cintas.

- **Sushi:** Se encontrará como componente en cada una de las piezas de sushi que aparezcan delante del jugador. Como único atributo tendrán el tipo del sushi al que acompañan. También se encargará de detectar cuando el sushi al que pertenecen ha sido seleccionado por el jugador y se encarga de añadirse al plato, con todo lo que ello requiere. También gestionará, si ha elegido palillos, a partir de ese turno, dejarle seleccionar dos piezas en vez de una.

- **BaseGame:** Será la encargada de guardar los valores de todas las variables globales. El estado de la *globalDeck*, los datos de todos los jugadores o incluso el número de rondas que se han jugado, son algunos de los atributos que contiene. También se encargará, al principio del juego, de preparar la partida antes de que empiece, estableciendo por ejemplo cuántas piezas de sushi deben aparecer al principio de la ronda a cada jugador, según cuántos jugadores sean. La implementación del patrón Singleton asegura que solo haya una instancia activa de *BaseGame* en la ejecución del juego.

- **Dish:** Representa cada uno de los platos en el juego. Se encarga de la representación y manipulación de los platos, permitiendo mostrar tanto los tipos de sushi y su cantidad, como quitarlos todos para una nueva ronda.



- Crockery: De la palabra inglesa "vajilla". Se encarga de la creación, limpieza y preparación de los platos, así como de mostrar los sushi seleccionados en ese turno.

- ChopsticksManager: Controla el comportamiento de los palillos en la escena.

- Score: Se encarga de la actualización del contador de puntuación del jugador.

- RestauranteMusicManager: Responsable de la gestión de dos AudioSource, uno para proporcionar música ambiental al restaurante, y otro para la realización de los efectos de sonido correspondientes.

- ButtonManager: Le da la función a los botones que van de una pantalla a otra y al botón que muestra la carta con el resumen de las combinaciones.

6.4.8. TitleRoundScene

Esta escena sirve como presentación de la escena que aparecerá después de ella. También recuerda las puntuaciones de qué ronda se van a ver.

- Interfaz

Esta pantalla es muy sencilla, ya que solo contiene un texto donde "Puntuaciones" seguido de la ronda que corresponda



PUNTUACIONES

RONDA 2

Figura 67: Interfaz
TitleRoundScene



- Scripts

- TitleRound: Se encarga de que aparezca la ronda correcta en el título y tras unos segundos, pasa a la siguiente escena.

6.4.9. ScoreRoundScene

Muestra las puntuaciones de cada jugador. Se detalla cuántos makis y puddins tiene cada uno además de indicar quiénes han sufrido cambios en su puntuación por tener más o menos de cada.

- Interfaz

Se muestran todos los jugadores, en el mismo orden en el que se podían ver sus platos en la GameScene, junto a su puntuación. A la derecha de cada uno hay un contador de cuántos makis y cuántos puddin tiene cada uno. Los jugadores que más makis tengan tendrán un "+6" ó "+3" al lado de dicho contador, mientras que el ganador y el perdedor de los puddin tendrán un "+6" y un "-6" al lado del otro contador. Obviamente estos últimos solo se mostrarán en caso de que sean las puntuaciones de la última ronda.



Figura 68: Interfaz ScoreRoundScene

Al final de la pantalla se encuentra el botón para continuar a la siguiente escena.



- Scripts

- ScoreRound: Se encarga de mostrar los jugadores correspondientes con toda su información.

- ButtonManager: Pulsando el botón se cambiará a la escena VideoRoundScene o WinnerLoserScene, según sea la última ronda o no.

6.4.10. WinnerLoserScene

Última escena de la partida. Muestra un resumen de las puntuaciones finales de todos los jugadores.

- Interfaz

Se muestra arriba en grande el usuario y más abajo sus oponentes. Aparece un texto que anuncia si el jugador ha ganado perdido o empatado y, seguidamente, se muestran las puntuaciones de todos los jugadores, apareciendo en dorado el jugador ganador.

Al final de la pantalla se encuentra el botón para volver a la MenuScene.

- Scripts

- FinalSceneManager: Se encarga de la presentación de la escena final del juego, destacando al ganador o indicando un empate, y mostrando la información relevante sobre cada jugador.



Figura 69: Interfaz WinnerLoserScene



- MusicMenu: Este script vuelve a encargarse de la música ambiental del juego a partir de esta escena.

- ButtonManager: Le da la función al botón que se encuentra abajo de todo para volver a la MenuScene.



CAPÍTULO 7

Incrementos del software

Los incrementos [49] son una estrategia que se centra en la entrega continua de valor a través de mejoras graduales, permitiendo adaptabilidad, retroalimentación continua y reducción de riesgos.

En el proceso de desarrollo se han construido diversos incrementos para comprobar que el videojuego cumplía con las expectativas. Se necesitaba comprobar que la interfaz era cómoda de utilizar, el usuario no tenía problemas para entenderla y no tenía problemas al mostrarse en un móvil. Por otro lado, también había que comprobar que la funcionalidad era correcta y sopesar qué funciones faltaban para dar al usuario una experiencia completa.

7.1. Primer incremento

El primer incremento dio como resultado una aplicación muy sencilla, únicamente para comprobar que las mecánicas básicas funcionaban.

- El mecanismo de instanciación de las piezas fuera de la pantalla y su desplazamiento hasta la mitad de la pantalla.
- Se detectaba con éxito cuando un jugador pulsaba una pieza y esta, como resultado desaparecía y sumaba su puntuación al contador.
- Se implementó la lógica del temporizador y, cuando llegaba a cero la cuenta atrás, las piezas de sushi se desplazaban hacia fuera de la

pantalla y se destruían.



Figura 70: GameScene - 1^{er} incremento

7.2. Segundo incremento

Tras este, se organizó la estructura de la partida, lo que permitiría empezar y acabar una partida, aunque aún no tuviese todas las funcionalidades deseadas.

- Se implementó una clase Enemy muy básica que se limitaba a seleccionar una pieza aleatoria de sushi entre las que le tocaban en ese turno.

- El jugador podía ver, al lado de su plato, unos palillos cuando seleccionase dicha pieza, señal de que podría usarlos a partir de esa ronda, y desapareciendo una vez se hubiesen utilizado.

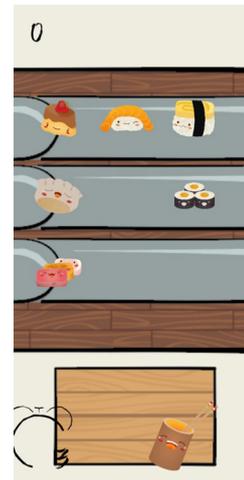


Figura 71: GameScene - 2^o incremento

- Al aumentar el número de detalles en la pantalla, se tuvieron



problemas al tener que hacer una interfaz responsive, es decir, que su contenido se adapte a los diferentes tamaños de pantalla de móviles. Como se puede ver en la figura 71, en caso de que se diseñe para un tipo de pantalla de móvil y se ejecute después en otro, provoca solapamientos y desajuste en la interfaz, cosa que hay que evitar a toda costa.

- Para poder mostrar las puntuaciones, se hicieron dos escenas nuevas: la primera mostraría un título anunciado que se iban a ver las puntuaciones de la ronda en la que se encontrasen y, la segunda, dichas puntuaciones. Para ello, se tuvo que estudiar cómo pasar datos de una escena a otra, ya que se destruyen todos los elementos de la anterior escena y es imposible acceder al script que tuviese esos datos. Se estudió ese problema y, para solucionar aquello, se realizó un Singleton en un script que contendría los datos importantes que deberían guardar de una escena a otra, como la puntuación del jugador o cuántas rondas se habían jugado ya.

La escena de puntuaciones era muy sencilla, ya que principalmente se habían centrado los esfuerzos en la funcionalidad.



Figura 72: TitleRoundScene - 2º incremento

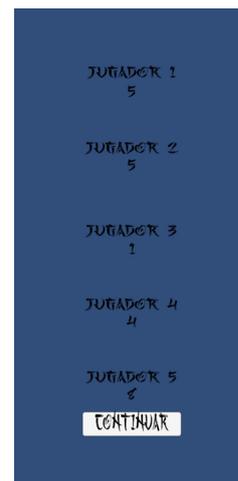


Figura 73: ScoreRoundScene - 2º incremento

7.3. Tercer incremento

Una vez ya se podían jugar partidas, la funcionalidad básica ya estaba hecha, faltaban añadir los detalles para que el jugador pudiese empezar a hacer sus estrategias.

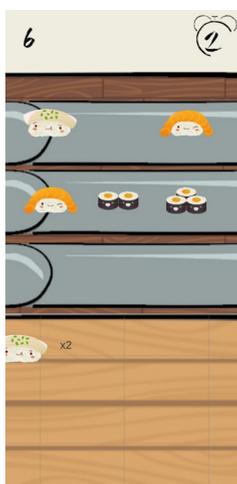


Figura 74: GameScene -
3^{er} incremento

- Una funcionalidad básica que se implementó en este incremento fue que el jugador pudiese ver qué piezas había seleccionado, algo vital para que el jugador tuviese el control de la partida. En la interfaz se dibujó un plato y, si se mantenía pulsado, se agrandaba y aparecían todas las piezas que el jugador había seleccionado.

• Probando este incremento se encontró un bug. Una vez el jugador había seleccionado una pieza de sushi, aunque pulsara en otras, ya no se le permitía elegir más, lo que funcionaba correctamente. Sin embargo, cuando esas mismas piezas de sushi se empezaban a desplazar hacia fuera de la pantalla, el jugador podía volver a seleccionar otro sushi. Esto causaba que el jugador hubiese escogido dos piezas del mismo turno, y no se le permitiese escoger al turno siguiente. Tras esto, el juego no presentaba ningún error y se podía seguir la partida con normalidad pero, a pesar de ello, no era una actitud deseada, por lo que se solventó para el siguiente incremento.

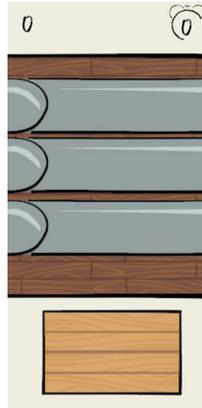


Figura 75: GameScene 2 - 3^{er} incremento

7.4. Cuarto incremento

Con el último incremento se había comprobado que la mecánica de mantener pulsado para ver el contenido del plato podría ser bastante útil para dar información al jugador, así que en un nuevo incremento se reutilizó esta mecánica.

- Se añadió una escena que permitía al usuario seleccionar el número de jugadores en la partida. Como interfaz era bastante simple, pero únicamente se busca la funcionalidad.



Figura 76: SelectPlayersScene - 4^º incremento



Figura 77: VideoRoundScene - 4º incremento

- Entre la pantalla de selección de jugadores y la partida, se añadió una animación que ayuda a situar al jugador en un restaurante japonés. En ella, jugador puede ver su plato y las cintas por las que le llegarán las piezas de sushi. Además de todo esto, también sirve para anunciar el inicio de la próxima ronda.

- En la escena principal de la partida, se añadieron, entre el plato del jugador las cintas donde se movía el sushi, unos círculos. Estos círculos tendrían en el futuro el icono y nombre de cada jugador rival y, manteniendo pulsado en cada uno de ellos, se abriría su plato para ver las piezas de sushi que habrían seleccionado hasta el momento.

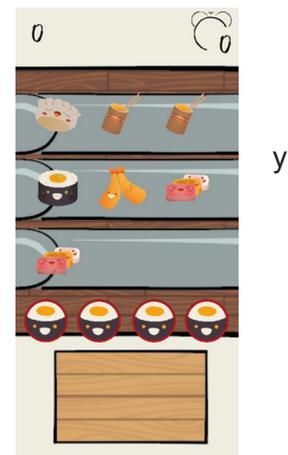


Figura 78: GameScene - 4º incremento

7.5. Quinto incremento

Como resultado del último incremento, se decidió mejorar la interfaz de las puntuaciones ya que, quedaba bastante confuso para el usuario, saber cuál era su propia puntuación. También se decidió cambiar la mecánica para ver los platos, tanto el propio como el del resto de jugadores. Para el usuario, resultaba confuso y añadía demasiado ruido a la pantalla principal.



Figura 79: GameScene -
5º incremento

- Se mejoró la interfaz de la pantalla de la partida. Se le añadió un recordatorio de la ronda en la que se encontraba el jugador y se quitaron los círculos que representaban a los otros jugadores. En el plato del jugador, ahora se mostrarían las piezas de sushi que seleccionase en directo, no haría falta mantenerlo pulsado, esto, le resultaría mucho más cómodo y le daría más tiempo para pensar. El resto de platos de los rivales se trasladarían a otra pantalla, a la cual el jugador podría acceder fácilmente mediante un botón abajo del todo con un texto indicativo.

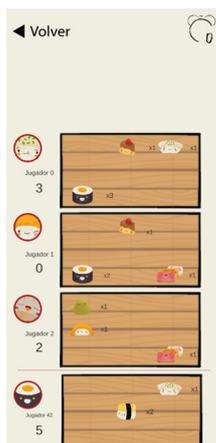


Figura 80: GameScene 2 -
5º incremento

- En la pantalla para ver los platos de los rivales, se podría ver el contenido de cada plato de los rivales, identificados por su nombre y su icono, junto a su puntuación. Abajo del todo, también estaría el plato del jugador, para que, de un vistazo, pudiese comparar su plato con el del resto, para así gestionar mejor una estrategia. Para asegurar que el jugador identifica rápidamente cuál es su plato, aparte de su nombre, hay un separador entre él y los oponentes.

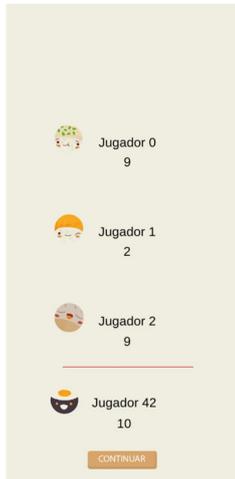


Figura 81: ScoreRoundScene -
5º incremento

- La pantalla de puntuaciones cambió su diseño. Se identificó correctamente a cada jugador y, con el mismo separador que en la pantalla que muestra los platos, se distingue perfectamente cuál es la puntuación del jugador.

- Al implementar la pantalla para visualizar los platos, se tuvo que cambiar el momento en el que los oponentes seleccionaban el sushi ya que, al hacerlo al principio del turno, el jugador podría ver primero qué habían seleccionado los rivales, antes de elegir, lo que sería jugar con ventaja. Para solucionar esto, se obligó a los oponentes a seleccionar justo al final del turno.

- Se añadió música a la pantalla de partida y a la animación de los cocineros al inicio de la ronda.

- Se añadieron efectos de sonido a los botones y a la selección de piezas de sushi.

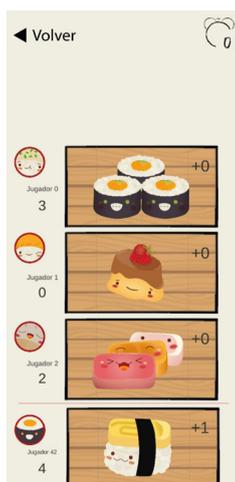


Figura 82: GameScene 3 -
5º incremento

- Al acabar cada turno, el jugador sería redirigido a la pantalla de visualización de todos los platos. Allí, encima de cada plato, se mostraría, por un breve periodo de tiempo, las piezas de sushi elegidas por cada uno de los jugadores junto a las puntuación que se le sumaba ese turno. Tras esto, se volverían a ver los platos normales y se redirigiría a la pantalla de la partida, donde se encontrarían las piezas de sushi del turno siguiente entrando por la cinta.

7.6. Sexto incremento

El último incremento había cambiado bastante el diseño de las pantallas de la partida. Sin embargo, aún quedaba mejorar un poco la presentación y añadir algunas ayudas para el jugador.



Figura 83: GameScene - 6º incremento

- Se corrigió un error que permitía que el jugador pudiese jugar con el móvil en horizontal, la cual no estaba preparada para ejecutarse de ese modo y se desorganizaba toda la interfaz. Para ello se bloqueó la rotación del móvil en los ajustes del proyecto.

- Se añadió en la pantalla de la partida un botón con un símbolo de interrogación. Si el jugador lo mantiene pulsado, se abre, lo que simula un carta del restaurante, y el jugador puede recordar las combinaciones que existen con las piezas de sushi.



Figura 84: ScoreRoundScene - 6º incremento

- En las puntuaciones, se añadió un desglose que especificara cuántos makis y cuántos puddins tenía cada jugador. Junto a este número y, en caso de ser necesario, se indicaría la suma o resta de puntos pertinentes.

- Los nuevos botones funcionaban correctamente, pero el usuario no recibía ningún feedback, por lo que se añadió una animación a cada botón al ser pulsado y un efecto de sonido.



Figura 85: MenuScene -
6º incremento

- Se añadió un menú al iniciar la aplicación. Esto daría la posibilidad de añadirle más funciones y quedaría más profesional. Se hizo más grande el botón de jugar para guiar fácilmente al jugador. En la parte de abajo se añadió una animación, a modo de presentación, donde se pueden ver todos los tipos de sushi que hay en el juego.

- Los Enemy, que hasta ahora solo seleccionaba sushi al hacer fue mejorada para hacer su comportamiento más realista.

- Se rediseñó la pantalla de selección de jugadores.

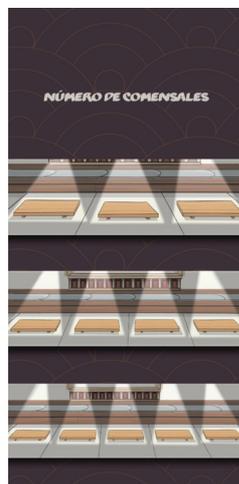


Figura 86: SelectPlayersScene -
6º incremento

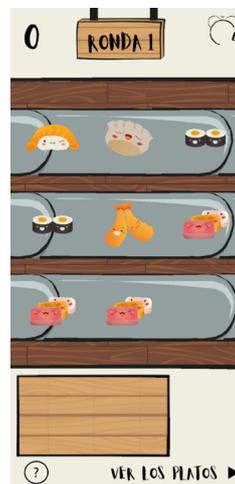


Figura 87: GameScene -
6º incremento

7.7. Séptimo incremento

Por último, en este incremento se añadieron unas últimas pantallas que harían más completa a la aplicación.



Figura 88: WinnerLoserScene -
7º incremento

- Al final de la partida, antes se volvía directamente al menú. Ahora, antes de eso, se accede a una última pantalla que sirve de resumen. En lo alto aparece el jugador con su puntuación y abajo sus rivales, y se indica quién ha sido el ganador de la partida.



Figura 89: HowToPlayScene -
7º incremento

- Se añadió la pantalla de cómo jugar, que mediante un breve texto y unos dibujos aclaratorios, resumen rápidamente las reglas del juego.



Figura 90: CombosScene - 7º incremento

- Se añadió la pantalla de combos, que sirve para que el jugador pueda buscar alguna duda que tenga sobre alguno de los tipos de sushi. También existe la posibilidad de navegar entre las explicaciones de las combinaciones del sushi, pensado para usuarios que aún no sepan jugar.

• Por último se añadió un sistema de registro de usuarios. En la pantalla del menú se encuentra un icono junto con un texto que pone "Iniciar sesión". Este icono nos lleva a una pantalla para iniciar sesión. Como bien se indica en la interfaz, también se puede crear una cuenta, en caso de no disponer de una previamente, o bien acceder a una pantalla para que nos ayude en caso de haber olvidado la contraseña. Una vez se ha iniciado sesión, se redirige automáticamente a la pantalla de perfil donde aparece el nombre del usuario y el botón para cerrar sesión. El texto al lado del icono para iniciar sesión en la pantalla de menú cambiará al nombre del usuario conectado. Este nombre podrá verse también reflejado en la partida. En caso de no haber usuario registrado, en la partida se referirá al jugador con el texto "Tú".



Figura 91: Detalle en MenuScene - 7º incremento



Figura 92: LoginScene - 7º incremento



CAPÍTULO 8

Problemas y desafíos técnicos

Debido a la metodología empleada, se dividió el trabajo en pequeñas tareas para que, empleando poco tiempo, el producto alcanzase más valor. Algunas de ellas presentaron un reto más duro debido a varios factores.

8.1. Estrategia de los oponentes

Por otro lado, dado que en este juego los rivales son controlados por el juego, se debía establecer qué estrategia seguirían a la hora de seleccionar las piezas de sushi.

Al principio, para tener algo básico con lo que probar se optó por hacer que los bots seleccionaran siempre piezas aleatorias. Cuando se jugaba de esta forma era muy fácil ganar para el jugador, lo que resultaba muy aburrido y no ofrecía ningún reto. Por ello, se estudiaron diversas estrategias que podrían seguir los bots como elegir siempre las piezas que diesen directamente la mayor cantidad de puntos, es decir sin necesidad de hacer ningún tipo de combinación, o por el contrario, también podrían seguir la estrategia de intentar conseguir siempre combinaciones completas. Otra idea fue que siempre se fijaran en qué piezas seleccionaba el jugador y fuesen a por las mismas, con la idea de, si por ejemplo el jugador cogía una tempura y necesitaba otra para cerrar el combo y conseguir 5 puntos, el resto de bots también irían a por la tempura, con el fin de que el jugador se quedase solo con una y no anotase ningún punto de ella. Esta estrategia podía resultar



bastante frustrante al jugador al elevar demasiado la dificultad, por lo que se descartó. También se pensó en una estrategia más completista, donde los bots seleccionarían cada vez un tipo de sushi, como si intentara hacerse con una pieza de todos los tipos. Esta estrategia resultó bastante absurda, ya que el resultado final se parecería bastante a la selección aleatoria de la que se huía. Tras descartar unas cuantas propuestas más que no acababan de encajar, se escogió una estrategia que humanizaba bastante su comportamiento, lo que haría la partida justa y divertida para el jugador.

La estrategia dota a los oponentes de una preferencia sobre los tipos de sushi respecto a otros. Cada Enemy tiene una lista de preferencia de los tipos elegidos aleatoriamente al inicio de cada partida. A partir de ahí, a la hora de escoger, cada bot mirará sus preferencias e intentará seleccionar una pieza de sushi acorde a ella. Sin embargo, antes de seleccionar la pieza finalmente, debe estudiarse si seleccionar esa pieza le va a venir bien a su puntuación, es decir, si tiene sentido coger ese sushi. Por ejemplo, si en el momento de seleccionar intentar coger una gyoza pero ya tiene cinco en su plato, al coger esa sexta gyoza no le sumaría ningún punto, por lo que mirará la siguiente pieza en su lista de preferencias.

De esta forma el juego sigue ofreciendo un reto al jugador, dando la sensación de que está jugando con personas reales.

8.2. Aparición de las piezas de sushi en pantalla

Por otro lado, se quería seguir la temática del restaurante japonés donde las piezas de sushi van en una cinta, por ello, para el juego, no servía que los sushi apareciesen en pantalla y desapareciesen, debían entrar transportados en una cinta y luego salir también con ella.

Al principio del desarrollo este fue uno de los problemas con los que se tuvo que lidiar. Entonces, la aparición de sushi se construía con un



método que recorría la miniDeck del jugador e iba instanciando los prefab correspondientes según el tipo, modificando su componente Transform para que se fueran colocando en una especie de cuadrícula fuera de la pantalla del jugador. A cada uno de ellos se le asignaba un mismo GameObject vacío que actuaría como padre. Este objeto padre sería el encargado de mover a todo el grupo hasta el centro de la pantalla y, después de que el temporizador llegase a cero, moverlos hacia el otro lado de la pantalla hasta que quedarán fuera de la vista del jugador. De esta forma funcionaba y daba el aspecto que se quería pero, más tarde, se estudió que la instanciación de un Prefab por código suponía un coste elevado en Unity [50], por lo que se intentaron estudiar otras opciones.

La opción que, finalmente se utilizó fue el uso de una plantilla. Se creó un prefab con varios GameObject de ejemplo en forma de cuadrícula, este Prefab se asignaría a la clase Turn, la encargada de gestionar la aparición de los sushi, mediante el Editor de Unity y esta, al recorrer la miniDeck correspondiente, iría sustituyendo los sprites de los GameObjects del Prefab por los sprites de las piezas correctas. De esta manera se podía incluso añadir el sprite de la cinta en la plantilla, y se ahorraban las llamadas al método de Unity para instanciar.

8.3. Problemas con la desactivación de objetos

Uno de los errores que se encontró en uno de los incrementos, fue que el indicador que avisaba al jugador de que podía usar los palillos. Idealmente, cuando el jugador seleccionara los palillos entre las piezas de sushi de su turno, unos palillos aparecerían al lado de su plato. Esto, en la realidad, funcionaba correctamente con los primeros palillos que seleccionase el jugador. Si llegaba a usarlos y se desactivaba el indicador, aunque posteriormente seleccionase otros palillos, este ya no volvía a aparecer. Tras examinarlo y depurar el código detenidamente, se descubrió el error.



Una vez el jugador utilizaba los primeros palillos, el GameObject que ejercía de indicar al lado del plato, se desactivaba mediante el método `SetActive(false)`. Lo que no se sabía, es que este método provocaba que, el script que manejaba este indicador, perdiese la referencia, por lo que al volverlos a seleccionar, intentaba volver a activarlos mediante el método `SetActive(true)`, pero no se encontraba el GameObject.

Finalmente, lo que se hizo para solucionar este problema, fue dejar de usar el método `SetActive(false)` en este caso. En su lugar, el GameObject siempre estaría activo pero se le quitaría el sprite, por lo que quedaría invisible al ojo del jugador. Después, para volver a activarlo, se le asignaría de nuevo el sprite correspondiente, que sería asignado al script mediante el Editor de Unity. Por lo tanto, ya se podrían seleccionar y utilizar tantos palillos como se quisiese en la partida, que el jugador siempre tendría su avisador y no habría más problemas.



CAPÍTULO 9

Conclusión

Tras todo el desarrollo de la aplicación y viendo los resultados obtenidos, considero que ha sido un proyecto bastante enriquecedor. A nivel personal ha resultado un reto poder combinar trabajos tan dispares como los diseños gráficos y las animaciones con la programación y el trabajo en Unity. Me ha resultado interesante el proceso de desarrollo de un videojuego en este motor gráfico y de cara al futuro resultaría interesante seguir investigando por este camino.

Gracias a los múltiples recursos en Internet he podido ir aprendiendo poco a poco cómo manejar Unity, en el cual no tenía experiencia anterior. También, los aspectos gráficos como el dibujo de los sprites, la creación de algunas ilustraciones como la del inicio de ronda con los cocineros, han requerido muchísimo trabajo y me han obligado a perfeccionar el uso de Adobe Illustrator, ya que era importante hacer un trabajo minucioso para luego poder llevarlo al programa de animación y poder darle vida con la libertad de estar trabajando con una ilustración consistente. En el caso particular de las animaciones se han estudiado los principios de la animación para poder dotarlas de mayor naturalidad y que el resultado no fuese torpe e impreciso.

Analizando el trabajo en retrospectiva se aprecia que el inicio del proyecto, junto con mi desconocimiento de Unity, fue un poco desorganizado lo que, poco a poco, fue solventándose gracias a una correcta utilización del sistema de versiones y al ir ganando experiencia poco a poco, tanto en Unity, como en la programación de sus scripts. Poco a poco se han ido



conociendo funcionalidades del motor que antes no se conocían y, muchas veces se ha tenido que rehacer trabajo para hacerlas de otro modo, tiempo que, con un mayor conocimiento de Unity, se hubiese evitado perder.

9.1. Relación con los estudios cursados

En este TFG se han aplicado los conocimientos adquiridos en diversas asignaturas a lo largo de mis estudios.

En relación con las metodologías y el desarrollo de proyectos, las asignaturas de Proyecto de Ingeniería de Software y Proceso de Software han desempeñado un papel crucial. La aplicación de la metodología Kanban en el desarrollo del juego ha sido posible gracias a la comprensión obtenida en estas asignaturas, permitiendo gestionar eficientemente las tareas y mantener una visualización clara del progreso.

El conocimiento de patrones de diseño, como el Singleton, derivado de la asignatura de Diseño de Software, ha influido directamente en la estructura del código del proyecto, resultando vital para algunas partes importantes del desarrollo.

Además, la experiencia en la asignatura de Interfaces Persona-Computador ha influido en la mejora de las interfaces del juego, asegurando una interacción más intuitiva y eficaz para el usuario. Se han diseñado interfaces que sigan principios de usabilidad y brinden una experiencia de juego fluida.

El estudio de la asignatura de Postproducción de Video Digital también ha mejorado el uso de After Effects.

El estudio de la asignatura de Análisis y Especificación de Requisitos también se ha visto reflejado en este trabajo, donde se han utilizado las



historias de usuario como método de especificación de requisitos. La habilidad para comprender y extraer unidades de trabajo a partir de estas historias ha sido esencial en la planificación y ejecución de las tareas del proyecto, asegurando una implementación coherente con los requisitos identificados.

Por último, se han aplicado los principios de Gestión de Versiones adquiridos durante el estudio de Mantenimiento y Evolución de Software. El uso efectivo de sistemas de control de versiones ha facilitado el seguimiento y la gestión de los cambios realizados en el proyecto a lo largo del tiempo.



CAPÍTULO 10

Trabajos futuros

Tras tener una aplicación estable como la obtenida de este TFG, se podrían implementar más funcionalidades, como la opción de un multijugador online donde poder crear partidas privadas, para jugar con amigos o públicas.

Esto obligaría a crear un sistema en el que los usuarios pudieran agregarse uno a otros como favoritos, además de, quizás un pequeño chat en el juego donde comunicarse.

Aparte de todo esto, se podrían añadir más funcionalidades al perfil de usuario, como la personalización de los iconos, pudiendo elegir entre un catálogo de ilustraciones. Este catálogo podría verse también como una pequeña tienda en la que el jugador pagase con una moneda del juego. Esta moneda del juego podría conseguirse con ciertos hitos en el juego o, directamente con micropagos en la aplicación.

Otra funcionalidad que se podría añadir al perfil del jugador, sería un registro de las últimas cinco o diez partidas, o también una estadística global que midiese sus partidas jugadas, las ganadas y las perdidas. También sería interesante integrar un sistema de logros, que podrían recompensar al jugador por ser activo en la aplicación o alcanzar diferentes objetivos.

Tras todo este trabajo, se podría valorar la idea de contactar con la empresa creadora del juego de cartas para informar de este desarrollo. Para entonces, la aplicación debe lucir profesional tanto en apariencia como en rendimiento, siendo una aplicación robusta, atractiva y fiable.



CAPÍTULO 11

Referencias

- [1] D. Losada Figueiras, "¿Cuáles son los géneros de Juegos Para Móvil Más jugados?", Game It - Consolas, videojuegos y hardware. Gaming Culture. <https://www.gameit.es/cuales-son-los-generos-de-juegos-para-movil-mas-jugados/> (2023, octubre 20)
- [2] I. Benlloch, "Casual", Geekno. <https://www.geekno.com/glosario/casual> (2023, octubre 20)
- [3] "Sushi go!", Gamewright. <https://gamewright.com/product/Sushi-Go> (2023, octubre 20)
- [4] "Overview", BoardGamesGeek. <https://boardgamegeek.com/boardgame/133473/sushi-go> (2023, octubre 20)
- [5] "Sushi Go Party!", Gamewright. <https://gamewright.com/product/Sushi-Go-Party> (2023, octubre 20)
- [6] "Sushi Roll", Gamewright. <https://gamewright.com/product/Sushi-Roll> ((2023, octubre 20)
- [7] "Sushi Go - Spin Some for Dim Sum", Gamewright. <https://gamewright.com/product/sushi-go-spin-some-for-dim-sum> (2023, octubre 20)
- [8] "Expansions", Gamewright. <https://boardgamegeek.com/boardgame/133473/sushi-go/expansions> (2023, octubre 20)



- [9] H. Kniberg, M. Skarin, *Kanban y Scrum – obteniendo lo mejor de ambos, Estados Unidos de América, 2014* [En línea] Disponible en: http://www.proyectalis.com/documentos/KanbanVsScrum_Castellano_FINAL-printed.pdf
- [10] A. Legastelois, "Los kaitenzushi", Japan Experience. <https://www.japan-experience.com/es/preparar-viaje/saber-mas/gastronomia-japonesa/los-kaitenzushi> (2023, octubre 21)
- [11] "SushiGoTM-RULES", Gamewright. <https://gamewright.com/pdfs/Rules/SushiGoTM-RULES.pdf> (2023, octubre 21)
- [12] Crossy Road - Endless Arcade Hopper Game. <https://www.crossyroad.com/> (2023, octubre 23)
- [13] "Classic Solitaire Klondike", Google Play. <https://play.google.com/store/apps/details?id=com.potatojam.classic.solitaire.klondike&hl=es&gl=US> (2023, octubre 23)
- [14] "Uno!", Google Play. <https://play.google.com/store/apps/details?id=com.matteljv.uno&hl=es&gl=US> (2023, octubre 23)
- [15] Piffle. <https://www.pifflegame.com/> (2023, noviembre 9)
- [16] "Tsuki Adventure", HyperBeardGames. <https://hyperbeard.com/game/tsuki/> (2023, noviembre 9)
- [17] S. Fernández, "Así fue Flappy Bird, el juego que se hizo tan popular que su creador tuvo que matarlo.", Xataka Android. <https://www.xatakandroid.com/juegos-android/asi-fue-flappy-bird-juego-que-se-hizo-popular-que-su-creador-tuvo-que-matarlo> (2023, noviembre 9)
- [18] Inkscape. <https://inkscape.org/es/> (2023, octubre 25)



- [19] Gimp. <https://www.gimp.org/> (2023, octubre 25)
- [20] Adobe Illustrator. <https://www.adobe.com/es/products/illustrator.html> (2023, octubre 25)
- [21] Figma. <https://www.figma.com/> (2023, octubre 26)
- [22] "Introducción a XD", Adobe. <https://helpx.adobe.com/es/xd/get-started.html> (2023, octubre 26)
- [23] Blender. <https://www.blender.org/> (2023, octubre 26)
- [24] Adobe After Effects. <https://www.adobe.com/es/products/aftereffects.html> (2023, octubre 26)
- [25] DaVinci Resolve. <https://www.blackmagicdesign.com/es/products/davinciresolve> (2023, octubre 26)
- [26] GarageBand. <https://www.apple.com/es/ios/garageband/> (2023, octubre 26)
- [27] Audacity. <https://www.audacityteam.org/> (2023, octubre 26)
- [28] Unreal Engine. <https://www.unrealengine.com/es-ES> (2023, octubre 27)
- [29] Unity. <https://unity.com/es> (2023, octubre 27)
- [30] "Aprendiendo la Interfaz ", Unity Documentation. <https://docs.unity3d.com/es/530/Manual/LearningtheInterface.html> (2023, octubre 27)
- [31] "Escena", Unity Documentation. <https://docs.unity3d.com/es/530/Manual/CreatingScenes.html> (2023, octubre 27)



- [32] " GameObject ", Unity Documentation. <https://docs.unity3d.com/es/530/Manual/class-GameObject.html> (2023, octubre 27)
- [33] " Etiquetas(Tags) ", Unity Documentation. <https://docs.unity3d.com/es/530/Manual/Tags.html> (2023, octubre 27)
- [34] " Prefabs ", Unity Documentation. <https://docs.unity3d.com/es/530/Manual/Prefabs.html> (2023, octubre 27)
- [35] Github. <https://github.com/> (2023, octubre 30)
- [36] Git. <https://git-scm.com/> (2023, octubre 30)
- [37] GitHub Desktop. <https://desktop.github.com/> (2023, octubre 30)
- [38] Auth0. <https://auth0.com/> (2023, octubre 30)
- [39] "Firebase Athentication", Documentación de Firebase. <https://firebase.google.com/docs/auth?hl=es-419> (2023, octubre 30)
- [40] Notepad++. <https://notepad-plus-plus.org/> (2023, octubre 30)
- [41] Visual Studio Code. <https://code.visualstudio.com/> (2023, octubre 30)
- [42] Eclipse. <https://www.eclipse.org/ide/> (2023, octubre 30)
- [43] Visual Studio. <https://visualstudio.microsoft.com/es/> (2023, octubre 30)
- [44] "Documentación de C#". Microsoft. <https://learn.microsoft.com/es-es/dotnet/csharp/> (2023, octubre 30)
- [45] "Documentación de Javascript", MDN Web Docs. <https://developer.mozilla.org/es/docs/Web/JavaScript> (2023, octubre 30)



[46] "Boo", Wikipedia. [https://es.wikipedia.org/wiki/Boo_\(lenguaje_de_programación\)](https://es.wikipedia.org/wiki/Boo_(lenguaje_de_programación)) (2023, octubre 30)

[47] M. Rehkopf, " Historias de usuario con ejemplos y plantilla ". Atlassian. <https://www.atlassian.com/es/agile/project-management/user-stories> (2023, noviembre 5)

[48] J. Cuello, J. Vitterone, *Diseñando apps para móviles*, CreateSpace Independent Publishing Platform, 2014 [En línea] Disponible en: <https://appdesignbook.com/es/contenidos/patrones-interaccion-moviles/#:~:text=La%20«Ley%20del%20pulgar»%2C,los%20elementos%20en%20la%20interfaz> (2023, noviembre 10)

[49] A. Pérez, " Características y fases del modelo incremental ", OBS Business School. <https://www.obsbusiness.school/blog/caracteristicas-y-fases-del-modelo-incremental> (2023, noviembre 17)

[50] "Recomendaciones de rendimiento para Unity", Microsoft. <https://learn.microsoft.com/es-es/windows/mixed-reality/develop/unity/performance-recommendations-for-unity?tabs=openxr> (2023, noviembre 18)



ANEXO A

OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.		X		
ODS 4. Educación de calidad.		X		
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.				X
ODS 9. Industria, innovación e infraestructuras.				X
ODS 10. Reducción de las desigualdades.				X
ODS 11. Ciudades y comunidades sostenibles.				X
ODS 12. Producción y consumo responsables.				X
ODS 13. Acción por el clima.				X
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos.				X



Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.

- **ODS 3 Salud y bienestar:** El videojuego desarrollado durante este TFG proporciona una forma de entretenimiento, brindando a los jugadores un respiro del estrés diario. Esta actividad lúdica contribuye al bienestar general. La dinámica relajada del juego, junto con su temática amigable, crea un ambiente propicio para reducir el estrés. Participar en un juego de estrategia ligera puede ayudar a aliviar tensiones y ofrecer un escape mental.

- **ODS 4 Educación de calidad:** Este juego involucra a los jugadores en la planificación y ejecución de estrategias para maximizar sus puntos. Este proceso fomenta el pensamiento estratégico y la toma de decisiones. La repetición del juego estimula al jugador a crear nuevas estrategias o probar nuevos enfoques, lo que también desarrolla su creatividad.